

Analog Layout Using ALAS!

J. D. Bruce, H. W. Li, M. J. Dallabetta, and R. J. Baker

Abstract—An analog layout assistant (ALAS!) is presented that automatically generates common-centroid, interdigitated device pairs and passive components. The user inputs a minimum of input parameters and has the ability to interactively alter the parameters of the program. The program can be used in conjunction with any layout editor that imports CalTech Intermediate Format (CIF) layouts and is platform and technology independent.

I. INTRODUCTION

ANALOG integrated circuits are typically highly sensitive to process variations. Interdigitation and common-centroid layout can make key analog components less sensitive to process variation, but requires complicated and time consuming full-custom layout. This paper presents an analog layout assistant (ALAS!) that automatically generates matched and balanced CMOS components with minimal input parameters from the user. There are CAD programs [1], [2] that layout a transistor given the width and length, but these are typically embedded within large synthesis or layout tools which allow the user very little flexibility for critical components.

The designer is allowed to specify the topology of stacked transistor configurations [3] used in common-centroid, interdigitated arrangements. The program inputs are the transistor's width, length, and process parameters. The user has the ability to adjust the size of the transistor layout by specifying the aspect ratio and the minimum and maximum width of each parallel device (the adjustment window), thus giving the designer a wide flexibility in the overall layout of the components. ALAS! has been tested and verified to be compatible with a variety of both PC and workstation layout tools and is portable to any scalable CMOS process. Currently, the program is capable of performing common-centroid, interdigitated layout for differential pair, common source current mirrors, floating source current mirrors, single transistors, capacitor pairs, and resistor pairs.

II. ACTIVE DEVICES

ALAS! consists of two main functions: an optimization algorithm and place and route algorithm.

Manuscript received July 5, 1995; revised September 8, 1995. This work was supported in part by the University of Idaho Microelectronics Research Center.

J. D. Bruce is with Micron Semiconductor, Boise, ID 83706 USA.

H. W. Li is with Department of Electrical Engineering, University of Idaho, Moscow, ID 83844-1023 USA.

M. J. Dallabetta is with Hewlett Packard, Greeley, CO 89631 USA.

R. J. Baker is with the Department of Electrical Engineering, UI-BSU Engineering, Boise, ID USA.

Publisher Item Identifier S 0018-9200(96)01292-9.

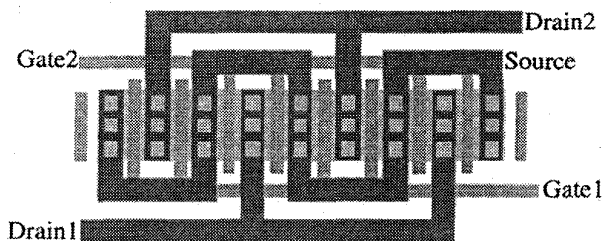


Fig. 1. Two interdigitated transistors (1 row, 8 columns).

```
> generate 144 2 144 2 2 1.4 10 20
One Lambda = 1.00u.
There are 3 row(s) with 8 devices per row.
The widths of each row are:
row 1 => width(L) = 12 lambda width(S) = 12 lambda.
row 2 => width(L) = 12 lambda width(S) = 12 lambda.
row 3 => width(L) = 12 lambda width(S) = 12 lambda.
The number of devices to be cut by one lambda in the bottom row are: 0 & 0
The larger transistor size (in lambda) without cutting the bottom devices is 144.0
The smaller transistor size (in lambda) without cutting the bottom devices is 144.0
The best fit aspect ratio is 1.33
The actual aspect ratio is 1.33
```

Fig. 2. Example of the row-column analysis.

A. Optimization

The layout can be considered as a matrix, with each element corresponding to a parallel device. Fig. 1 illustrates an interdigitated pair corresponding to a 1×8 matrix. Notice the dummy poly strips on the ends of the rows used to reduce the undercutting of the gate oxide on the end devices [4].

An optimization algorithm is used to determine the number of rows and columns required for the matrix. The algorithm is based on the following equation that determines the aspect ratio

$$Ar = \frac{\sum_{i=1}^R w_i + R \cdot K_3}{C \cdot K_1 + K_2} \quad (1)$$

where Ar is the aspect ratio, w_i is the row device width, R is number of rows, C is the number of columns, K_1 is a function of the transistor length, K_2 corresponds to the space for vertical routing, and K_3 is the space for horizontal routing.

Also,

$$C \sum_{i=1}^R w_i = 2 \cdot W_{total} \quad (2)$$

where W_{total} is the total width of the largest transistor.

