

An Efficient Paired-net Routing Algorithm for High-speed Bipolar LSIs

Y. Takei, A. Onozawa[†], K. Kawai^{††} and H. Kitazawa
NTT System Electronics Laboratories

[†]NTT Multimedia Networks Laboratories

^{††}NTT Optical Networks Laboratories

3-1, Morinosato Wakamiya, Atsugi-Shi, Kanagawa, 243-01 Japan

E-mail: take@aecl.ntt.jp

Abstract

This paper proposes an efficient paired-net routing algorithm. A paired net consists of two nets assigned to adjacent tracks and columns. The paired-net routing approach is applied for differential drive nets in a bipolar ECL LSI to decrease signal skew and enhance signal integrity. Paired nets can cause a new type of cyclic constraint, which does not appear in the conventional vertical constraint graph. To wire these paired nets, the new type of cyclic constraint must be detected and resolved. We applied our approach to two actual bipolar ECL circuits: a 4.0-k-gate circuit, which operated at 3 Gbit/s, and a 5.6-k-gate circuit, which operated at 2.5 Gbit/s. The experimental results verify the effectiveness of our paired-net routing approach.

1. Introduction

As VLSI fabrication technology advances, interconnection technology has become a bottleneck what is adversely affecting LSI performance. In a high-speed, bipolar ECL LSI, differential drive nets (Fig. 1) and other nets are routed independently. This is because low skew and high signal integrity in differential drive nets are critical to chip performance. The paired-net routing approach shown in Fig. 1 is applied to the routing of differential drive nets in a bipolar ECL LSI to decrease signal skew and enhance signal integrity.

A paired net consists of two nets, A and B, assigned to adjacent tracks and columns. Crosstalk between the two nets is negligible because spacing between the adjacent tracks/columns is adjusted before routing to avoid coupling capacitance between interconnection wires in a bipolar ECL LSI.

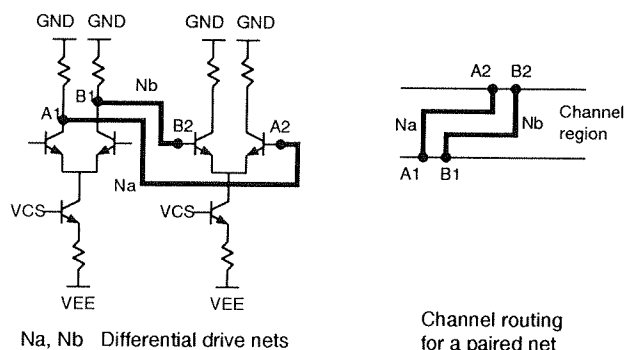


Figure 1. Differential drive nets.

A paired net possesses two advantages. One is that the two nets composing a paired net are equal in length and are routed along the same path. The other is that the two nets are equal in their number of crossings. This means that the number of crossings between net A and nets other than net B, and the number of crossings between net B and nets other than net A, are equal as shown in Fig. 2(a).

Equality in length is important for reducing signal skew in differential drive nets in a bipolar ECL LSI. It is clear that paired-net routing makes two nets composing a pair of differential drive nets equal in length.

It is also important to ensure there are an equal number of crossings in the nets. This is because if the number of crossings is not equal, there will be a difference in the coupling capacitance between the two nets, which will increase the signal skew.

With paired-net routing, equality in the number of crossings yielded by differential drive nets can be accomplished. Figure 2(a) shows an example of equality in the number of crossings obtained with paired-net

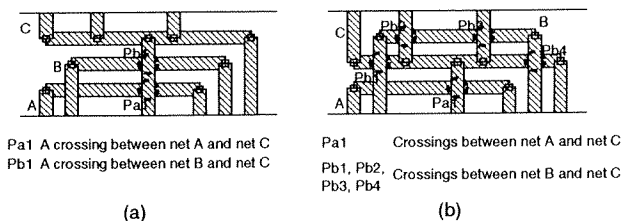


Figure 2. Crossings between a pair of differential drive nets and another net. A pair of differential drive nets, A and B routed (a) with paired-net routing and (b) without paired-net routing.

routing and Fig. 2(b) shows an example of how the number of crossings becomes unequal when paired-net routing is not applied.

