VLSI DESIGN AUTOMATION
COURSE NOTES
FABRICATING GATES

Peter M. Maurer
ENG 118
Department of Computer Science & Engineering
University of South Florida
Tampa, FL 33620

1. Fabricating a NAND gate.

Let us suppose that you are designing a tool to fabricate CMOS NAND gates. The layout of the NAND gate will be a collection of rectangles of different types arranged in such a way as to create the desired gate. All coordinates in the layout must be specified in terms of an abstract unit called *Lambdas*. The location and size of each rectangle is specified by giving four coordinates (in terms of Lambdas) and the layer to which the rectangle belongs. The coordinate system is arbitrary, but for the sake of concreteness, let's assume that the lower left corner of the NAND gate is at point (0,0), and that coordinates get larger as one goes up and to the right. It is possible to have both positive and negative coordinates. To begin the fabrication of the NAND gate, we begin by laying a strip of metal along the bottom edge to create the ground connection. The starting point of this rectangle is point (0,0), its height is determined by the design rules, and its length will be determined by a more complex calculation that takes the number of transistors into account. (For the moment we won't worry about the length.) To determine the width of the ground connection, we consult the design rules, which tell us that a metal strip must be at least **M** lambdas wide. In the absence of any other guidelines, we decide to make the metal strip, exactly **M** lambdas wide.

> When the design is sent to the factory, the coordinate system must be changed to match the factory's requirements. It is a simple matter to examine the design, find the maximum and minimum X and Y coordinates, and then add or subtract the appropriate constants to put the origin in the right place. This does not alter the relationship of the rectangles. One can also change from a right-handed to a left-handed coordinate system by subtracting all Y coordinates from the maximum Y coordinate. The factory will then treat the abstract unit, Lambda, as a real physical measurement. The precise size of the measurement (1-micron, 2-microns, 0.5 microns, etc.) depends on the requirements of the factory's equipment, and on the fabrication process

Next, we create a vertical jog to connect to the N-Diffusion layer. There are several design rules that apply to this strip. One design rule states that when a right angle turn is made in the metal layer, that each leg of the right angled turn must be at least **J** lambdas long. The minimum metal width, **M**, applies to *both* dimensions of the rectangle, so we must worry about that as well. This metal strip will contain a Metal-to-N-Diffusion contact, and the design rules state that such a contact must be at least **C** lambdas wide (in both directions). There may be an additional requirement that the metal and diffusion surrounding the contact cut must extend beyond the cut by a certain amount. These amounts may be different for Metal and Diffusion. Assume that the minmum metal overhang is **OM**, and the minimum diffusion overhang is **OD**. The width of the vertical strip can be calculated as Max(**C**+2**OD**,**M**). The height can be calculated as Max(**C**+**OD**,**M**,**J**). These sorts of design-rule calculations are illustrated in Figure 1.

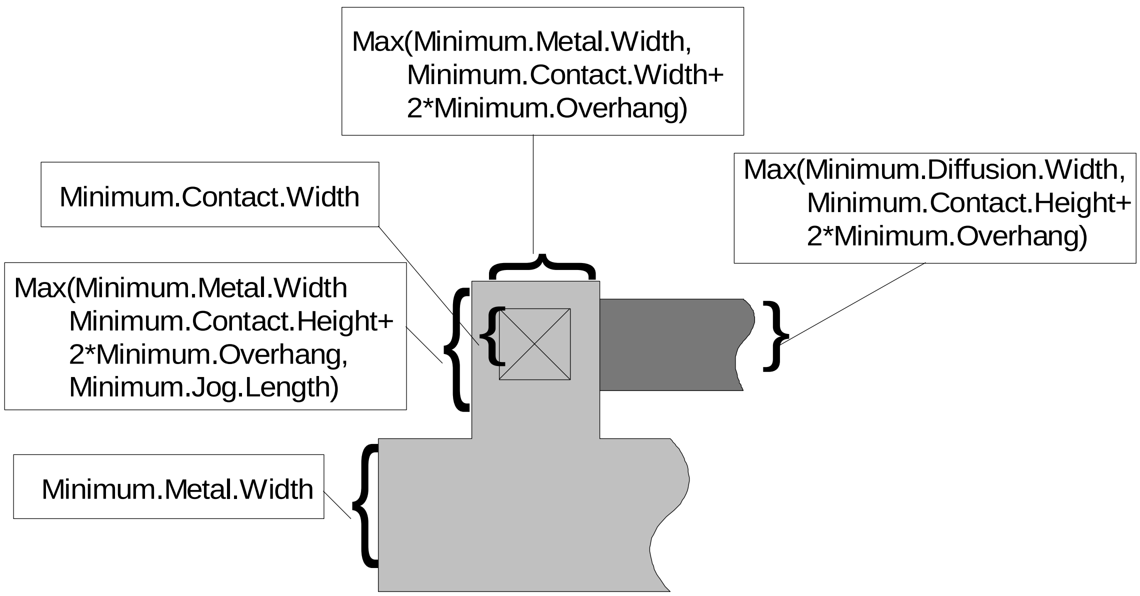

Figure 1. Design Rules A.

