Hspice CMOS Transistor Macros
© R. Hobson, P.Eng.
School of Engineering Science
V1.0 2009-3

Introduction

CMOS circuit simulation is an important part of the chip design flow. There are many ways to connect NMOS and PMOS transistors in general CMOS circuits. Accuracy, performance, and ease-of-use (or re-use) are important. This document shows a variety of transistor structures and introduces Hspice macros to help save designer time and effort. In what follows, the values of "Abx" and "Abx#' are the same.

The Macro library location is described in the hspice_setup document.

The Single Transistor

Transistors that could be considered "single" or "stand-alone" are used in structures like inverters (both N- and P-MOS). Their macro names are NT_ST and PT_ST (the Unix include files have lower case letters). Fig. 1 shows a layout with width (W) and length (L). Metal1 is shown as an outline on the drain and source. Not shown in Fig. 1 is the substrate connection. For NMOS this is assumed to be GND, and for PMOS it is assumed to be Vdd. This is assumed for all Macros.

Figure 1 : An NMOS (or PMOS) transistor with width (W) Abx and min. length (L).

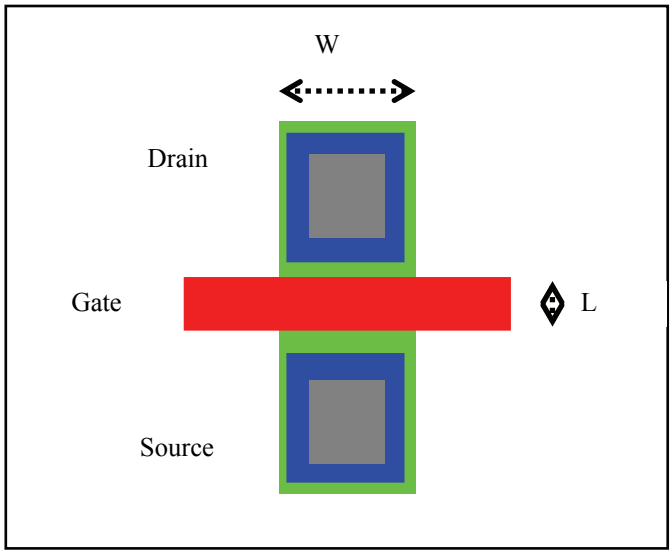


Fig. 2 shows how to invoke these macros in Hspice. There are minimum default values for both W and L. The macros take into account the drain and source capacitance and resistance (from the channel to the Metal1 layer) as a function of W. If W is large enough, you can put in more contacts, as in INV2. You need W>= Ccc+Abx to have room for a second contact. Ccc is about 1.2*Abx for reference.

Figure 2 : Hspice inverter code.

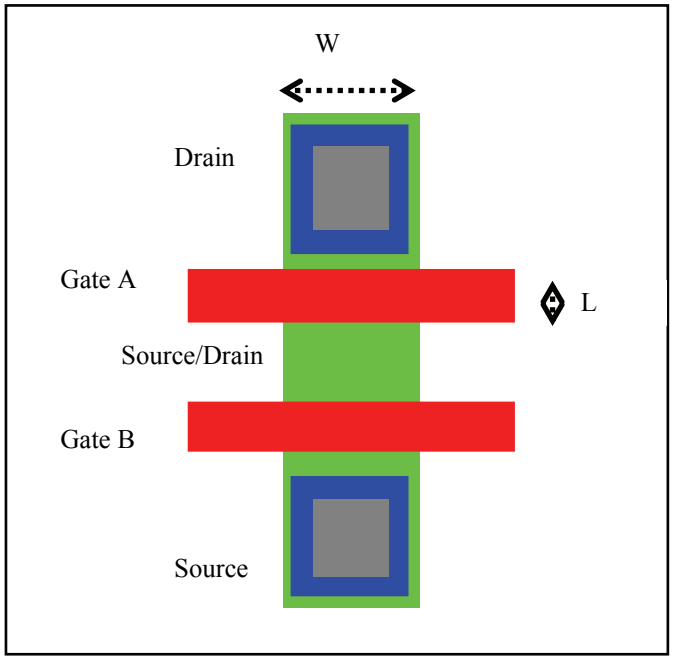
```
.SUBCKT  INV1  A  Z
X1  Z  A  VDD  PT_ST  W='2.0*Abx#'
X2  Z  A  GND  NT_ST  W='1.0*Abx#'
.ENDS

.SUBCKT  INV2  A  Z
X1  Z  A  VDD  PT_ST  NC=2 W='2.2*Abx#'
X2  Z  A  GND  NT_ST  W='1.0*Abx#'
.ENDS
```

