Magic Technology Manual #1: NMOS

John Ousterhout

Computer Science Division
Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720

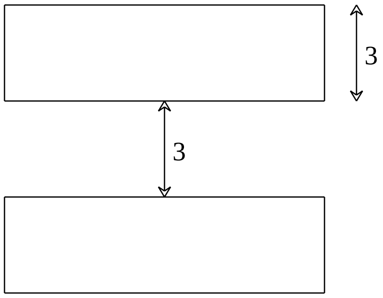
(Warning: Process details often change. Contact MOSIS or your fab line to verify information in this document.)

1. Introduction

This document describes Magic's NMOS technology. It includes information about the layers, design rules, routing, CIF generation, and extraction. This technology is available by the name **nmos** (run Magic with the shell command **magic -T nmos**). The design rules described here are for the standard Mead and Conway NMOS process with butting contacts omitted and buried contacts added. There is a single layer each of metal and polysilicon. If you've been reading the Mead and Conway text, or if you've already done circuit layout with a different editing system, don't forget that these are not the layers that actually end up on masks. Contacts and transistors are drawn in a stylized form that omits implants, vias, and buried windows.

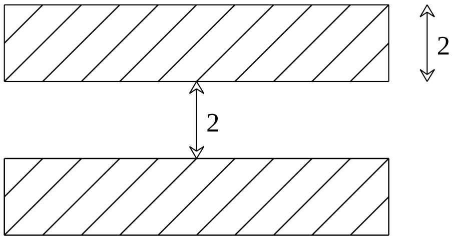
2. Layers and Design Rules

2.1. Metal



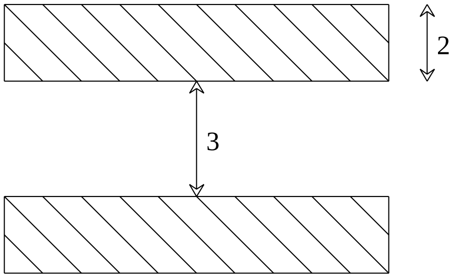
There is only one layer of metal, and it is drawn in blue. Magic accepts the names **metal** or **blue** for this layer. Metal must always be at least 3 units wide and must be separated from other metal by at least 3 units.

2.2. Polysilicon



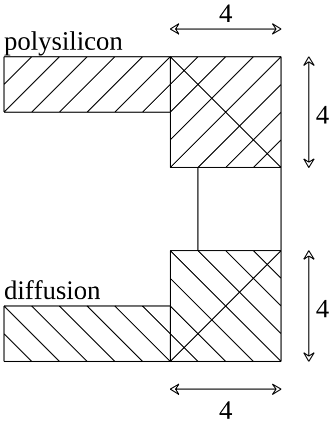
Polysilicon is drawn in red, and can be referred to in Magic as either **polysilicon** or **red**. It has a minimum width of 2 units and a minimum spacing of 2 units.

2.3. Diffusion



Diffusion is drawn in green, and can be referred to in Magic as either **diffusion** or **green**. It has a minimum width of 2 units and a minimum spacing of 3 units.

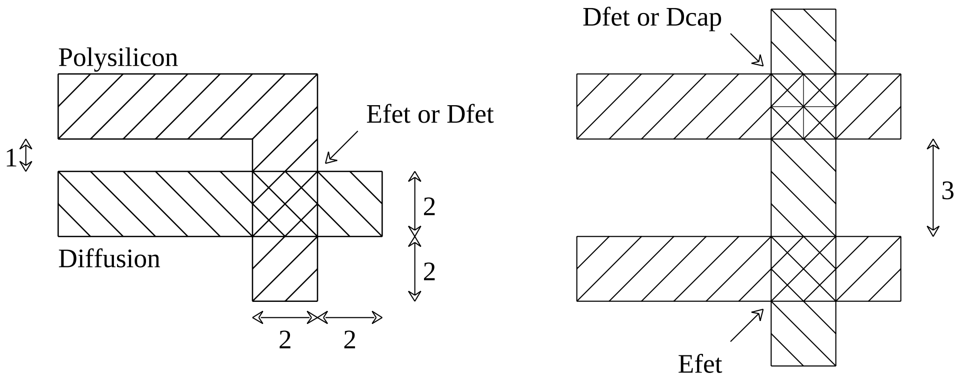
2.4. Contacts to Metal



Contacts between metal and polysilicon, and between metal and diffusion, have similar forms. Poly-metal contacts can be referred to as **pmc** or **poly_metal_contact**; they are drawn to look like metal running on top of poly, with an ‘‘X’’ over the area of the contact. Diffusion-metal contacts are similar, except that they look like metal running on top of diffusion, and have names **dmc** and **diff_metal_contact**. Contacts are drawn differently in Magic than they will appear in the CIF: you do *not* draw the via hole. Instead, you draw the outer area of the metal pad around the contact, which must be at least 4 units on each side. Magic will fill in the appropriate via when CIF is generated. If you draw contacts larger than 4 units on a side, Magic will fill in as many 2-by-2 CIF via holes (with 2-unit spacings) as it can. Contacts areas must be rectangular in shape: contacts of the same type may not abut.