To create the N-Channel transistors, we will run a strip of diffusion parallel to the ground strip. It may be possible for all or part of this strip to run under the Ground Strip, but let's assume that the diffusion and the ground strip must be separated by a distance of **GN** lambdas. This will affect the height of the vertical metal strip, and the location of the contact cut. The height of the metal strip must then be at least **GN**+**OD**+**C**+**OM**. This can easily be added to the existing calculation.

When calculating the width of the N-Diffusion, it is necessary to consider minimum diffusion width, minimum transistor height, minimum contact size and minimum overhang. These sorts of design rules are illustrated in Figure 2 When determining the position of the second transistor, it is necessary to take minimum transistor separation and minimum polysilicon separation into account. The width of the polysilicon is determined by *maximum* transistor width and minimum polysilicon width.

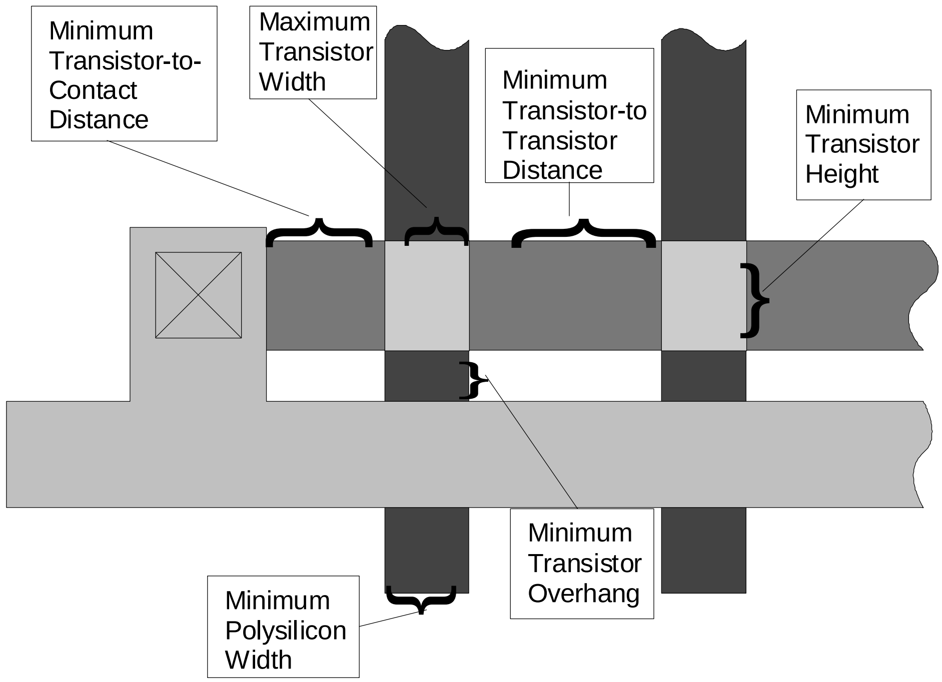

Figure 2. Design Rules Part B.

Figure 3 illustrates that there are similar design rules for the P-Channel side of the the gate. In most cases, the design rules for P-Diffusion and N-Diffusion are different in magnitude.