Note that there are no other interconnections implemented in these macros. If you know you need "significant" Poly and/or Metal wires, their capacitance and resistance should be modeled separately. Hspice is NOT case sensitive, so the above could be written in lower case.

Transistor Stacks

Logic functions often require transistor "stacks". These are for "ANDing" signals together. You could use a pair of NT_ST (or PT_ST) macros for this, but then you would end up with unwanted Source/Drain material between the 2 transistors. This degrades area, power, and speed. Fig. 3 shows a more efficient method for stacked transistor layout.

Figure 3 : An NMOS (or PMOS) transistor stack with width (W).



The top transistor has a "single" Drain as before. However between gates A and B, there is a "common" area that acts as both Source to A and Drain to B. This region should be divided in half and shared equally between A and B. The bottom of the stack has a "single" Source as before. Macros for such stacks are:

1. NT_CS (common Source). This one goes on top.
2. NT_CD (common Drain). This one goes on the bottom
3. NT_CC (common-common). This one goes in the middle if you have stacks that are 3 or more tall.

Fig. 4 shows how to use these macros for Nand and Nor gates. Note that the "common" area is given a node name (T2 in NAND1, and T1 in NOR1). The outputs are node Z. The NOR1 gate has wide P transistors to compensate for their weakness relative to N. Three contacts are specified (min. width 2*Ccc+Abx).

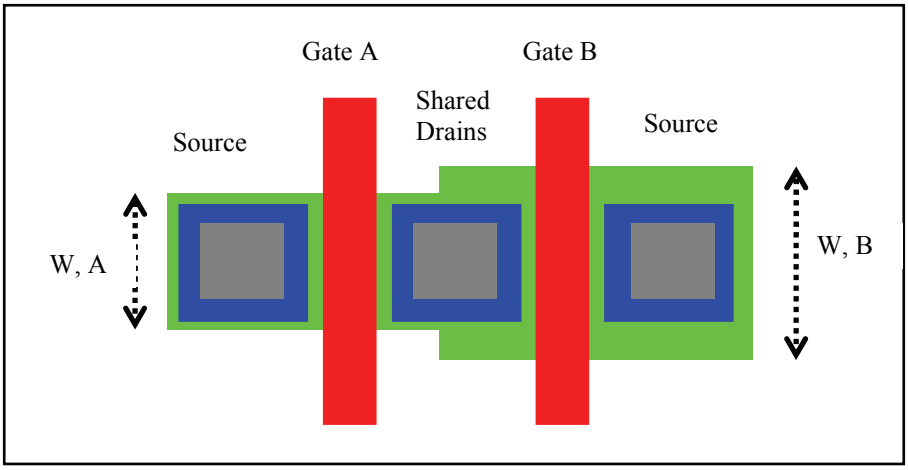
Figure 4 : Hspice code for Nand/Nor gates.

```
.SUBCKT  NAND1  A  B  Z
X1A  Z  A  VDD  PT_ST  W='2.0*Abx#'
X1B  Z  B   VDD  PT_ST  W='2.0*Abx#'
X2A  Z  A  T2  NT_CS  W='2.0*Abx#'
X2B  T2  B GND NT_CD  W='2.0*Abx#'
.ENDS

.SUBCKT  NOR1  A  B  Z
X1A  T1  A  VDD  PT_CD  NC=3 W='4.0*Abx#'
X1B  Z    B  T1     PT_CS   NC=3  W='4.0*Abx#'
X2A  Z  A  GND  NT_ST  W='1.0*Abx#'
X2B  Z  B   GND  NT_ST  W='1.0*Abx#'
.ENDS
```

Shared Drains

When you have an Or structure at a complex CMOS gate output you could end up with 2 or more separate Drain connections. If you want to minimize the capacitance of the Drain for higher performance, then you should share the Drains. Such an example is shown in Fig. 5. If you start with 2 copies of the structure shown in Fig. 1, you have 2 separate Drains with metal contacts. The area of 2 such Drain regions is clearly higher than the shared region of Fig. 5. Note that the transistors don't have to be the same width. Also note that this shared Drain region is larger than the "common" region in Fig. 3.

