設 計 自 動 化 23－2
（1984. 9. 18）

並列ルーティング・プロセッサの試作
A Parallel Routing Processor: Experimental Result

中島 聡　　鈴木 敬
Satoshi NAKAJIMA　Kei SUZUKI

橘 昌良　　大附 辰夫
Masayoshi TACHIBANA　Tatuo OHTUKI

早稲田大学 理工学部
School of Science and Engeneering　WASEDA University

（1） はじめに

ＬＳＩやプリント基板の設計において，レイアウト設計は最も時間とコストの要する部分である．そして，近年の集積度の増加に伴い，この傾向はますます強くなっている．

我々は，そのレイアウト設計のうち，最も手間のかかる部分の１つである配線径路探索を，ハードウェアで並列的に高速に行なわせるルーティング・プロセッサ（以下ＲＰと呼ぶ）の研究を行なって来た．[1),2)]

径路が存在すれば，必ず最短径路をみつけるＬｅｅの迷路法のアルゴリズムは，図１－１のように，ソースからターゲットまで "Wavefront" を拡げることにより探索を行なうものである．そのため，通常のソフトウェアによる方法では，一つの径路当り$O(L^2)$（Ｌ：距離）の時間を要する．それに対し，グリッド上の１セルに１つのプロセッサを割当てて，並列に処理を行なえば，時間複雑度はＯ（Ｌ）となる．このような１セル１プロセッサ型のＲＰについての研究は，すでに，いくつか行なわれている[3)-6)]

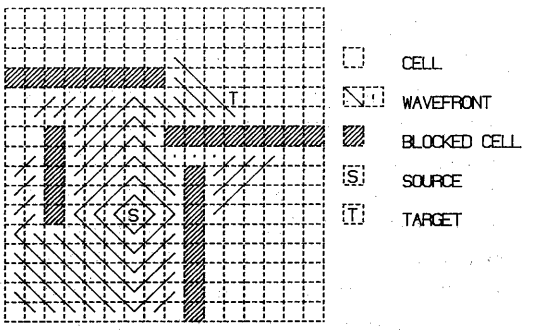


図１－１ Ｌｅｅの迷路法による径路探索

（プロセッサ数は，グリッドをＮ＊Ｎとしたとき，N^2 である．）

しかし，１セル１プロセッサ型では，Wavefront上にあるセル（ブロックのない状態で最大２Ｎ個）しか働かず，並列性が十分に生かされない．また，現在のようにＮが大きな値（数千）になると，それに必要なプロセッサ数（N^2）は膨大なものとなる．そのため，プロセッサのLSI化が必要となるが，その場合でも１チップあたりのプロセッサ数は限られるので（ピン数による），やはり装置は大規模になる．

そこで，我々は，１つのプロセッサに複数のセルを受もたせることにより，現実的なプロセッサ数（Ｏ（Ｎ））で並列処理をさせるＲＰの設計を行った．ただし，セルの割当てには，Wavefront の性質を利用し，各プロセッサの稼働率ができるだけ高くなるよう工夫をした．これにより，１セル１プロセッサと同程度の処理速度を得られる見通しが立った．

今回は，その設計に基づき，実際のＲＰの試作を行ったので，その結果について報告する．

（2） 並列ルーティングプロセッサの構成

[1] 基本構成

ＲＰの運用形態としては，図２－１に示すように，ホスト・コンピュータのバックエンドプロセッサとしての役割を想定する．よって，グリッド情報，ネット情報を収容したファイルの管理，オペレータとの情報交換などはホスト・コンピュータが行ない，並列処理に適する部分（主に迷路法による径路探索）のみをＲＰが実行する．ホスト・コンピュータ，ＲＰ間で転送されるデータは，ホスト・コンピュータの負荷を減らすために，必要最小限のものになる．つ

