

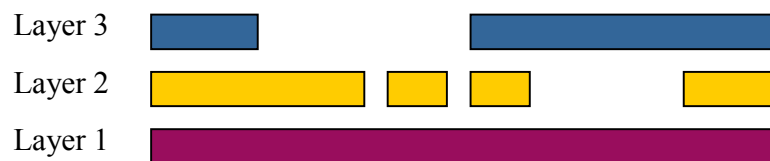
A Method of Identifying Analog Cell Mismatch Issues Caused by Photo Mask Misalignment

David Schwan
Manager: CAD and Layout
Micro Linear Corporation
2050 Concourse Drive
San Jose, CA 95131
schwan.david@microlinear.com

Most analog circuits are susceptible to mismatch. One source of the mismatch is the misalignment of layers on a wafer, e.g. between poly mask and active layers. For two MOS transistors placed as mirrored devices in the layout, if the poly misaligns perpendicular to the gate endcaps, the source and drains areas become unequal. These misalignments are acceptable for digital circuits, but can cause yield problems in analog circuits. Other sources of mismatch can be misalignment between Metal-1 and Poly, or between Metal-2 and Metal-1. Creating a layout that is insensitive to this form of mismatch is important to good yield. Existing physical verification flows do not catch mismatch errors due to mask misalignment. There are five possible combinations of any two layers, center, up, down, left and right. A flow will be presented using Assura RCX to simulate the mask misalignments using the geomShift command. A script creates each of the desired layer pair misalignments followed by parasitic capacitance extraction to generate a spice netlist. A PERL script then looks at each node of each run and compares the base capacitance value for the net with the shifted mask capacitance. If the values differ, that node is marked as having a mask alignment sensitivity.

The Problem

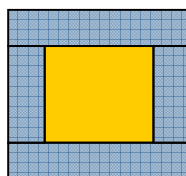
The IC fabrication process uses photomasks for patterning each layer of the IC. These photomasks are aligned either to a “Zero Layer” (newer processes) or to the previous layer (older processes and fabs, large feature size). This alignment is subject to variation in X and Y directions, and will obey normal statistical distributions. These variations can cause mismatches in analog circuits. Since each layer has independent alignment the total offset or misalignment will be the sum of the two layers of interest. Figure 1 shows the vertical stacking of layers.



Vertical stacking of Layers

Figure 1

Perfect Alignment



Allowed Misalignment

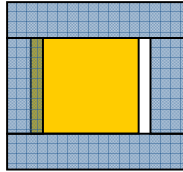


Figure 3

Figure 2 shows ideal alignment of two layers, and Figure 3 shows the allowable misalignment of two layers.

High performance analog systems use differential signals. Differential signal paths are used because they offer high noise immunity because their balanced nature cancels out noise. Differential signal paths can be poorly designed, and subsequently can show worse performance than a non-differential signal path. This poorer performance is due to poor matching, which creates mismatches. Mismatching lowers overall system performance. This mismatch can be of many forms, the two we are going to look at here are transistors and interconnect.

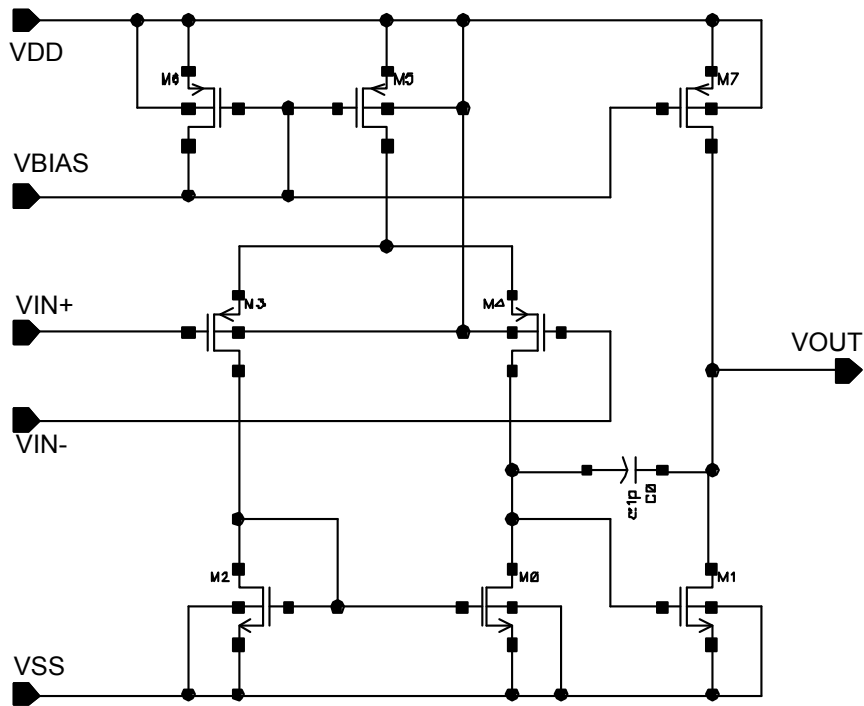


Figure 4

In Figure 4 a simple differential input op-amp is shown. There are three places we will be concerned with for matching. The input transistors M3 and M4 are the most important devices in the circuit, and need to be matched. The bias transistors M2 and M0 need to be matched. Lastly the interconnect on nets VIN+ and VIN- need to be matched.