In Figs. 2(a) and (b), nets A and B compose a pair of differential drive nets. Net C does not compose the differential drive net. In Fig. 2(a), pair A and B are applied with paired-net routing. One crossing P_{a1} results between net A and net C and one crossing P_{b1} results between net B and net C. The number of crossings between the differential drive net and another net is equal. In Fig. 2(b), pair A and B are not applied with paired-net routing. One crossing P_{a1} results between net A and net C and four crossings $P_{b1}, P_{b2}, P_{b3}, P_{b4}$ result between net B and net C. The number of crossings are not equal. Figure 2 illustrates how that paired-net routing can accomplish equality in the number of crossings between a differential drive net and another net.

Paired nets are necessary in the design of bipolar ECL LSIs that operate at the GHz level. In traditional routing, differential drive nets are modified into paired nets manually after all the nets have been routed automatically. This is because it is impossible to wire paired nets using existing routers. However, recent increases in chip size [1]-[4] have made such manual routing impossible.

This paper proposes an efficient paired-net routing algorithm that wires the paired nets and the other nets in the same channel simultaneously. The algorithm is based on grid-based channel routing. Also presented are the results of applying our proposed approach to two actual ECL LSIs.

2 Routing Paired Nets

In what follows we describe our paired-net routing approach in detail. Let H and V be respectively the

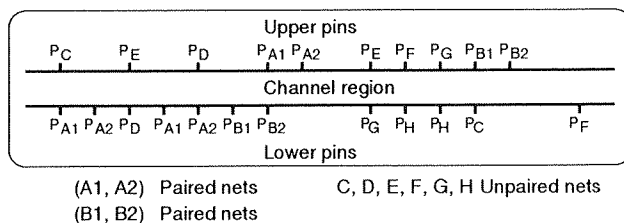


Figure 3. Example of pin placement including pins of paired nets.

number of tracks and columns in the channel routing area. These tracks are indexed 1 through H/V from the top/left. Let $i, (1 \leq i \leq H - 1)$ index a track and $j, (1 \leq j \leq V - 1)$ index a column. Adjacent horizontal wire segments of the paired nets are assigned to tracks $i, i + 1$ and adjacent vertical wire segments of the paired nets are assigned to columns $j, j + 1$.

We call a net assigned to track i an upper net, and a net assigned to track $i + 1$ a lower net. All pins connected with the paired nets are assumed to be located at adjacent columns.

2.1 Paired nets and vertical constraint graph

Figure 3 shows an example of pin placement including pins of differential drive nets. P_X ($X = A_1, A_2, B_1, B_2, C, \dots, H$) represents a pin. The pins are at both the top and bottom of a channel. The value of X represents a net that connect pins. In these nets, there are two pairs of differential drive nets, (A_1, A_2) and (B_1, B_2) , which are applied with paired-net routing. All other nets are not applied with paired-net routing. These nets are called unpaired nets.

No two nets can occupy the same part of a column. Therefore, if it is assumed that there is only one horizontal segment per net, then it is clear that the horizontal segment of a net connected to the upper pin of a particular column must be positioned above the horizontal segment of a net connected to the lower pin of that column. This relation is a vertical constraint.

Vertical constraints are represented by a directed graph called a vertical constraint graph [5]. The vertical constraint graph (VCG) for the pin placement in Fig. 3 is shown in Fig. 4. Each circle denotes a node that represents a net. (We use the terms “node” and “net” interchangeably in this paper.) Each arrow denotes a directed edge that represents a vertical constraint between two nets. A directed edge from net a to net b means that net a must be placed above net

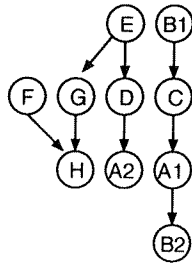


Figure 4. Conventional vertical constraint graph generated by the pin placement in Fig. 2.

b. In this paper, we call such a VCG a **conventional VCG** to distinguish it from a modified VCG, which will be described later.