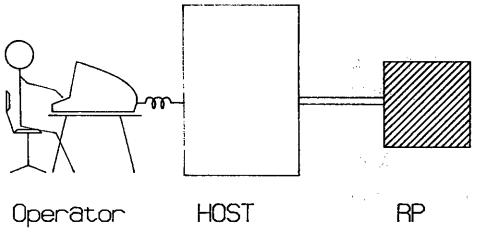


図２－１　ＲＰの運用形態

まり，ホスト・コンピュータからは，ネット情報のみがＲＰに渡され，ＲＰからは，径路情報が返される．

ＲＰのハードウェア構成を図２－２に示す．ＲＰは，ホストとの通信を行なうコントロール・ユニット（ＣＵ）と，実際に径路探索を行なうプロセッサ・ユニット（ＰＵ）とから構成される．ＰＵは，格子状に配置されたプロセッサ・エレメント（ＰＥ）群からなる

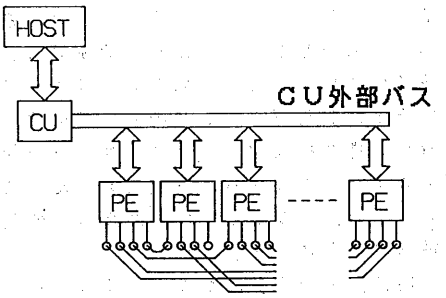


図２－２　ＲＰの構成

ＣＵは，ホストとの通信，グリッド情報の管理，ＰＵの制御を行なう．グリッド情報をＣＵが持つことにより，ホストの負担は，大幅に軽減される．また，ＰＵの制御としては，ＰＥの動作の指示，ＰＥ間の通信の制御を行なう．ＰＥ間の同期は，このＣＵの制御により行なわれる．

このＰＥの配列である格子グラフの性質として，２部グラフで表せる点がある．これは，ＰＥを黒／白の２群に分けることができることを示し，これはまた，ソースが１つである場合のWavefront，必ず同一の色のＰＥが来ることを示す．よって，Wavefront の”伝播”は，すべてが黒から白，または，すべてが白から黒の２通りしかない．この性質を利用して，ＰＥを黒／白の２群に分けて制御することにより，ＰＥ間の通信制御の煩雑さを，大幅に軽減している．

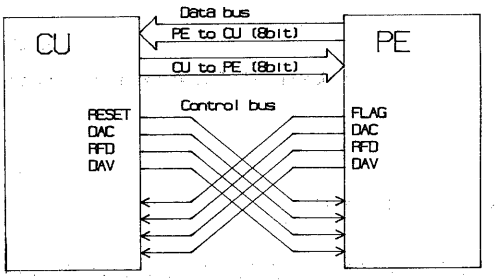


図２－３　ＣＵバス

［２］試作したＲＰのハードウェア

今回試作したプロセッサは，ＣＵとしては，プログラムの記述を簡単にするために仮想的にホスト・コンピュータに持たせた．ホスト・コンピュータとは，８bit パラレルインターフェースにより接続され，高速なデータ転送が可能である．また，グリッド情報としては，１セルあたり，１byteで，１２８＊１２８セルの情報を持つことができる．

ＣＵとＰＵ間は，ＣＵバスと呼ぶバスで接続されている（図２－３）．ＰＵ内のすべてのＰＥは，このバスと接続され，このバスを通じて，

１）ＣＵからすべてのＰＥ，
２）ＣＵから特定のＰＥ，
３）１つのＰＥからＣＵ，

の３通りの通信が行なわれる．そのため，ＣＵバスは図２－３に示すように３線のハンドシェイクとなっている．ＰＥからＣＵへの信号線は，オープンコレクタ方式により，ＣＵは複数のＰＥの信号のＡＮＤ（又はＯＲ）を読み込むことができる．また，ＣＵバスにアドレスバスに相当するものはないが，これはＣＵがＰＥに与えるコマンド列中にＰＥの指定を含むことにより解決している．