Figure 5 shows the input differential pair placed in a mirrored layout. The important layers to look at here are active and poly. In Figure 6 the poly mask has been shifted to the right. Note that the outside source drain areas do not have the same area. The left source became larger and the right source got smaller. The drains have an equivalent shift. Mirrored devices are not good candidates for matched circuits.

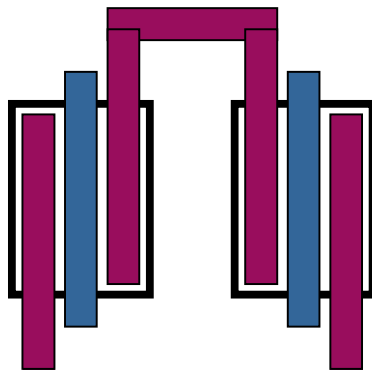


Figure 5

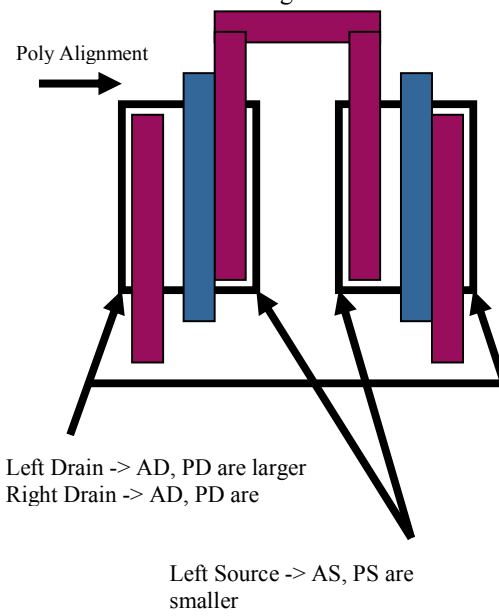


Figure 6

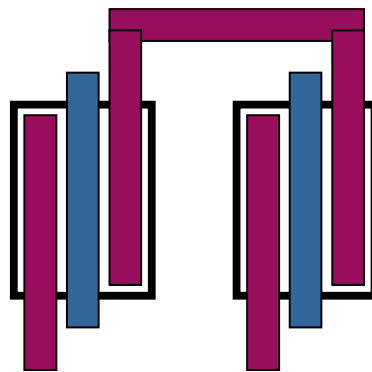


Figure 7

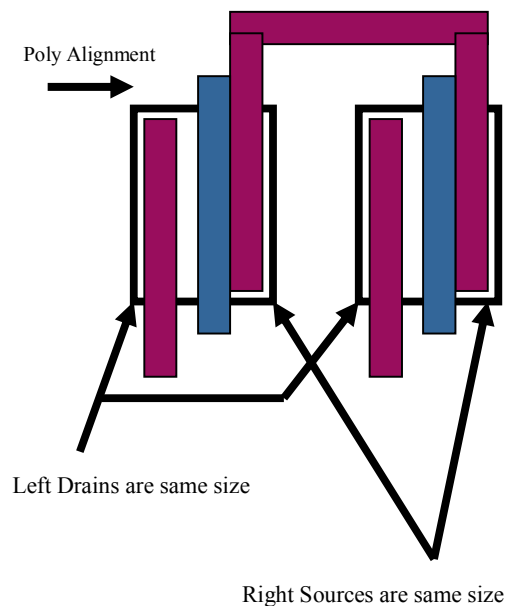


Figure 8

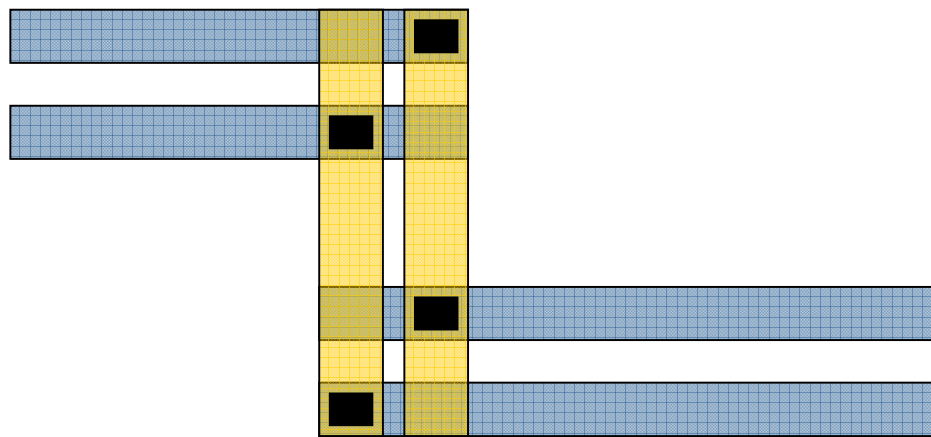


Figure 9

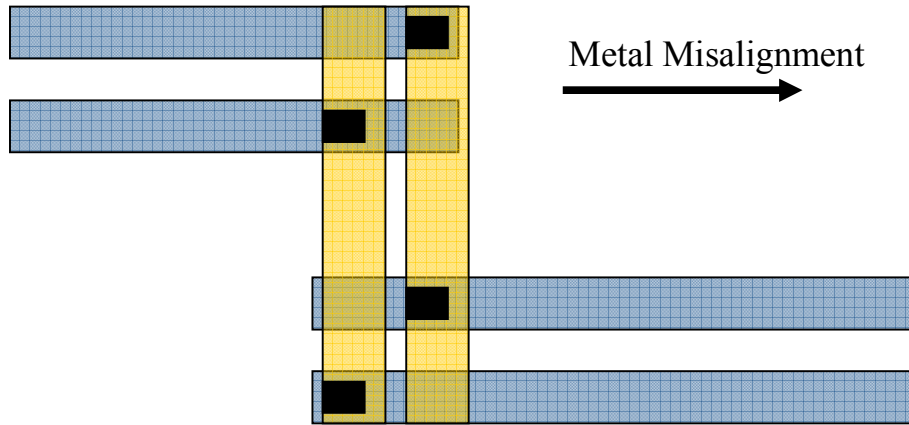


Figure 10

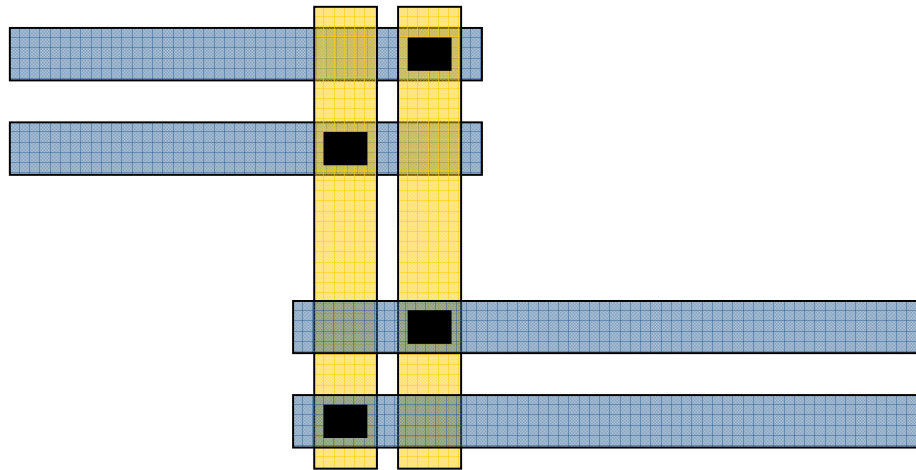


Figure 11

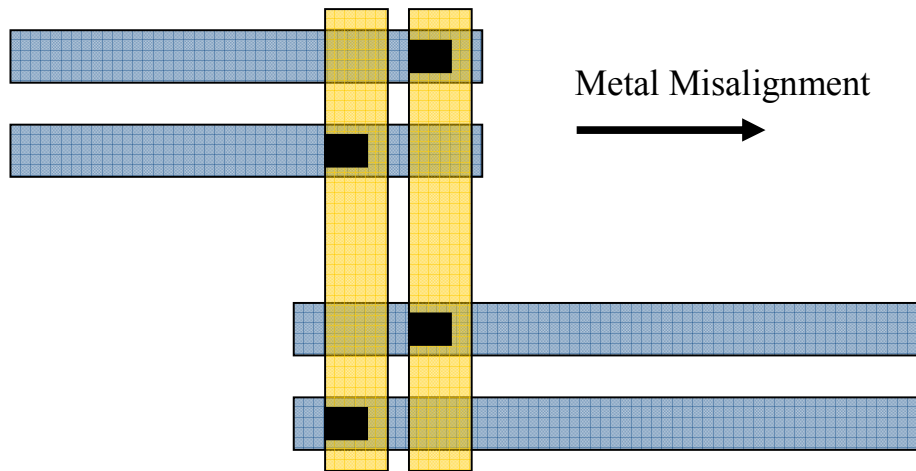


Figure 12

Figure 7 shows the input differential pair placed in a stepped layout. As above the important layers to look at here are active and poly. In Figure 8 the poly mask has been shifted to the right. Both of the sources get larger, and both of the drains get smaller. Since both sources and both drains have the same dimensions, the circuit remains balanced. Stepped devices create well matched circuits.

Figure 9 shows a portion of differential interconnect. Here the layers to look at are Metal-1 (blue) and Metal-2 (yellow). In Figure 10 the Metal-2 mask has been shifted to the left. The right Metal-2 trace has more parasitic capacitance to substrate than the left Metal-2 trace. This means the two traces will not have the same value parasitic capacitors, and could be unbalanced.

Figure 11 shows a layout of differential interconnect that should be immune to mask misalignment. Endcaps have been added to both the Metal-1 and Metal-2 traces. Figure 12 shows the effect of shifting the Metal-2 to the right. The parasitic capacitance stays the same irregardless of mask misalignment.