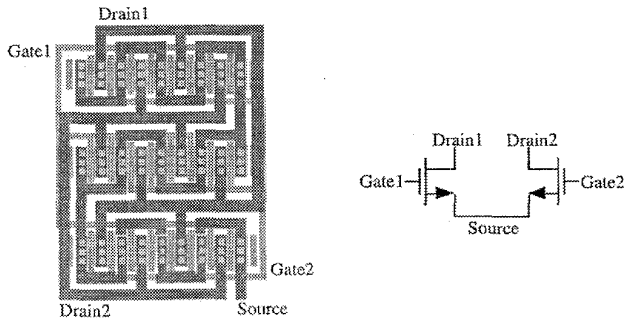


Fig. 3. Interdigitated differential input pair layout.

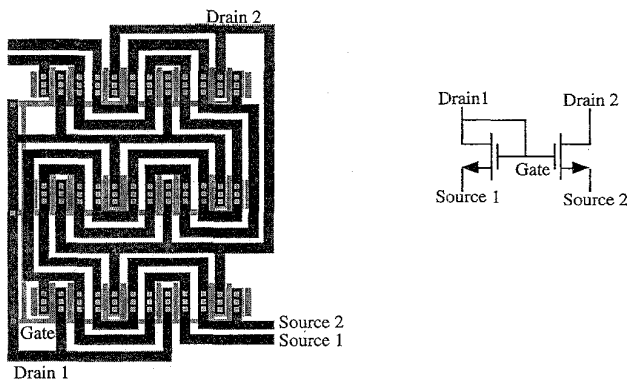


Fig. 4. Stacked current mirror pair.

Substituting (2) into (1) yields the final equation for the aspect ratio

$$Ar = \frac{2 \cdot W_{total} + R \cdot C \cdot K_3}{C^2 \cdot K_1 + C \cdot K_2} \quad (3)$$

R (row) and C (column) can then be determined using an optimization algorithm based on the requirements that the number of rows is odd and the number of columns is even and both are integers. The number of rows is required to be an odd integer so that the drain metal routing is symmetrical about a midpoint. This ensures complete parasitic balance between the two transistors. If the rows are to be kept odd, then the columns have to be kept even, to also insure symmetrical routing and balanced parasitics for both devices. These requirements also significantly reduce the search space for the optimization algorithm. Thus the row-column analysis requires the following input from the user

```
GENERATE(W1)\(L1)\(W2)\(L2)
      (process)\(aspect)\(device min)\(device max).
```

$W1$ and $W2$ are the transistor widths, $L1$ and $L2$ are the transistor lengths. For layout of single devices, $W2$ and $L2$ are omitted. The input variable, *process*, corresponds to the minimum feature size for the user's particular process (in microns) and the *aspect* parameter represents the desired aspect ratio, allowing the user to define the overall shape of the layout. The parameters, *device min* and *max*, are the minimum

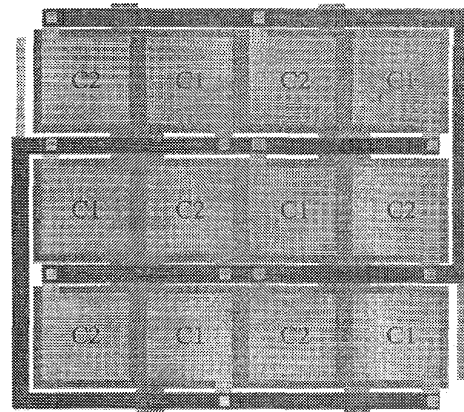


Fig. 5. Matched capacitor pair.

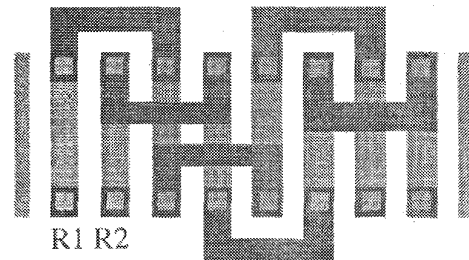


Fig. 6. Matched resistor pair.

and maximum allowed width of each parallel device. These parameters enable the user to consider process gradients in the layout and also allows adjustment in achieving the desired aspect ratio. If these last two parameters are omitted, then ALAS! will assume a nonstacked layout topology. This is convenient for small devices.

