Recommended Rules NOT Recommended
Lou Scheffer

Originally, design rules were the contract between designer and fab. If the designre followed the rules, then the chip could be built with satisfactory yield. (For digital designs, this is functional yield - parametric yield is another matter.)

However, modern rules are not really binary and are set aggressively. This means a small chip might have an OK yield with everything set to minimum, but a large chip may have poor yield. Fabs don't want to penalize the designer of a small chip, and perhaps more importantly do not want to advertise a larger minimum dimension than their competitors. How can they approach this?

Mathematically, the best minimum spacing to use is a function of chip size and desired yield, and can be computed through critical area analysis. So one alternative might be that manufacturers use a set of rules that depend on the size of the chip. But rules that vary with chip size have practical problems - the chip size may not be known (IP for example), small spacings may be needed in a few parts of big chips, and there is little infrastructure for rules that depend on chip size.

Hence the fabs have introduced 'recommended rules'. These are just what their name implies - if you follow this rule your yield will be better. Critically, how much better, the rule does not say.

Typical recommended rules are

- Double vias where possible
- Add extra spacing where possible
- Add more extension around vias where possible

Unfortunately, these rules do not work well at all if you try to optimize yield. The reason is simple - if you have enough room so all recommendations can be followed then these rules might be OK. But this is seldom the case. Historically, smallest chips are best, so no designer will make their chip bigger just to follow recommended rules. After all, they are just recommended, and not required, right? Therefore designer must normally choose which recommended rules to follow, but the rules give no guidance for this.

Here is a simple experiment we have run, in a case where the real data is available. We are trading off the two recommended rules – "double vias where possible", and "avoid min spacing where possible". Both of these rules can be quantified with the cooperation of the fab – you need quantitative data on single vs double via failure rates, and the defect density for opens/shorts of different wire spacings and widths, from which the effects of different widths and spacings can be computed through critical area analysis.

Using this data, we can see how well recommended rules work. The original hypothesis was that the “double via” rule is more important than the “avoid min spacing” rule. We used a router with different settings for inserting double vias. The least aggressive approach is to double vias where it’s easy, doing this as a post process. This inserts the smallest number of double vias. The next level will move lines to insert double vias, but leaves the topology unchanged. The most aggressive level inserts vias during the routing process (concurrent via insertion). This approach can change the topology, and inserts the largest number of double vias.

The results show that even considering just two rules, the best results require tradeoffs. The more aggressive router setting always achieves more double vias. However, this does not always result in the highest yield, since the router must create smaller spacings in many areas to cram in so many vias.

The future looks even worse for recommended rules. Soon, if not now, lithography effects should be included in the router. These are very difficult to express as recommended rules, as many are non-monotonic, such as forbidden pitches. They are not trivial to do quantitatively, either, since they must first be turned into process windows and then into yield info, so they can be compared to via and spacing tradeoffs.

Also, soon we will need recommended rules for width, too. In the past this was not needed, for although the exact models for point defects vary (1/x^3, exponential, binomial), they all fall off very fast with radius, so bigger width/spacing is always better. Also, sine traditionally, shorts were a bigger problem than opens, extra spacing was always the right solution. Furthermore, extra spacing helps timing, so traditional wire spreading spreads wires, does not increase their width, and needs no hard data.

However, with the copper damascene process, opens are also becoming important. So soon we will see recommended rules for width, too. So soon, for every spot in every route, we will have 3 or 4 recommended rules that want any extra space that might be present. Wider wires, bigger spacing, multiple vias, extra via enclosure, and perhaps lithography considerations will all want any extra space that might be present. How can the user possible make the right decisions in this case in the absence of quantitative information?

On a more constructive note, what should be done instead? Here are my suggestions:

First, replace recommended rules on width/spacing with defect density data. Then the router or other tool can find the odds of a short/open based on particle size distribution and geometry by computing the critical area curves, multiplying by the particle density, then integrating over all particle sizes. The data required fot this is the particle density for each layer, for opens and shorts independently. This can easily be expressed as a piecewise linear curve on a log scale. Just a few points are needed for most models.

Next, we should replace the recommended rules on via doubling with a via manufacturability model. A typical model first assumes a 3 (or more) cut via is so much better than a 1 or 2 cut via that extra cuts don't matter. Then all vias can be classified as 1-cut, 2-cut, or many (>= 3) cut. A common customer model then has 1 cut vias that have a cost that depends on overlap, 2 cut vias that have a smaller and constant cost, and 3 or more cut vias that have no cost. A possible extension is a cost that depends on distance to nearest other via (PDF is advocating this).

Next, we need to add data on lithography, which is hard to do with recommended rules A simple representation of process window for long parallel lines (for use by a router) might be a table f(w1,s1,w,s2,w2). This table contains the litho windows under which the wire width is within various tolerances. Note that this table must be computed after the prodution style OPC is run.

A final big problem is giving feedback to the designer. The users of a design tool are ***NOT*** process experts. One advantage of recommended rules is that they are already in their language - either a rule is met, or it is not, and feedback such as error markers are already integrated into layout editors. If we replace these markers by a quantitative calculation we must explain to the user what is going on, and how it can be fixed if it is not satisfactory. Quantitative versions of yield versus width, spacing, multiple vias, and enclosures are more or less intuitive and pose little problems. Likewise, replacing the objective of minimum size with "good chips per wafer" makes intuitive sense. However litho constraints and fixes are not intuitive. Internally, the process is clear- we compute a process window, and from this an estimated yield. However, telling designers what to fix, and how to fix it, is hard. This is especially hard to explain since there is **NO MENTION** of lithography in the design rules. We've looked in detail at the rules from 2 large 65nm manufacturers. There are no litho considerations even in the "recommended" rules, so if the results are unsatisfactory, and the designer says "show me why this is bad", we cannot point to anything in the rules

Conclusions: Recommended rules work poorly. They all cost area, they conflict with each other, and there is no way to choose among them. The situation will get worse – considering opens will create more conflicting rules, and lithography considerations are not easily expressible by recommended rules. Instead, we need real numerical data - particle densities, via fail rates, and lithography information, and we need to improve reporting to match.