Mismatch due to misalignment is not found by running Monte Carlo simulations. Monte Carlo simulations generally vary model parameters. Conventional RC extraction assumes perfect mask alignment. To see the effects of mask misalignment we need to do post layout analysis. Misalignment creates systemic mismatches. The circuit will function, but yield will follow normal distribution.

The Solution

Circuit should start with RC extracted layout being properly balanced. The end result that is desired is the difference in capacitance on a pair of nets. Conventional RCX will let the designer balance all nets under ideal conditions. Assura C extraction is used. Only differences in capacitance and MOS AD, AS, PD, and PS are of interest. The geomShift command is used to simulate the mask misalignment. A Spice format netlist is generated because it is easier to parse in a perl script. The extraction is done flat (with no hierarchy) to simplify netlist postprocessing.

The key to simulating the mask misalignment is using the geomShift command; geomShift takes three arguments, the layer to shift, and the X and Y shift values. The syntax is shown below.

```
Outlayer = geomShift( inlayer, X shift, Y Shift)
```

Rule Deck is modified so that input layer(s) are renamed, including the pin layers, and any resistor layers (if defined). Example:

```
met1 = layer("met1" type("drawing")) -> met1_mask_misalign = layer("met1" type("drawing"))
met1pin = layer("met1" type("pin")) -> met1pin_mask_misalign = layer("met1" type("pin"))
met1res = layer("met1" type("resistor")) -> met1res_mask_misalign = layer("met1" type("resistor"))
```

After the layerDefs, insert the geomShift command(s).

```
met1 = geomShift(met1_mask_misalign, 0, 0.05)
met1pin = geomShift(met1pin_mask_misalign, 0, 0.05)
met1res = geomShift(met1res_mask_misalign, 0, 0.05)
```

This methodology allows the rest of rule deck to be unmodified. This means that as new rule decks are released from the foundry, the effort needed to add this functionality is low. Additionally by putting the changes at the beginning of the rule deck, we insure that the functionality of the rule deck is not impacted.

To use the geomShift command we must run lvs with the shifted layer, and then run capacitance extraction. The extract.rul file is modified. A perl script can be used to create the variant extract.rul files. For each layer to be shifted we need four variants: up, down, left, and right. The techRuleSets file is modified to add each layer shift variant. Label each variant, by adding "- layer_name {up, down, left, right}" example: "RCX - Metal1 up." By using the techRulesets file we have any easy mechanism to execute the variants, either manually or through a script.

At this point the user runs an unshifted LVS job, and a “C” extraction. You then use the created .rsf files (lvs and rcx) as templates to run shifted jobs. Create a new <runname>.rsf file and a new rcx.<runname>.rsf file. Within each of these .rsf files replace ?techRuleSet value with the appropriate layer shifted version. Example:

```
?techRuleSet “RCX” -> ?techRuleSet “RCX - Metal1 up”
```

In the rcx .rsf file replace ?output with a unique name so that the spice netlist has a unique name and is not overwritten. Example:

```
?output “{runName.sp}” -> ?output “{runName}_Metal1_up.sp”
```

The last step is to create a script to run all of the shift extractions in one run. An example script is shown:

```
<run_directory>/assura {runName}_Metal1_up.rsf  
<run_directory>/assura rcx. {runName}Metal1_up.rsf  
<run_directory>/assura {runName}_Metal1_down.rsf  
<run_directory>/assura rcx. {runName}Metal1_down.rsf  
<run_directory>/assura {runName}_Metal1_left.rsf  
<run_directory>/assura rcx. {runName}Metal1_left.rsf  
<run_directory>/assura {runName}_Metal1_right.rsf  
<run_directory>/assura rcx. {runName}Metal1_right.rsf
```

This script is run from the same directory that icfb is run from.

The Results

Two types of test cases were run to prove the effectiveness of the technique. The first test case involves creating two versions of the simple op-amp presented above (Figure 4). The first case is a mirrored differential pair (Figure 13), which will be subject to imbalance due to mask misalignment, and the second case is a stepped differential pair (Figure 14), which will not be subject to imbalance because of mask misalignment. For these examples, only shifts in the poly layer were performed, up, down, left, and right. The dimension used was half the spacing between the source/drain contact and the gate. For real applications this offset will be to be provided by the foundry.