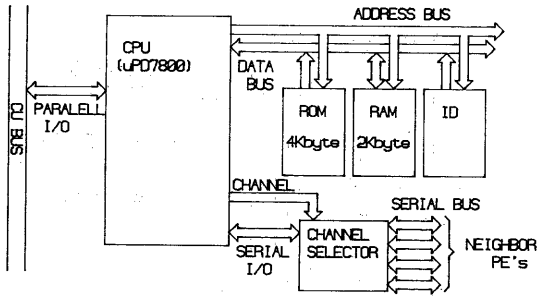


図２－４　ＰＥの構成

ＰＥの構成を図２－４に示す．ＰＥは１つ１つがストアードプログラム方式で動作するプロセッサである．ＣＰＵにワンチップマイクロプロセッサμPD7800を採用することにより，１ＰＥあたり１１チップのコンパクトな構成となっている．メモリはＲＡＭ ２Kbytes，ＲＯＭ ８Kbytesを持ち，ＲＯＭはＰＲＯＭを使用している．ＲＯＭには，ＰＥの動作が記述されているが，その内容はすべてのＰＥで全く同じである．そのため，各ＰＥはＩＤナンバーをスイッチにより持ち，自分の担当するセルを知る．

また，隣接するＰＥどうしは，シリアルバスにより接続され，相互に通信が可能である．ただしμPD7800がシリアルチャンネルを１つしか持たないため，ＣＭＯＳスイッチにより４チャンネルに切り換えて使っている．そのため，通

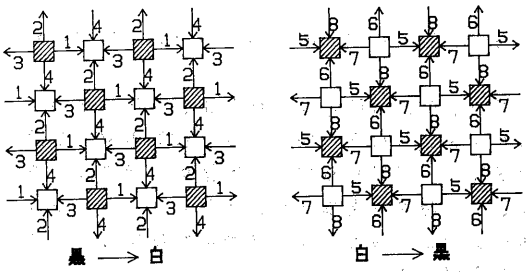


図２－５　競合しないチャンネルセレクト

信時には，両ＰＥともにチャンネルを正しく選択しなければならない．そこで，黒から白４通り，白から黒４通りの，計８通りの競合しないチャンネル選択をＣＵからの制御で行なわせる（図２－５）．このシリアルバスを通じては，ラベル付けと，逆追跡が行なわれ，必要なセルの座標値のみが，転送される．

試作したＲＰの諸元を表２－１に示す．

表２－１．ＲＰ諸元

プロセッサ構成	8*8 (TWISTED TORUS)
対象セル	128*128
グローバル通信	双方向8bitパラレル
	Tg=650μs/2bytes
ローカル通信	４方向シリアル
	Tl=85μs/1byte
ＰＥ構成	CPU μPD7800 2MHz
	ROM 8Kbytes
	RAM 2Kbytes

［３］セル割当規則

また，このＲＰでは，一台のＰＥに複数のセルを担当させている．そのため，セルの割当て方によっては，Wavefront 上のセルのＰＥに対する割当てが偏ったものとなり，並列性を著しく低下させる．よって，セルの割当てには，できるだけWavefront上のセルのＰＥへの割当てが平均するような工夫をする必要がある．

最適な割当ては，Wavefront の形状がそのつどことなるため，動的にしか求められないが，それでは割当てに大きな手間がかかってしまう．そこで，Wavefront の性質を利用して，静的だが，比較的良い割当てを定める．

Wavefront が２次元に，線分の集りの形で広がることに着目すれば，セル割当ては，１つのプロセッサがいくつもの連続したセルを受けもつことは，好ましくない．そこで，１つのプロセッサが担当するセルは，とびとびに（かつ規則的に）並ぶようにする（条件１）．また，Wavefront が２種の斜め線分（45°,135°）から構成されることより，その２種の斜め線分上では，セル割当てが均等になるようにする（条件２）．以上の二つの条件を満たすためプロセッサを”ひねったトーラス（twisted torus)”型に接続する．この接続による，セルマップ上での，プロセッサ番号の配列は，一般化すれば，(2a*2b) の長方形を，
(2c)づつずらした形となる（図２－６）．ただしこのとき以下の関係が成立っていなければならない．