Figure 5 : A shared Drain structure.



The NAND1 and NOR1 gate examples in Fig. 4 both have Drains that can be shared, P-Drains for NAND and N-Drains for NOR. The macros for this are PT_SD and NT_SD. Fig. 6 shows how to use the macros.

Figure 6 : Hspice code for a shared Drain structure.

```
.SUBCKT  NAND1  A  B  Z
X1A  Z  A  VDD  PT_SD  W='2.0*Abx#'
X1B  Z  B   VDD  PT_SD  W='2.0*Abx#'
X2A  Z  A  T2      NT_CS  W='2.0*Abx#'
X2B  T2  B GND  NT_CD  W='2.0*Abx#'
.ENDS

.SUBCKT  NOR1  A  B  Z
X1A  T1  A  VDD  PT_CD  NC=3 W='4.0*Abx#'
X1B  Z    B  T1     PT_CS   NC=3  W='4.0*Abx#'
X2A  Z  A  GND   NT_SD  W='1.0*Abx#'
X2B  Z  B   GND   NT_SD  W='1.0*Abx#'
.ENDS

.SUBCKT  STUB  A  B  Z
X1A  Z  A  GND  NT_SD  W='1.0*Abx#'
X1B  Z  B   GND  NT_SD  W='1.5*Abx#'  DF='0.5*Abx#'
.ENDS
```

For the NAND and NOR gates, you simply change the name of the Macro. If the transistors have different widths, you have to account for this "difference" by setting the value of DF, as in STUB/X1B, where the B device is 0.5*Abx wider than the A device. If you have parameters for the A and B widths, you can put their difference in the DF expression. The default for DF is 0.

Tapered Stacks

It is sometimes beneficial to taper a stack such that the lower transistor(s) are made progressively wider. Fig. 7 shows this for 2 transistors. This is much like the example in Fig. 3 except that the widths are different. The "common" region is now a bit less efficient due to the requirement of having Active-Poly spacing which causes the "notches" in the common region. Transistor macros for this construction are NT_TP_CS, NT_TP_CD, and NT_TP_CC (also PT's).

Figure 7 : A tapered stack structure.

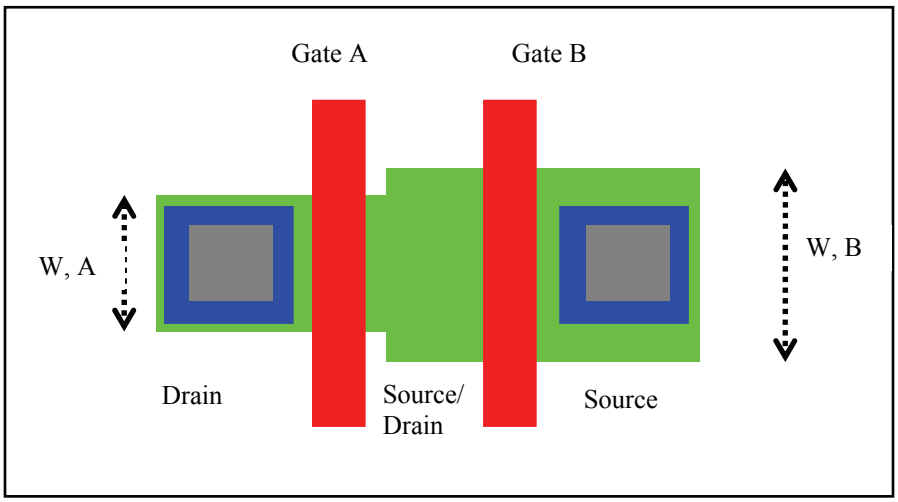


Fig. 8 shows how to use these Macros. In this case there is no need for the DF variable because the Macros automatically choose the correct common areas.