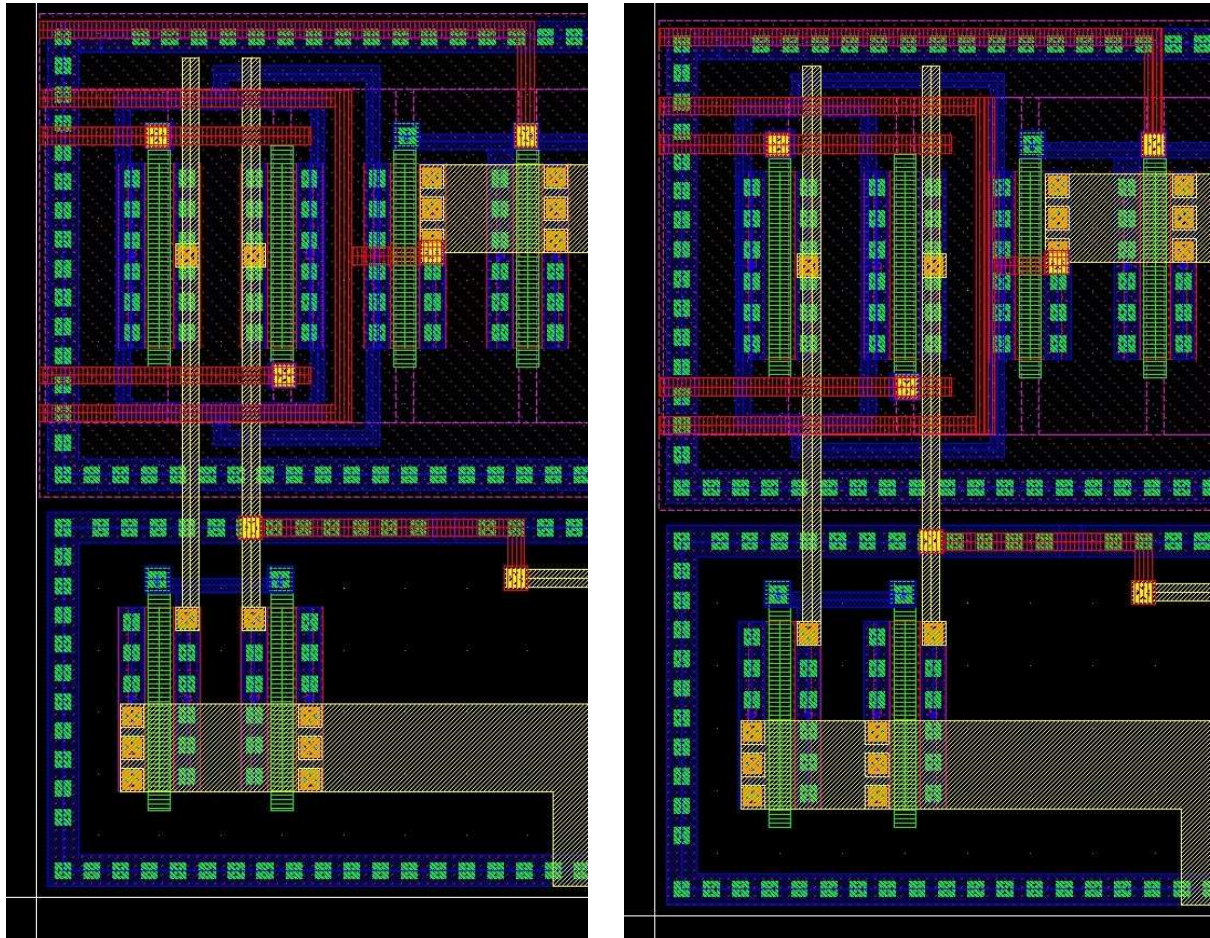


Figure 13 Left Picture: Bad layout for differential pair, subject to mask misalignment.

Figure 14 Right Picture: Good layout for differential pair, not subject to mask misalignment.

Table 1 shows the results of poly shifting the mirrored layout. The layout was created with p-cells that have the source and drain terminals tagged, and correspondingly we do not have to try to figure out which side is which (in some pdk's the source and drain may be considered equivalent, and thus could be swapped). As expected shifting in the up and down directions had no impact to source/drain areas, due to the vertical alignment of the devices. Shifting in the left and right directions gives differences between the matched transistors sources and drains. M4 and M3 along with M2 and M0 have sources and drains changing with opposite values.

Table 1

Original	M4	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M3	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M2	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M0	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
Poly Left	M4	ad=1.68P	as=1.2P	pd=7.12U	ps=6.8U	nrd=8.017E-02	nrs=2.684E-02
	M3	ad=1.2P	as=1.68P	pd=6.8U	ps=7.12U	nrd=2.684E-02	nrs=8.017E-02
	M2	ad=1.68P	as=1.2P	pd=7.12U	ps=6.8U	nrd=8.017E-02	nrs=2.684E-02
	M0	ad=1.2P	as=1.68P	pd=6.8U	ps=7.12U	nrd=2.684E-02	nrs=8.017E-02
Poly Right	M4	ad=1.2P	as=1.68P	pd=6.8U	ps=7.12U	nrd=2.684E-02	nrs=8.017E-02
	M3	ad=1.68P	as=1.2P	pd=7.12U	ps=6.8U	nrd=8.017E-02	nrs=2.684E-02
	M2	ad=1.2P	as=1.68P	pd=6.8U	ps=7.12U	nrd=2.684E-02	nrs=8.017E-02
	M0	ad=1.68P	as=1.2P	pd=7.12U	ps=6.8U	nrd=8.017E-02	nrs=2.684E-02

Poly Up	M4	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M3	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M2	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M0	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
Poly Down	M4	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M3	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M2	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M0	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02

Table 2 shows the results of poly shifting the stepped layout. Shifting in the left and right directions gives the same values between the matched transistors sources and drains. M4 and M3, along with M2 and M0 have sources and drains changing in the same direction. Since the value changes are the same on both sides of the differential pair, no offset is induced by the mask misalignment. Because the absolute values on AD, AS, PD, PS change, a design engineer will probably want to simulate the worst case shifted version.