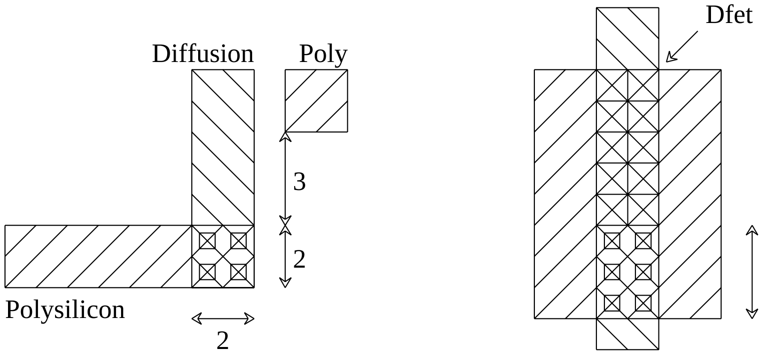
An additional kind of contact, called **glass_contact**, is used to generate holes in the overglass layer for use in bonding to pads. This layer is drawn as gray stripes over blue, and includes both metal and the overglass hole.

2.5. Transistors



There are three transistor structures in the NMOS technology. Enhancement transistors are known by the names **efet** and **enhancement_fet**, and are drawn to look like red over green, with green stripes. You get efet automatically when you paint poly over diffusion or vice versa. Depletion transistors are known by the names **dfet** and **depletion_fet**, and are drawn the same way, except with yellow stripes. A third type of material is called **depletion_capacitor** or **dcap**. It is displayed with yellow crosses over the transistor area, and is identical to dfet except that there are no overhang design rules for it since it is assumed to be used only as a capacitor. You do not drawn any implants in Magic, but just use a different material for the transistor. Magic will generate the implants automatically. All transistors must be at least 2 units on each side, and there must be a poly or diffusion overhang for 2 units on each side of efet or dfet (this is not required for dcap). Poly must be separated from diffusion by at least one unit except where it is forming a transistor. Dfet and dcap must be at least 3 units from efet in order to keep the implant from contaminating the enhancement transistor.

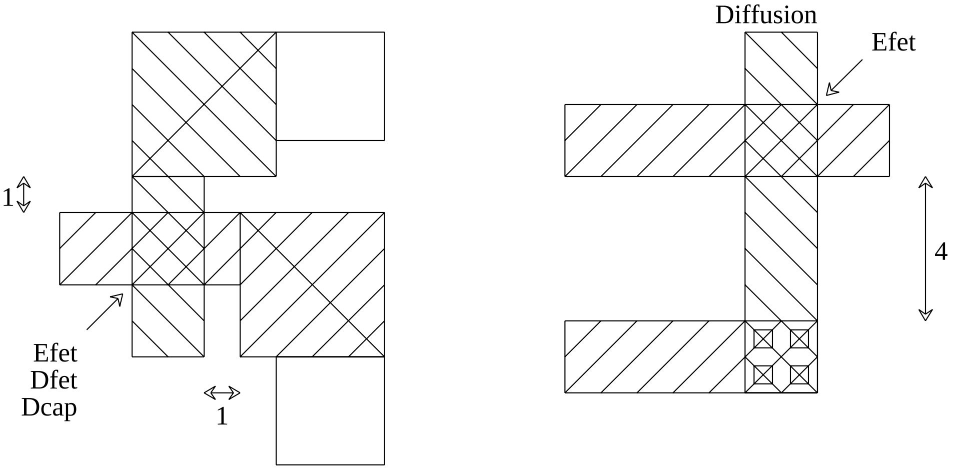
2.6. Buried Contacts



Buried contacts go by the names **bc** and **buried_contact**. They are drawn in a brownish color (the same as transistors), except with solid black squares over their area. As with other contacts, you draw just the area where the two connecting materials (poly and diffusion) overlap; Magic will generate the CIF buried window, which is actually larger than the overlap area. Buried contacts come in two forms. The normal form is 2 units on a side, and no poly or diffusion overhang is required. The second form is used only next to depletion transistors, and is a 3-by-2 structure abutting the depletion transistor. This form is a little controversial, since it results in larger-than-normal variations in the size of the depletion transistor. As a consequence, Magic reports design-rule violations wherever buried contacts abut depletion transistors less than 4 units long. In butting bc-dfet structure, you should measure the transistor length from the bc-dfet boundary.