$$G.C.M(|a-c|,b)=1 \qquad -(2.1)$$

$$G.C.M(|a-b+c|,b)=1 \qquad -(2.2)$$

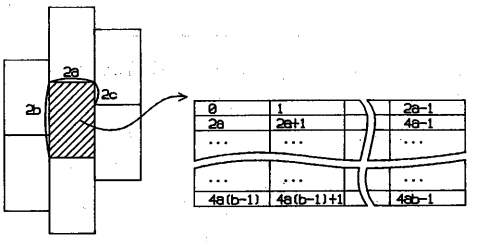


図２－６　セルマップ

この条件が満たされていれば，Wavefront を構成する２種の斜め線上には，Np/2（Npはプロセッサ数）個おきにしか同じプロセッサの担当するセルは現われない。

(2.1),(2.2)式を満たす a,b,cの組を決めるのは，容易である。例えば，$a=2^i$，$b=2^j$，c=1（i,j は整数）であれば,(2.1),(2,2)式は常に満足される。

試作したＲＰは，a=8，b=8，c=1 のプロセッサ数 64 個のものである（図２－７）。

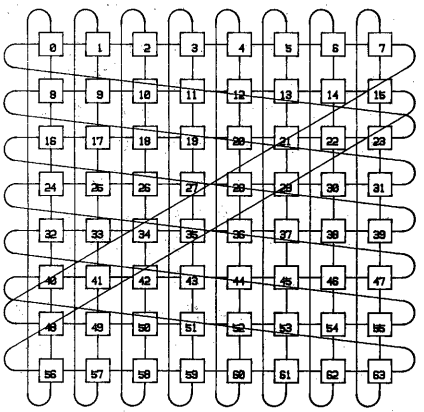


図２－７　ＰＥ接続図

［４］ＲＰの動作

動作としては，細分化されたＰＥの動作を，ＣＵにより指示してゆく形を採った。つまりＰＥは，ＣＵからコマンドの形で指示されたプロセスを実行することになる。

コマンドは，すべてのプロセッサにたいして送られるため，あるコマンドが受付けられるのは，１つ前のコマンドに対するすべてのＰＥの処理が終了してからである。ＰＥ間の同期は，このＣＵからのコマンドにより行われる。

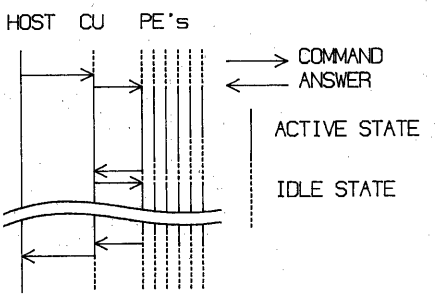


図２－８　ＣＵ，ＰＥの実行状態

そのため，ＰＥの状態は，コマンドによりなんらかのプロセスを実行している時（active state）と，次のコマンドがＣＵから来るのを（すなわち他のＰＥの処理の終るのを）待っている時（idle state）とに分けられる。（図２－８）

ここで，”稼働率”の定義をしておく。＃ｉのプロセッサの active state，idle state の各時間をそれぞれ Tia，Tib としたとき，稼働率Ｐを以下のように定義する。

$$P = \sum_i Tia \;/\; \sum_i (Tia + Tib) \qquad -(2.3)$$

(2.3)式は，ＣＵによる待時間を無視し，かつすべてのＰＥが同期しているとすれば，以下のように変更できる。

$$P = \sum_i Tia \;/\; (\max_i (Tia) * Np) \qquad -(2.4)$$

プログラムの記述は，細分化されたＰＥのプロセスをＰＥのＲＯＭ上にμPD7800のマシン語で記述し，ＣＵからは，Ｃ言語中でサブルーチンとして呼ぶ形を採った。ただし，間にはサブルーチンコールをコマンド列に変換するプロセスが必要である。