A report is generated upon each execution which tells the user the layout characteristics (Fig. 2 illustrates an example report for a differential pair). The user can then readjust the parameters before generating the actual layout. The diff-amp used in Fig. 2 contains three rows each containing eight parallel devices. The user is also informed of how many parallel devices are to be fine cut to meet the desired width. Since each row has a common width for each transistor, if the summation of these widths do not equal the user's defined width, a number of the bottom row's devices will be cut by the necessary amount. The optimization routine determines the best fit aspect ratio. The actual aspect ratio is given to account for extra contacts that might be needed. If the aspect ratio does not meet the user's needs, the user may alter the adjustment window parameters, resulting in a more exact fit.

B. Place and Route

The second function of ALAS! creates the CIF file for the user. The placement and routing program reads a data file passed from the row/column analysis. This data file can be

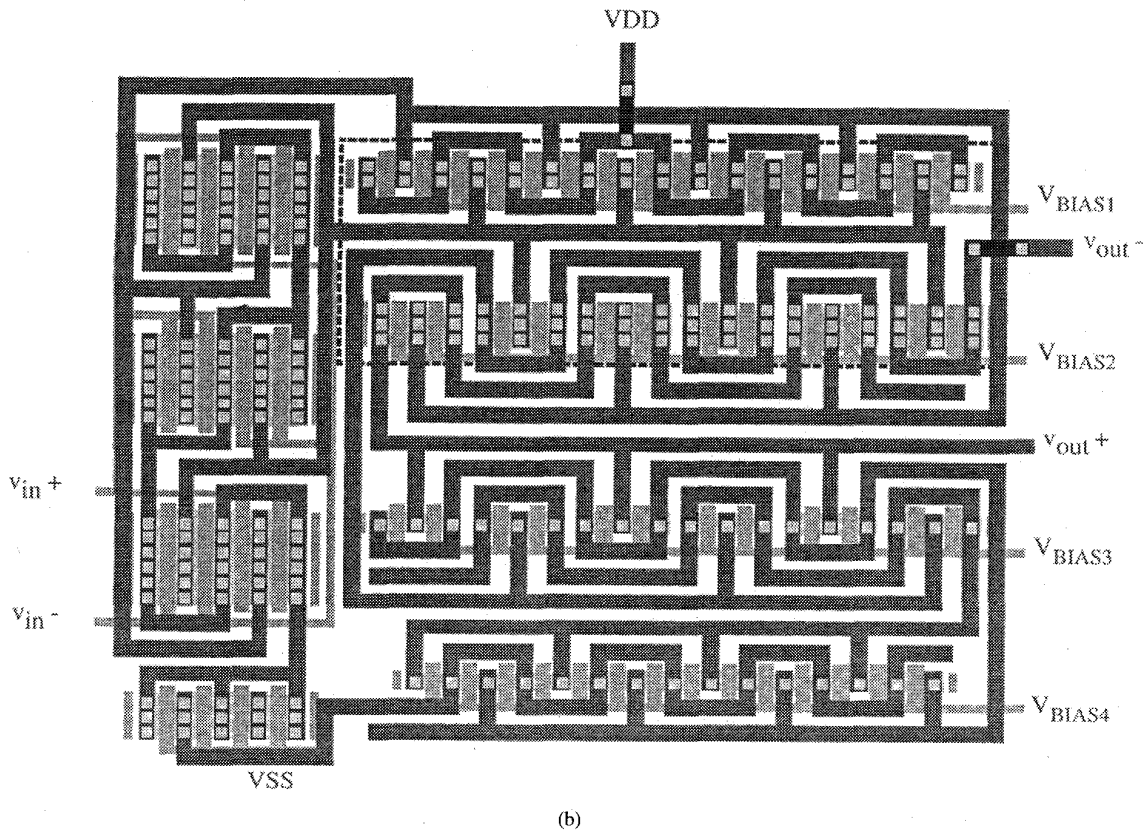
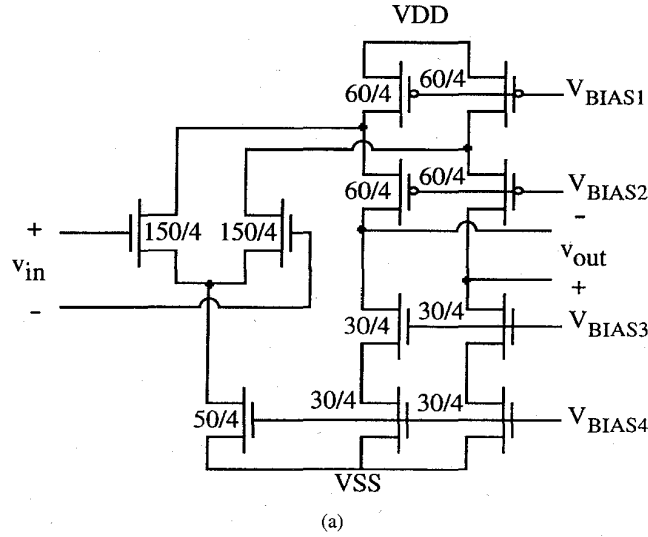