Table 2

Original	M4	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M3	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M2	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M0	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
Poly Left	M4	ad=1.68P	as=1.2P	pd=7.12U	ps=6.8U	nrd=8.017E-02	nrs=2.684E-02
	M3	ad=1.68P	as=1.2P	pd=7.12U	ps=6.8U	nrd=8.017E-02	nrs=2.684E-02
	M2	ad=1.68P	as=1.2P	pd=7.12U	ps=6.8U	nrd=8.017E-02	nrs=2.684E-02
	M0	ad=1.68P	as=1.2P	pd=7.12U	ps=6.8U	nrd=8.017E-02	nrs=2.684E-02
Poly Right	M4	ad=1.2P	as=1.68P	pd=6.8U	ps=7.12U	nrd=2.684E-02	nrs=8.017E-02
	M3	ad=1.2P	as=1.68P	pd=6.8U	ps=7.12U	nrd=2.684E-02	nrs=8.017E-02
	M2	ad=1.2P	as=1.68P	pd=6.8U	ps=7.12U	nrd=2.684E-02	nrs=8.017E-02
	M0	ad=1.2P	as=1.68P	pd=6.8U	ps=7.12U	nrd=2.684E-02	nrs=8.017E-02
Poly Up	M4	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M3	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M2	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M0	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
Poly Down	M4	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M3	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M2	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02
	M0	ad=1.44P	as=1.44P	pd=6.96U	ps=6.96U	nrd=5.354E-02	nrs=5.354E-02

The second test case involves creating a mismatch sensitive differential interconnect pair. Figure 15 shows the layout used. To simply the RCX output only two input nodes INN, and INP and two output nodes are used, OUTN, and OUTP. The output nodes should not be subject to mask misalignment. Metal-1 and Metal-2 are shifted in this example M1-up, down, left, right, M2-up, down, left, right. The value of the shift is the same as the previous example.

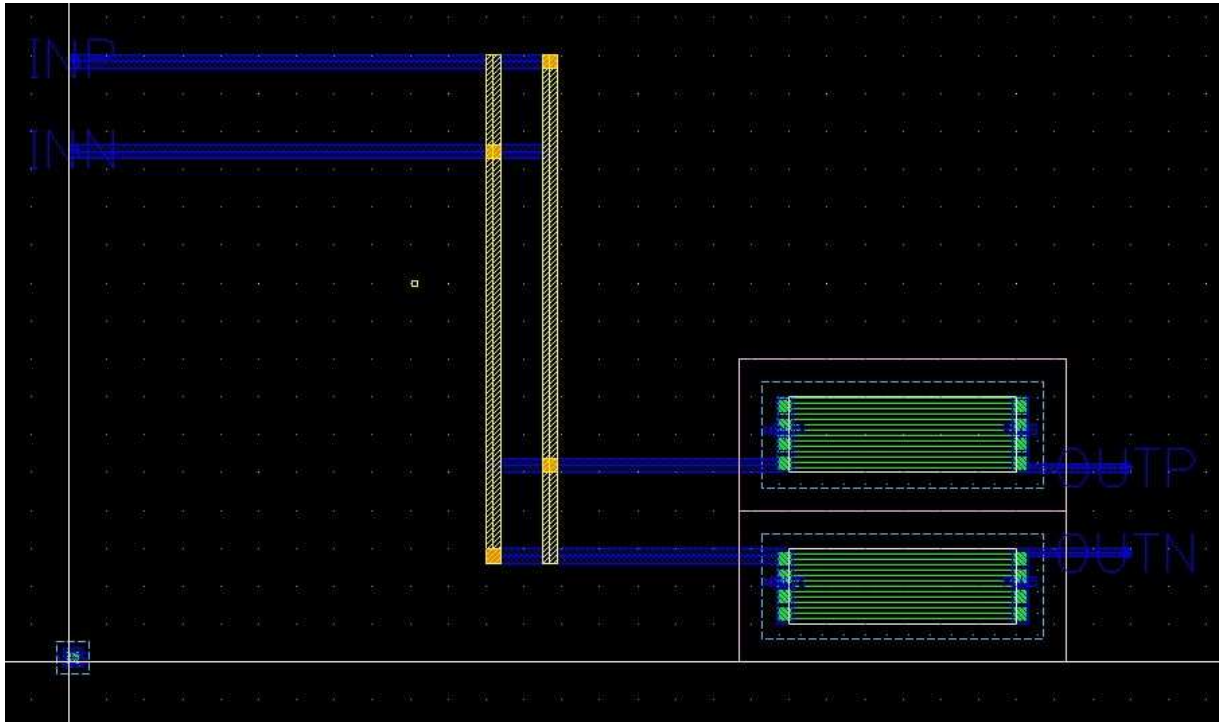


Figure 15 – Differential Interconnect Pair that is subject to mask misalignment.