実際にＲＰによるルーティングは，Wavfront を広げるフェーズＰと，逆追跡を行うフェーズＢより成る。これらの動作は，ＣＵとのコマン

表２－２．２つのフェーズにおける通信回数

フェーズ	グローバル通信	ローカル通信
Ｐ	１０	４＊Ｑ
Ｂ	１０	１

Ｑ：ラベルキューの深さ

ド／データのやりとりのグローバル通信と，ＰＥ間のローカル通信に大部分の時間がかかる．各フェーズの通信は，表２－２のように行われる．
よって，各フェーズに要する時間は，

$$Tp = Tp0 + 10Tg + 4Q(Tl+Tq) \quad -(2.5)$$
$$Tb = Tb0 + 10Tg + Tl \quad -(2.6)$$

Tg: グローバル通信にかかる時間
Tl: ローカル通信にかかる時間
Tp: フェーズＰに要する時間
Tp0: Tp中通信に関係しない一定の時間
Tq: Tp中ラベル付けに関係する時間
Tb: フェーズＢに要する時間
Tb0: Tb中通信に関係しない一定の時間

で表される．しかし，フェーズＰの方は，ＰＥごとに持っているラベルの数（＝Ｑ）が異なり，かつ，Wavefront の性質上，早く処理の終ったＰＥも，１番時間のかかったＰＥの処理が終るまで次の処理に移れないので，実際に要する時間は，

$$Tp = Tp0 + 10Tg + 4\max_i(Qi) * (Tl+Tq) \quad -(2.6)$$

Qi: #iのＰＥのキューの深さ

そして，長さｄの径路を求める時間Trは，

$$Tr = \Sigma(Tpj + Tbj) = d(Tp0 + Tb0 + 20Tg + Tl) + \Sigma_j 4\max_i(Qij) * (Tl+Tq) \quad -(2.7)$$

である．
なお，Tqは，送信側がラベルキューからデータを取り出して，座標変換するのと，受信側が受け取ったデータよりセル座標を見て，必要であればセル情報を変更し，キューへ入れる時間である．
(2.7)式より，Tp0, Tb0, Tg, Tl, Tqは一定なので，
「ルーティングにかかる時間が距離ｄに比例するためには，$\max_i(Qij)$が距離ｄによらず，一定値以下であることが必要である．」
ということが分る．

（３） 実験結果および考察

実験は，まず単一の径路に対して，禁止領域のない場合，ある場合について，フェーズＰにおける各プロセッサのラベルキューの深さ，処理時間をWavefront 一世代ごとに求めた．そして次に，いくつか具体的な複数の配線例について，処理時間，ＰＥの稼働率，ラベル付け回数を記録した．

[１] 単一径路

四つの例について，フェーズＰの時間経過を，Wavefront 一世代ごとにプロットしたのが図３－１である．(1) と(2) が禁止領域のない場合で，(1) が配線領域の中心から，(2) がすみから初めたものである．(3) と(4) が禁止領域のある場合で，(3) は10ブロック，(4) は20ブロックの禁止領域がランダムに存在している例である．
以上４つの例について，同時に各世代におけるラベル付け回数（Qij の最大値）を記録して，(2.7) 式に基づき，Tg, Tl, Tp0, Tq, Tb0を求めた（表３－１）．

表３－１．ＲＰ各種処理速度

Tg （グローバル通信）	650μｓ
Tl （ローカル通信）	85μｓ
Tp0（フェーズＰ固定時間）	700μｓ
Tq （フェーズＰラベル付け時間）	3300μｓ
Tb0（フェーズＢ固定時間）	2500μｓ

そして，これらの値に基づいて，ラベル付け回数が１および８で一定だった場合の処理速度を，それぞれ実線ａおよびｂとして図３－１に付した．
図３－１を見て分る通り，(1)～(4)のすべての配線例について，a,b の間にはさまれた領域にプロットがあり，ラベル付け回数は，高々８（実際にこの例では７）に押えられている．また，最悪の（禁止領域のない）場合の例(1) では，Wavefront が壁にぶつかるまで，距離の２乗のオーダーの処理時間がかかっているのに対し，通常の（禁止領域のある）例においては，ほぼリニアに上昇していることが分る．