In a conventional VCG, there is a possibility that some vertical constraints may form a loop, a condition called a cyclic constraint [5][6]. An example of a cyclic constraint is $a \rightarrow b \rightarrow a$. This means that net a must be placed above itself. Therefore, if there is a cyclic constraint in a conventional VCG, the routing requirement cannot be achieved without dividing some nets.

In Fig. 4, no cyclic constraints exist. If there are no paired nets in the given routing requirements, it is possible to wire all nets completely using traditional channel routing.

Using traditional channel routing, it is not, however, possible to wire paired nets (A_1, A_2) and (B_1, B_2) . This is because these paired nets cause a new type of cyclic constraint, which does not appear in the conventional VCG shown in Fig. 4. To wire these paired nets, a new type of cyclic constraint must be detected and resolved.

To create the new type of cyclic constraint, we modify the conventional VCG. The procedure for modifying the conventional VCG is as follows. First, pairs of nodes representing paired nets in the conventional VCG are merged. This is done because we want to route each paired net as if it were one single net. Figure 5(a) shows the procedure for merging two nodes of the paired nets (A_1, A_2) to A and (B_1, B_2) to B . A and B are called virtual nodes. We call such a modified VCG a **PN-VCG**.

2.2 Resolving the new type of cyclic constraint

Let us examine the cyclic constraints in a PN-VCG. In Fig. 5(b), we can detect the cyclic constraint $C \rightarrow A \rightarrow B \rightarrow C$ by decomposing the PN-VCG into strongly connected components. Note that this cyclic constraint is different from the one that appears with

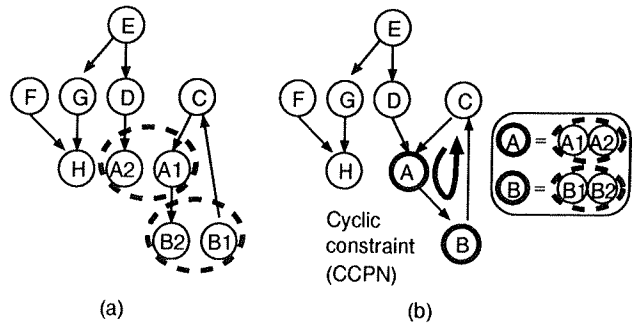


Figure 5. Modified vertical constraint graph (merging two nodes of paired nets).

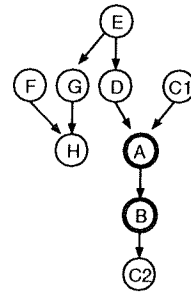


Figure 6. Vertical constraint graph after resolution of the CCPN.

the conventional VCG. This cyclic constraint is a product of the paired nets, and we call it a **CCPN** (cyclic constraint for paired nets). A CCPN is defined as a cyclic constraint that consists of virtual nodes.

To connect all pins with the paired nets in Fig. 3, the CCPN and the other cyclic constraints must be resolved by splitting the horizontal wire segments of a net.

We resolve the CCPN by splitting the horizontal wire segments of an unpaired net that is included in the CCPN. The number of additional vias required for splitting the paired nets is twice that required for splitting an unpaired net. From the electrical standpoint then, it is desirable to limit the number of vias, because they increase capacitance. The PN-VCG obtained after resolving the CCPN is shown in Fig. 6.

2.3 Track assignment

After the CCPN is resolved, an upper net and a lower net must be determined. The vertical order of these nets can directly cause vertical and horizontal wire segments to cross. At such crossings, the capac-

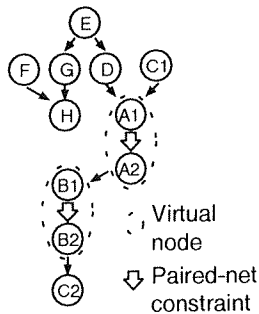


Figure 7. Paired-net constraint.

itance between the two metallic wires becomes large. We specify the vertical relation of a pair of wiring nets so as to minimize the number of crossings, and thereby the capacitance, between two metallic wires.