Fig. 7. (a) Fully-differential cascode amplifier and (b) layout generated by ALAS!

edited by the user if customization of the number of rows and columns is desired. The drawing and routing programs utilize the fact that the interdigitated common-centroid layouts have a regular pattern and are very symmetrical. Fig. 3 illustrates the layout of the differential input pair example in Fig. 2. Note that each device has the exact same metal and poly routing and that the layout is completely symmetrical about the center of the circuit.

Some designs require a current mirror that has two separate drains and sources (such as a cascode amplifier) as seen in Fig. 4. The same algorithms used for the common node topologies are used, the difference being the values of K_1 , K_2 , and K_3 . This layout scheme ensures that both sources and drains have the same metal routing. ALAS! is capable of generating layouts for differential amplifiers, current mirrors, floating current mirrors, and parallel single devices.

III. PASSIVE COMPONENTS

ALAS! will also create a CIF file for interdigitated capacitors and resistors.

A. Capacitor Optimization

As with the active devices, the capacitor interdigitation is essentially an array. Each element of the array is a unit capacitor. The optimization algorithm is very similar to the active device algorithm and uses the following equations:

$$\text{Ar} = \frac{\sum_{i=1}^R W_i + R \cdot K_1}{C \cdot [L + K_2] + K_3} \quad (4)$$

and

$$W = \sqrt{\frac{C_U}{C_A}} \quad (5)$$

and

$$C \cdot \sum_{i=1}^R C_{U_i} = 2 \cdot C_{\text{total}} \quad (6)$$

where Ar is the aspect ratio, W_i is the row device width, R is number of rows, C is the number of columns, C_U = unit capacitance, C_A = capacitance/m², C_{total} = total capacitance desired, and $K_{1,2,3}$ = layout constants.

Making C_U square will minimize process gradient effects [4]. Thus, with $L = W$ and using (4) with (5), an optimization algorithm very similar to the one used for the active devices is performed. Fig. 5 shows a layout of a pair of matched capacitors. Notice how the metal routing is identical for both capacitors.

B. Resistor Optimization

The resistor array consists of only one row because the resistance is typically a function of length. The equation to determine the aspect ratio is

$$\text{Ar} = \frac{L + K_2}{C[W + K_1] + K_3} \quad (7)$$

The total resistance can be found by

$$2 \cdot R_{\text{total}} = C \cdot \left[2 \cdot R_C + \left(\frac{L}{W} + 1 \right) \cdot R_S \right] \quad (8)$$

where R_C = resistance of the metal to poly contacts and R_S = sheet resistance of the poly layer. The last parameter signifies if the resistor is a pair or a single component. Fig. 6 illustrates an automatically generated resistor pair.

IV. EXAMPLE

A fully differential folded cascode amplifier was laid out using ALAS! and is seen in Fig. 7. The entire layout was completed in less than 20 minutes in conjunction with an industry standard CAD tool.

V. CONCLUSION

Written in C on an HP series 700 workstation running HP-UX, the program currently generates neutral active areas, since some processes require separate N -select or P -select layers. Eventually, ALAS! will be expanded to accommodate either type of process automatically.

Capabilities will continue to be updated to include guard rings and binary weighted capacitor arrays. The amount of time required to layout any circuit that uses charge storage components will be greatly reduced.

REFERENCES

- [1] J. Cohn, D. Garrod, R. Rutenbar, and L. Carley, "KOAN/ANAGRAM II: New tools for device-level analog placement and routing," *IEEE J. Solid-State Circuits*, vol. 26, no. 3, pp. 330-342, Mar. 1991.
- [2] B. Owen, "The intermediate layout language," Ref. Doc., Univ. of Toronto, Nov. 1994.
- [3] U. Gatti, F. Maloberti, and V. Liberali, "Full stacked layout of analogue cells," in *Proc. IEEE Int. Symp. on Circuits and Systems*, May 1989, pp. 1123-1126.
- [4] M. Ismail and T. Fiez, *Analog VLSI: Signal and Information Processing*. New York, McGraw-Hill, ch. 15, 16, 1994.