［２］複数配線

次に，適当にソースとターゲットを定めて作ったネットリストについて，径路を求めた結果にを報告する．

配線例は，図３－２(a)～(d)の四通りで，それぞれの径路について，フェーズＰに要した時間をその径路長を横軸にプロットしたのが図３－３である．図３－１と同じく実線a,b を付してある．

図３－３においても，すべてのプロットは実線a,b 間にあり，ラベル付け回数の径路あたりの平均が８を越えていないことが分る．

また，距離と時間の関係を分りやすくするために，同じデータをLog-Log でプロットしたのが，図３－４である．

図３－４をみると，バラ付きはあるものの，だいたい直線上に乗っており，直線を引くと，傾き1.16であり，ほぼＯ（ｎ）の時間複雑度と言える．

また，図３－５(a),(b) には，径路の距離を横軸にとって，平均ラベル付け回数，稼働率をプロットしてあるが，図３－３に対応して，８を越えるものはなく，距離との相関関係も持っていない．また，稼働率は10%～70%まであるが，径路長の短い所で稼働率が低い傾向があるのは，Wavefront 上のセルの数が少ないためで，この場合は，平均ラベル付け回数が１～２回にしかならないためである．

表３－２．複数配線例の結果

	(a)	(b)	(c)	(d)
ネット数	59	30	35	15
ブロック数	6	24	14	7
総配線長	3079	1510	989	1451
平均配線長	52	52	28	97
処理時間(s)	78	41	23	37
時間／net(s)	1.3	1.4	0.7	2.5
時間／step(ms)	26	27	24	26
ラベル付け／step	2.5	3.1	2.0	2.7
稼働率(%)	45	53	50	44

また，表３－２に，図３－２(a)～(d)の配線例についての各測定値を示した．それぞれ，平

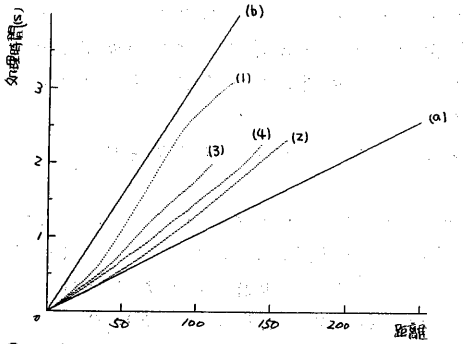


図３－１　フェーズＰの処理時間（途中経過）

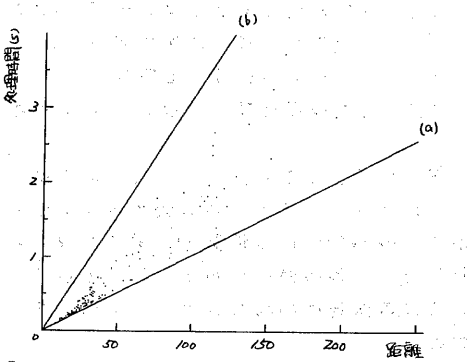


図３－３　フェーズＰの処理時間

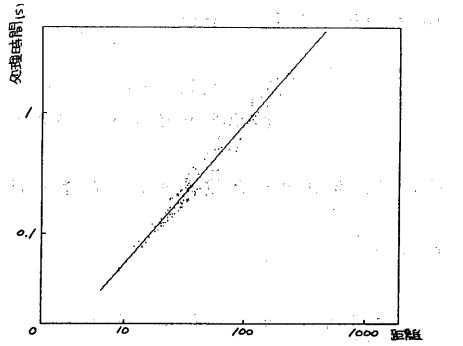
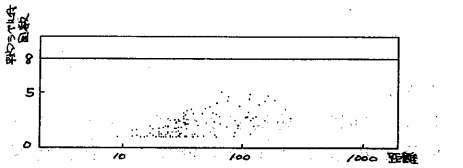
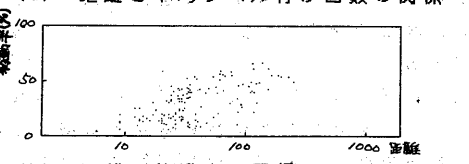