The vertical relation between an upper and lower net is represented as a constraint in the PN-VCG. We call this constraint a **paired-net constraint**. The paired-net constraint does not forbid overlapping of vertical wire segments for distinct nets; it only represents the vertical relation between two nets that compose the paired net. In Fig. 7, paired-net constraints are shown by an unfilled arrow. In the PN-VCG, a paired-net constraint is directed from the node of an upper net to the node of a lower net.

After an upper and lower net have been decided upon, all nets are assigned to tracks in the channel. Existing channel routing algorithms (for example, Refs. [5], [6], and [7]) cannot be used to assign the paired nets. We modified the left-edge algorithm [7]. The modified parts in the assignment procedure are as follows:

```

Procedure track_assignment(i) ;
begin
01:   while Assignment of track i is possible do
       begin
02:     /* Select a net */
       Select a node N without incoming edges
       in PN-VCG that does not overlap
       with the other nets assigned to track i
       /* Assign the paired nets */
03:     if N is a virtual node then
           begin
04:       Assign the upper net of N
           to track i ;
05:       Assign the lower net of N
           to track i + 1 ;
           end;
       /* Assign an unpaired net */
06:     else Assign N to i ;
       end

```

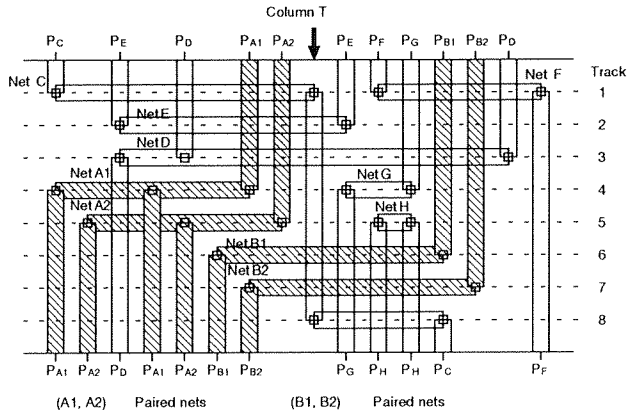


Figure 8. Routing results.

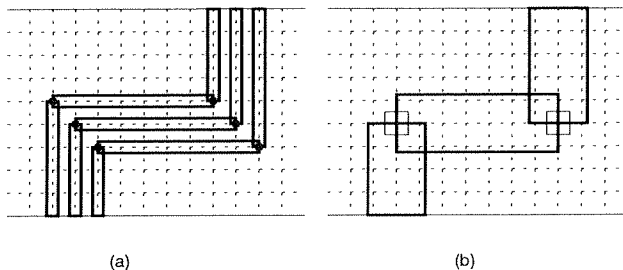


Figure 9. (a) Bundle and (b) variable-width net.

```

end;
end;

```

Figure 8 shows a routing result. In Fig. 8, (A_1, A_2) and (B_1, B_2), the hatched patterns, are the paired nets. These nets are confirmed as being assigned to adjacent tracks. A horizontal segment of net C, which is an unpaired net, is split into two parts at column *T* to resolve the CCPN.

With a little modification, our paired-net routing algorithm can be applied to the routing of a bundle, which consists of more than three nets, and a variable-width net. Examples of both are depicted in Fig. 9.

3 Experimental Results

Our paired-net routing algorithm was implemented in C language on a SparcStation 20. We applied it to the routing of two actual LSIs, which we refer to as the F1 chip and the F2 chip.

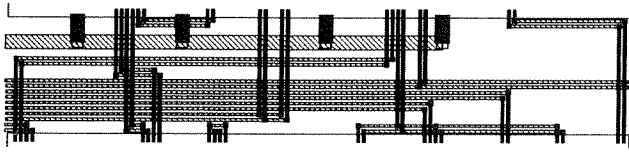


Figure 10. Routing results with paired nets in a portion of the F1 chip (Layout 1).

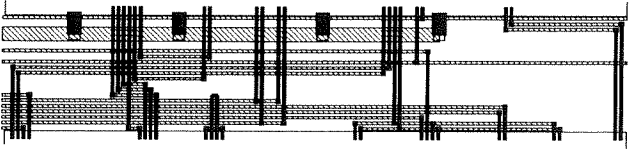


Figure 11. Routing results without paired nets in a portion of the F1 chip (Layout 2).