WARNING: there is one additional rule for buried contacts that is NOT enforced by Magic. Where diffusion enters a buried contact, there must be no unrelated polysilicon for 3 units on that side of the buried contact. This rule is necessary because the buried window extends outward from the buried contact by one unit on the diffusion side, and polysilicon must be far enough away to avoid shorting to the diffusion through the buried window. Unfortunately, there is no way to check this rule in Magic without being extremely conservative (the rule would have to require no poly whatsoever on the diffusion side, even if the poly was connected to the buried contact). So, for now, this rule is not checked. Be careful!

2.7. Transistor Spacings



Transistors must be spaced at least 1 unit from any contact to metal, in order to keep the contact from shorting the transistor. In addition, buried contacts must be at least 4 units from enhancement transistors in the diffusion direction. This rule applies only to the side of buried contact where diffusion leaves the contact.

2.8. Hierarchical Constraints

The design-rule checker enforces several constraints on how subcells may overlap. The general rule is that overlaps may be used to connect portions of cells, but the overlaps must not change the structure of the circuit. Thus, for example, it is acceptable for poly in one cell to overlap poly-metal contact in another cell, but it is not acceptable for poly in one cell to overlap diffusion in another (thereby forming a transistor).

For contacts, there are additional restrictions. A contact in one cell may not overlap a contact in any other cell unless the two contacts have same type and they occupy exactly the same area. Partial overlaps are not permitted, nor are abutting contacts of the same type (contacts of different types may abut, as long as the abutment doesn't violate any other design rules). The contact restrictions are necessary to guarantee that CIF can be generated correctly in a hierarchical fashion.

3. Routing

If you use Magic's automatic routing tools on an NMOS design, the routing will be run in metal and polysilicon, with metal as the primary layer. The routing will be placed on a 7-unit grid.

4. Reading and Writing CIF

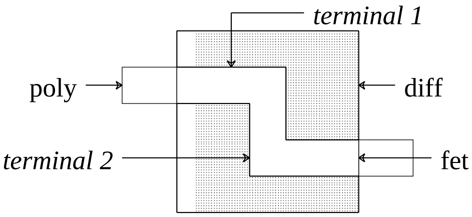
There is only one CIF output style available in the NMOS technology: **lambda=2**. The CIF layers in this style, and their meanings, are:

Name	Meaning
NP	polysilicon
ND	diffusion
NM	metal
NI	depletion implant: generated around depletion transistors and depletion contacts
NC	contact via: generated as small squares inside poly-metal contacts and diffusion-metal contacts
NB	buried window: generated around buried contacts
NG	overglass via: generated for overglass contacts

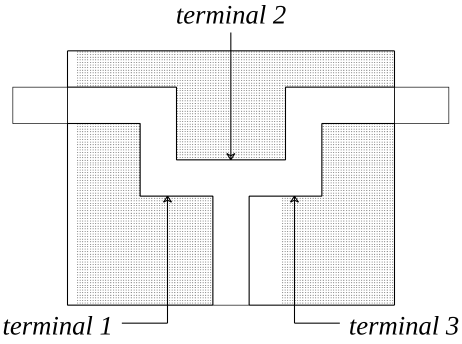
To see exactly where each CIF layer is generated for a particular design, use the **:cif see** command. There is also just one CIF input style. It is called **lambda=2** and can be used to read files written by Magic in the **lambda=2** style, or files written by Caesar using the standard NMOS technology with a scale factor of 200.

5. Extraction

Transistors of type **efet** or **dfet** in the NMOS technology must have at least two diffusion terminals. A diffusion terminal is a contiguous region along the perimeter of the transistor channel that connects to diffusion, as shown below:



2 diff terminals



3 diff terminals

A transistor may have more than two diffusion terminals, in which case it is modeled as a collection of two-terminal transistors. If only one diffusion terminal is present, the the extractor flags this as an error and outputs a transistor with the source and drain shorted together.

Transistors of the special type **dcap** may have as few as one diffusion terminal. Although their normal use is as capacitors, the extractor will output them as though they were a **dfet**. It is up to simulation programs to compute the capacitance of a **dcap** from the area and perimeter of its channel.

The NMOS technology file currently contains little information on parasitic coupling capacitances. As a result, overlap capacitance, and sidewall overlap capacitance will always be zero.