図３－４　距離と処理時間の関係



(a)　距離と平均ラベル付け回数の関係



(b)　距離と稼働率の関係

図３－５　距離と平均ラベル付け回数，稼働率の関係

均配線長，ブロック数などが異なるものであるにもかかわらず，配線時間は距離に比例して，約26ms/stepという結果を得た．

また，稼動率は，ほぼ50％と，セル割当てが効果的に働いていることを示している．また，平均ラベル付け回数（１stepあたりのラベル付け回数）が平均２～３回と，小さい値におさまっているのも良い結果で，これは，同じ構成の１セル１プロセッサ型のものと比べて（プロセッサ数は 1/256であるにもかかわらず）２～３倍の処理速度ですむことを示している．

また，以上の結果により，もっとセル数の多い場合についても，一般的に，

「Ｎ＊Ｎセルの配線領域に対して，適当なセル割当てを用いてＮ／２個のプロセッサで並列処理させれば，１本の径路を求める時間は，１セル１プロセッサ型のそれに対して，（プロセッサ数は１／２Ｎであるにもかかわらず）高々８倍，平均して２～３倍におさまる．つまりＯ（Ｎ）の時間で処理できる．」

と言うことができる．

（４）おわりに

今回は，汎用のマイクロプロセッサを用いて実際にＲＰを試作し，Ｎ＊Ｎの規模のルーティングは，Ｏ（Ｎ）個のプロセッサの並列処理により，Ｏ（Ｎ）の時間で処理できることを示した．次のステップとしては，ＲＰ自身をLSI 化して，より高速の，実用規模の処理をおこなわせることを考えている．

［参考文献］

1) 橘，大附：”配線処理のハードウェア化に関する一考察”，信学技報 CAS81-108, pp.99-104, (1982).
2) 橘，中島，大附：”並列ルーティング・プロセッサの一構成法”，通学技報 CAS83-75, pp.25-30 (1983).
3) A.Losupouicsz : "Design of an Iterative array Maze Router", Proc. of ICCC, pp.908-911 (1980).
4) M.A.Breuer, K.Shamsa : "A Hardware Router", J. of Digital Systems, Vol.4, pp.393-408 (1981).
5) T.Blank, M.Stfik, W.vanCleempt : "A Parallel Bit Map Architectur for DA algorithms", 18th DA Conf., pp.837-845 (1981).
6) S.J.Hong, R.Nair, E.Shapiro : "A Physical Design Machine" in VLSI81, Academic press, pp.257-266 (1981).
7) P.Agrawal, M.A.Breuer : "Some Theorecical aspects of algorithmic Routing", 14th DA Conf., pp.23-32 (1977).

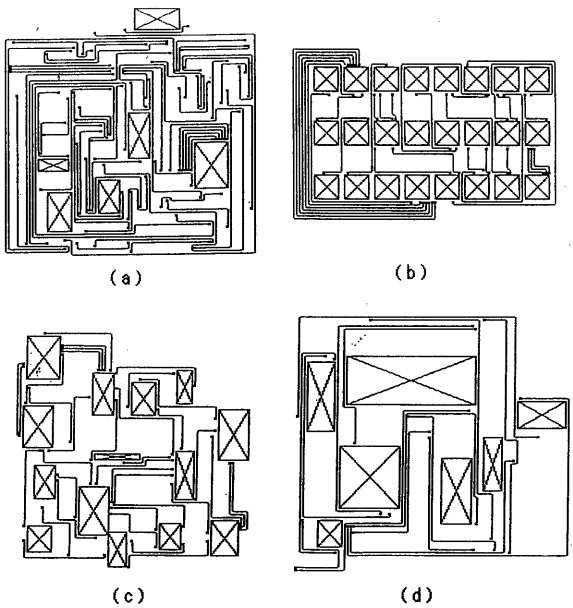


図３－２ 配線例