3.1 F1 chip

The F1 chip is a 4.0-k-gate bipolar ECL circuit and consists of 2059 nets. It is comprised of 1016 pairs of differential drive nets (2032 nets) and three variable-width nets.

We produced two layouts of the F1 chip to evaluate our paired-net routing approach. Layout 1, which corresponds to a real circuit, is the routing that resulted by applying paired-net routing to all the differential drive nets. Layout 2 is the routing result without paired-net routing. Both layouts involve variable-width nets. The cell placement and global routing, which is performed by *TAGROUTE* [8], are used for both layouts. Figures 10 and 11 show the routing results for Layout 1 and Layout 2 respectively in a portion of the F1 chip.

We examined the inequality in length of the two nets comprising the differential drive nets and the inequality in the number of crossings yielded by the differential drive nets in one channel of the F1 chip. Both of these greatly influence signal skew and integrity.

The experimental results in Table 1 show that with our paired-net routing approach, the maximum inequality in length decreased by 86% and the maximum inequality in the number of crossings decreased to 0. There was a slight increase in CPU time, total net length, and chip size. Our experimental results substantiate the effectiveness of our paired-net routing algorithm. The F1 chip achieved 3-GHz operation [4].

	Layout 1	Layout 2
max. inequality in length %	21 μm 14	147 μm 100
max. inequality in the number of crossings %	0 0	16 100
CPU time %	43.23 sec 108	40.17 sec 100
total net length %	1301.71 mm 100.3	1297.36 mm 100
area %	12.53 mm^2 102	12.34 mm^2 100

Table 1. Experimental results

3.2 F2 chip

The F2 chip is a 5.6-k-gate bipolar ECL circuit and consists of 1184 nets. The F2 chip is composed of 82 pairs of differential drive nets (164 nets) and 1020 unpaired nets.

The F2 chip achieved 2.5-GHz operation [3]. The entire chip layout of the F2 chip is shown in Fig. 12. We applied our proposed approach to the differential clock nets of the F2 chip.

4 Conclusion

We have proposed an efficient paired-net routing algorithm. We applied it to two ECL circuits: a 4.0-k-gate circuit, which operated at 3 Gbit/s, and a 5.6-k-gate circuit, which operated at 2.5 Gbit/s. Our experimental results verify the effectiveness of our paired-net routing algorithm.

References

- [1] K. Koike, K. Kawai, and H. Ichino, "A design methodology of bipolar standard cell LSIs for Gbit/s signal processing," 1993 BCTM, pp. 236-239.
- [2] K. Koike, K. Kawai, A. Onozawa, Y. Kobayashi, and H. Ichino, "Low-power design methodology for Gbit/s bipolar LSIs", 1995 BCTM, pp. 106-109.
- [3] K. Kawai, K. Koike, H. Ichino, and Y. Kobayashi, "High speed regenerator-section terminating LSI

operating up to 2.5 Gbit/s using 0.5 μ m Si bipolar standard-cell technology”, Electronics Letters 10th May 1995 Vol. 31, No. 10, pp. 791-792.

- [4] K. Kawai, K. Koike, Y. Takei, A. Onozawa, H. Obara and H. Ichino, “2.5Gb/s 557mW STM-16 Regenerator Section Terminating LSI Chip”, ISSCC’97, to be presented.
- [5] T. Yoshimura and E.S. Kuh, “Efficient algorithms for channel routing,” IEEE Trans. CAD, Vol. CAD-1, no. 1, pp. 25-35, Jan. 1982.
- [6] M. Burnstein, “Channel Routing, Layout Design and Verification,” T. Ohtsuki (Ed.) North-Holland, pp. 133-167, 1986.

- [7] A. Hashimoto and S. Stevens, “Wire routing by optimizing channel assignment within large apertures,” in Proc. 8th Design Automation Workshop, pp. 155-169, 1971.
- [8] I. Harada and H. Kitazawa, “A global router optimizing timing and area for high-speed bipolar LSIs,” Proc. of DAC, pp. 177-181, 1994.