Table 3 shows that the output nodes OUTN and OUTP are not sensitive to mask misalignment.

Table 3

Original	c2	OUTN	OUTP	5.629E-17
	c5	OUTN	VSS	5.218E-16
	c6	OUTP	VSS	5.218E-16
Metal-1 Up	c2	OUTN	OUTP	5.629E-17
	c5	OUTN	VSS	5.218E-16
	c6	OUTP	VSS	5.218E-16
Metal-1 Down	c2	OUTN	OUTP	5.629E-17
	c5	OUTN	VSS	5.218E-16
	c6	OUTP	VSS	5.218E-16
Metal-1 Left	c2	OUTN	OUTP	5.629E-17
	c5	OUTN	VSS	5.218E-16
	c6	OUTP	VSS	5.218E-16
Metal-1 Right	c2	OUTN	OUTP	5.627E-17
	c5	OUTN	VSS	5.217E-16
	c6	OUTP	VSS	5.217E-16
Metal-2 Up	c2	OUTN	OUTP	5.629E-17
	c5	OUTN	VSS	5.218E-16
	c6	OUTP	VSS	5.218E-16
Metal-2 Down	c2	OUTN	OUTP	5.629E-17
	c5	OUTN	VSS	5.218E-16
	c6	OUTP	VSS	5.218E-16
Metal-2 Left	c2	OUTN	OUTP	5.629E-17
	c5	OUTN	VSS	5.218E-16
	c6	OUTP	VSS	5.218E-16
Metal-2 Right	c2	OUTN	OUTP	5.629E-17
	c5	OUTN	VSS	5.218E-16
	c6	OUT	VSS	5.218E-16

Table 4 shows that the input nodes INN and INP are sensitive to mask misalignment. We only care about differences between INN and INP and do not care about capacitor C1. Capacitors C3 and C4 show a difference in value. This could cause offsets in the signal path.

Table 4

Original	c1	INN	INP	1.096E-15
	c3	INN	VSS	2.725E-15
	c4	INP	VSS	2.725E-15
Metal-1 Up	c1	INN	INP	1.079E-15
	c3	INN	VSS	2.738E-15
	c4	INP	VSS	2.762E-15
Metal-1 Down	c1	INN	INP	1.079E-15
	c3	INN	VSS	2.762E-15
	c4	INP	VSS	2.738E-15
Metal-1 Left	c1	INN	INP	1.096E-15
	c3	INN	VSS	2.740E-15
	c4	INP	VSS	2.755E-15
Metal-1 Right	c1	INN	INP	1.096E-15
	c3	INN	VSS	2.755E-15
	c4	INP	VSS	2.740E-15
Metal-2 Up	c1	INN	INP	1.080E-15
	c3	INN	VSS	2.763E-15
	c4	INP	VSS	2.739E-15
Metal-2 Down	c1	INN	INP	1.079E-15
	c3	INN	VSS	2.738E-15
	c4	INP	VSS	2.762E-15
Metal-2 Left	c1	INN	INP	1.096E-15
	c3	INN	VSS	2.755E-15
	c4	INP	VSS	2.740E-15
Metal-2 Right	c1	INN	INP	1.096E-15
	c3	INN	VSS	2.740E-15
	c4	INP	VSS	2.755E-15