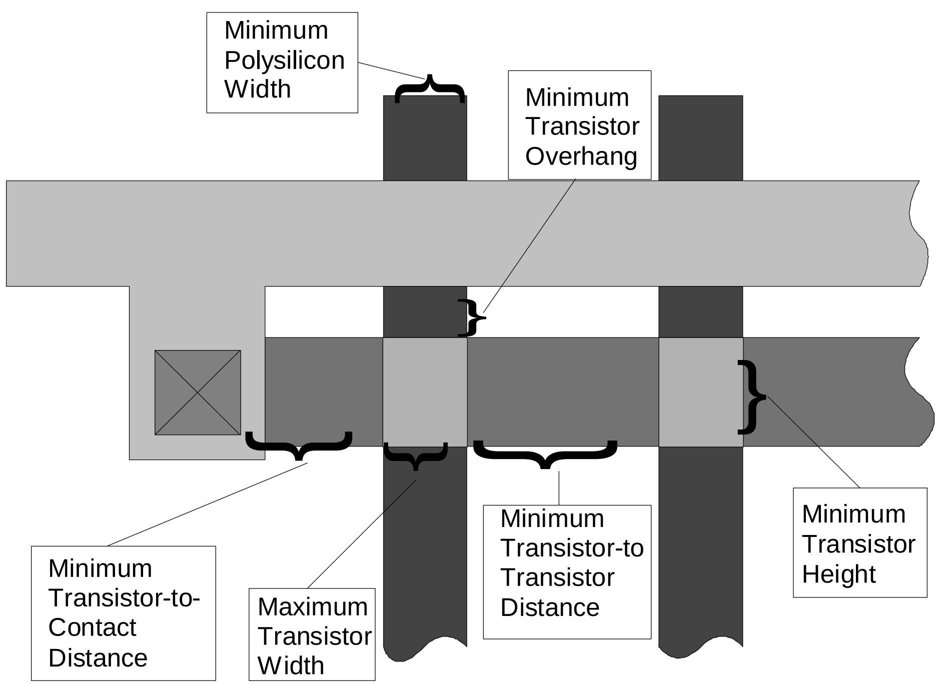

Figure 3. Separate Design Rules for P-Channel.

Finally, the distance between the P-Type and N-Type diffusion must be determined. In addition to the usual computations involving contacts and rectangle widths, P and N type diffusion must be separated by some minimum distance. One reason for this separation, is that a rectangle of P-Type diffusion must be surrounded by a diffused N-Well, as illustrated in Figure 4. The N-Well may be automatically added by a separate tool, or it could be explicitly added by the gate-creation software.

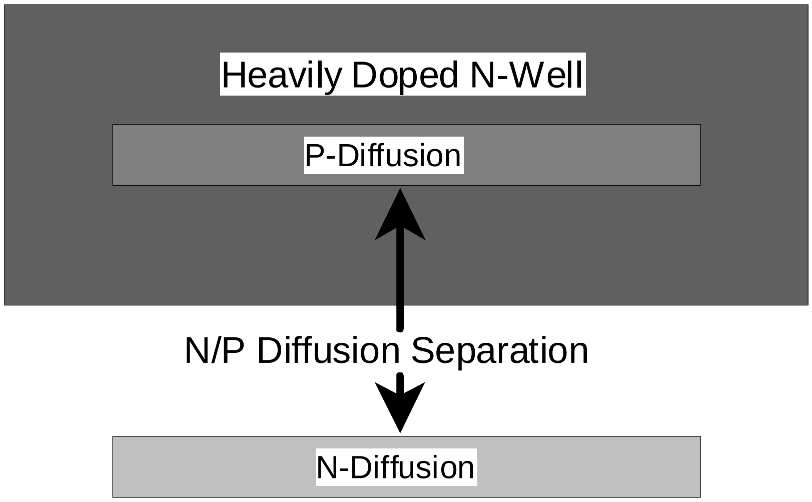

Figure 4. N/P Design Rules.

Although observing the design rules may seem tedious, layout programs usually use parameter files to obtain the values of the minimum widths, separations, and so forth. Once a program has been designed in such a fashion, a new set of design rules may be implemented by creating a new parameter file.

2. A Sample Layout File.

The following is an example of a MAGIC layout file. The file consists of a number of rectangles. "Scmos" is the name of a design-rule parameter file.

```
magic
tech scmos
timestamp 695338699
<< polysilicon >>
rect -33 20 -31 39
rect -22 20 -20 39
rect -11 20 -9 39
rect -33 6 -31 17
rect -22 6 -20 17
rect -11 6 -9 17
rect 1 14 3 39
rect -33 -23 -31 3
rect -22 -23 -20 3
rect -11 -23 -9 3
rect 1 -23 3 10
<< ndiffusion >>
rect -35 3 -33 6
rect -31 3 -22 6
rect -20 3 -11 6
rect -9 3 -7 6
<< pdiffusion >>
rect -35 17 -33 20
rect -31 17 -29 20
rect -25 17 -22 20
rect -20 17 -18 20
rect -14 17 -11 20
rect -9 17 -7 20
<< metal1 >>
rect -42 24 33 28
rect -39 21 -36 24
rect -17 21 -14 24
rect -28 13 -25 17
rect -6 13 -3 17
rect -28 10 1 13
rect -6 7 -3 10
rect -39 -1 -36 3
rect -42 -5 33 -1
<< polycontact >>
rect 1 10 5 14
<< ndcontact >>
rect -39 3 -35 7
rect -7 3 -3 7
<< pdcontact >>
rect -39 17 -35 21
rect -29 17 -25 21
rect -18 17 -14 21
rect -7 17 -3 21
<< ntransistor >>
rect -33 3 -31 6
rect -22 3 -20 6
rect -11 3 -9 6
<< ptransistor >>
rect -33 17 -31 20
rect -22 17 -20 20
```

```
rect -11 17 -9 20       |   << end >>
```