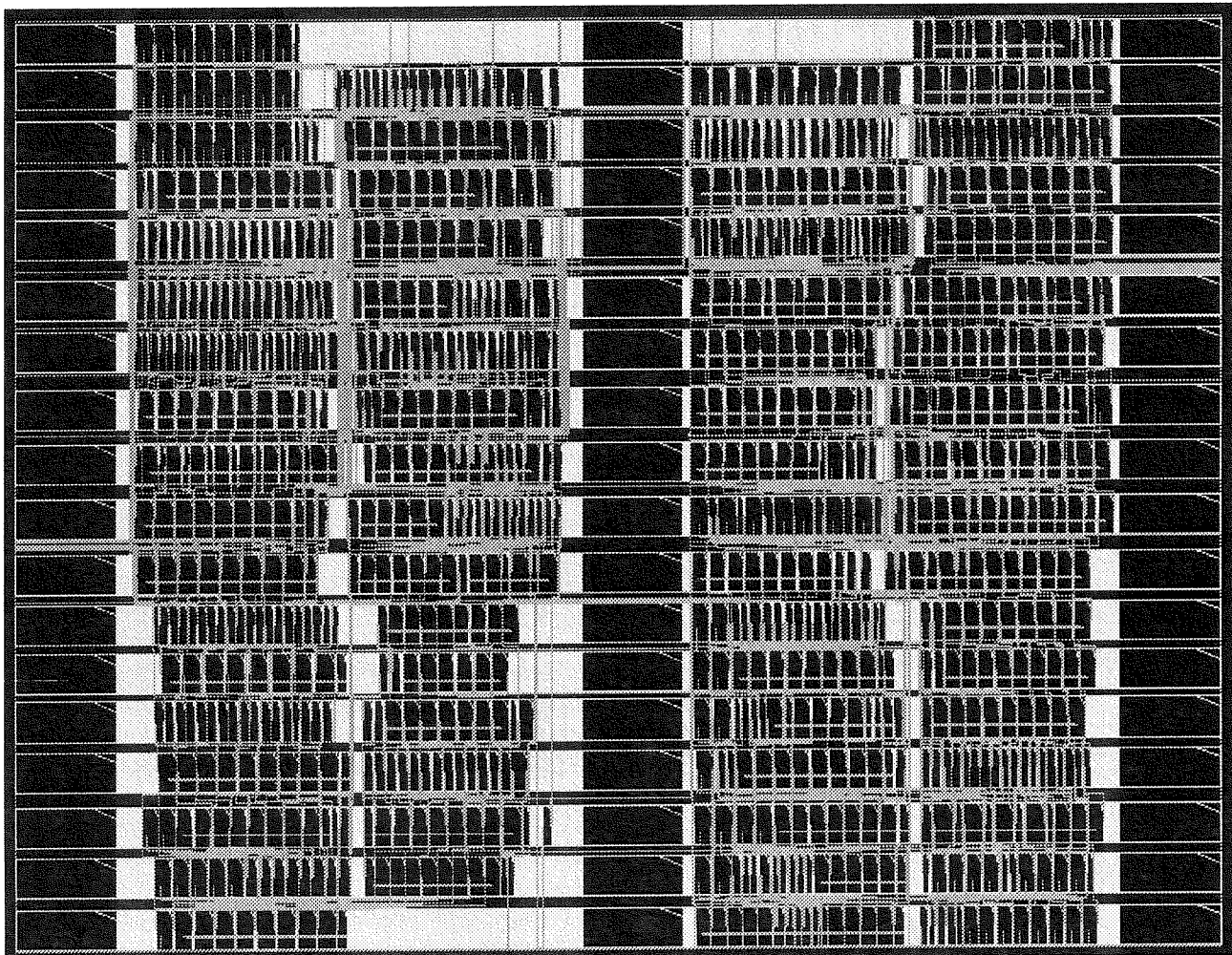


Figure 12. The entire chip layout pattern of the F2 chip.