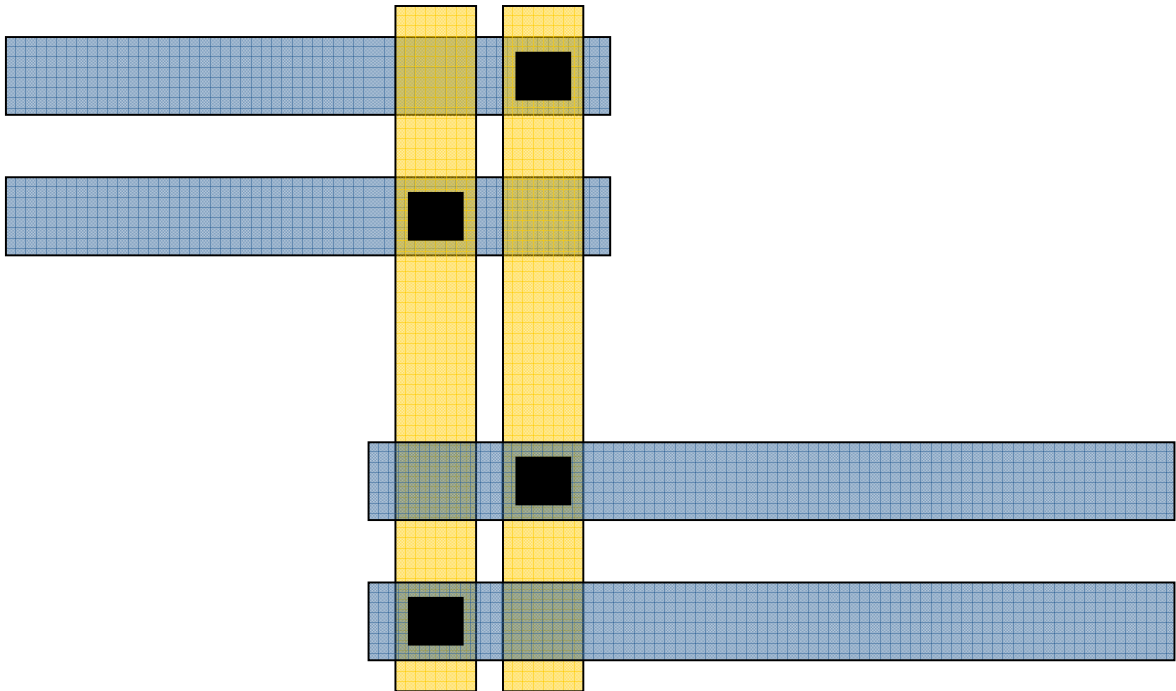


Figure 16 – Detail of endcaps

Figure 16 shows the detail of the endcap fixes for the differential interconnect. The overhang dimension was set to twice the shift distance. Table 4 shows the results. There is still mismatch between the nets, but it has been reduced from 0.024E-15 to 0.002E-15, over one order of magnitude smaller. This is due to making the length of the endcaps too short.

Table 4

Original	c3	INN	VSS	2.834E-15
	c4	INP	VSS	2.834E-15
Metal-1 Up	c3	INN	VSS	2.834E-15
	c4	INP	VSS	2.835E-15
Metal-1 Down	c3	INN	VSS	2.835E-15
	c4	INP	VSS	2.834E-15
Metal-1 Left	c3	INN	VSS	2.826E-15
	c4	INP	VSS	2.828E-15
Metal-1 Right	c3	INN	VSS	2.828E-15
	c4	INP	VSS	2.826E-15
Metal-2 Up	c3	INN	VSS	2.835E-15
	c4	INP	VSS	2.835E-15
Metal-2 Down	c3	INN	VSS	2.834E-15
	c4	INP	VSS	2.835E-15
Metal-2 Left	c3	INN	VSS	2.828E-15
	c4	INP	VSS	2.826E-15
Metal-2 Right	c3	INN	VSS	2.826E-15
	c4	INP	VSS	2.828E-15

Conclusions

Assura RCX can be used as a tool for identifying Mask Misalignment mismatches. The technique can be automated thru use of perl scripts to create the appropriate .rsf files, creating a script to run all of the mismatch variations, and using a script to mine the RCX output data.

As feature size decreases, interconnect mask misalignment loses significance due to thickness of dielectric being greater than the same layer spacing. Large feature sizes have intralayer spacing the same or equal to the dielectric thickness. For very sensitive circuits such as a 16-20 bit A/D converter, the mask misalignment could still be significant irregardless of feature size.

The brute force method may limit applicablity to small circuits. Runtime is a multiple of the basic RCX job. Trying to mine the resulting data for many pairs of devices, could be difficult; yet mistakes could be made at the system level, if one cell in a differential signal path has a poor layout, the entire differential signal path could be at risk. It is easy to limit extraction to a chosen set of nets. There is not an easy way to only extract the MOS devices of interest. Dracula used to have a feature that allowed the used to state the input and output pins, and it would extract the net with MOS devices (passives were not included).

As number of metal layers increases, runtime goes up. For small cells this could lead to huge amounts of data to look at. Some processes which have MIM (metal-insulator-metal) capacitors, where the capacitor may be at the top 2 layers of an 8 layer process.

Currently no methods exist to automatically identify critical nodes in the design. Tools like Accelicon's AVP could be potentially used to identify the critical nets. Running a sensitivity analysis helps identify and circuit sensitivity to model variables. The expertise of an analog design engineer is required to analyze the results.

Future Investigations

The described technique was limited in this paper to poly gate MOS circuits. High voltage MOS devices (LDD devices) may require different layers to be shifted.

NPN's are subject to misalignment mismatch, with the emitter located inside the same active area as the base. Unlike a MOS device, there are no easily extracted parameters on an NPN to indicate any offsets from one terminal to another. Rule deck changes for NPN extraction would be easy.

Logic circuits are sensitive to many forms of timing variation. Changing AD, PD and NRD on a MOS drain used in a gate; should effect the timing of the gate. Thus timing will dependent in part on mask misalignment. As of August 2006, there are no DFM EDA vendors that take mask misalignment into account. The current approach builds all variation into a pre-layout model, and sets timing statistically, as a worse case. One interesting experiment would be to create a set of standard cells where all of the transistors are placed in the same direction (most likely vertically), and all of the sources being on left side, and the drains being on the right. Such a cell may ultimately show more variation in timing because the current layout may have sections that get faster, and some that get slower, offsetting each other. One place where a mask misalignment aware standard cell layout may have impact would be a bus, where differences in timing between signals on the bus could be reduced. This assumes that the place and route program would allow placement of cells without mirroring them.

Parasitic Resistance is certainly affected by mismatch, these variations need to be looked at.