Figure 8 : Hspice code for a tapered stack structure.

```
.SUBCKT  STUB  A  B  Z
X1A  Z   A   T1    NT_TP_CS  W='1.0*Abx#'
X1B  T1  B   GND  NT_TP_CD  W='1.5*Abx#'
.ENDS
```

Variations

More complex circuit structures may require Macros that are combinations of the above scenarios. Fig. 9 shows a tapered stack with a side node "A(B Or C)". The tapered stack could have transistors with the same width (non-tapered). This is similar to having a shared drain in the middle of the stack, except the shared drain is to provide a connection to the "parallel" transistor with gate C. Macros for this are NT_TP_SS (for A), NT_TP_SD (for B), and NT_PR (for C).

Figure 9 : Layout for a complex stack structure.

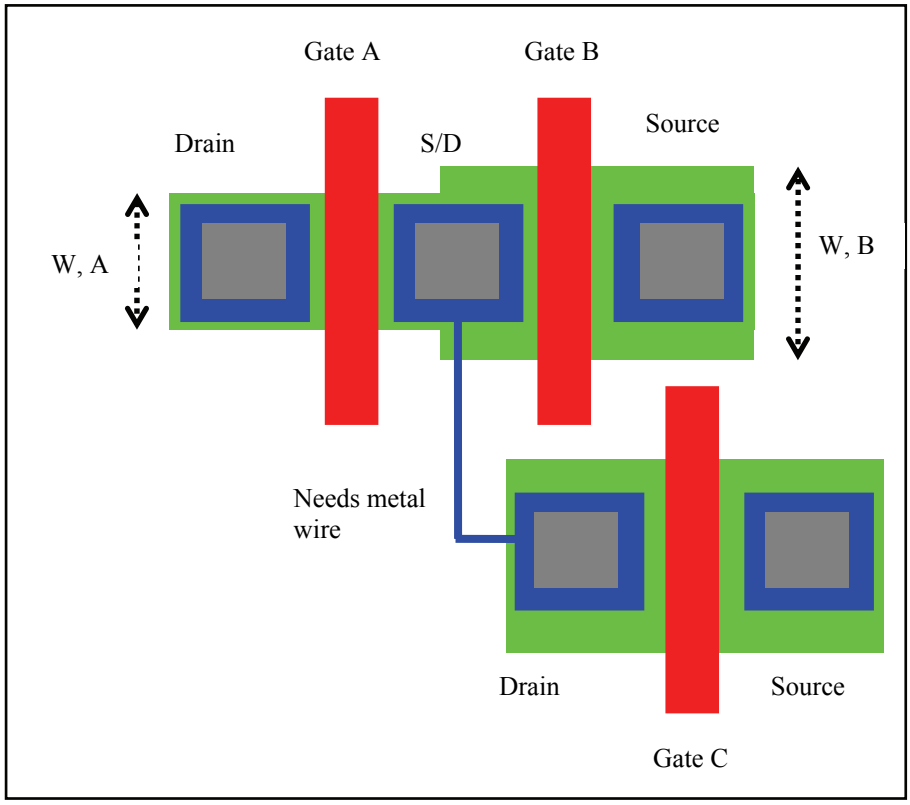


Fig. 10 shows some Hspice code for this example. Note the widths could be all the same, or all different. When A and B have different widths, the DF variable must be set to the difference. The metal1 wire that joins the side node to the stack has a small capacitance, but it is not included in these models. However, the NT_PR drain resistance includes 2 contacts rather than 1. The tapered stack does not have any contact resistance in its common area, as that contact is only used by the side node.

Figure 10 : Hspice code for a complex stack structure.

```
.SUBCKT  STUB  A  B C  Z
X1A  Z    A   T1      NT_TP_SS  W='1.0*Abx#'
X1B  T1   B   GND   NT_TP_SD  W='1.5*Abx#'  DF='0.5*Abx#'
X1C  T1   C   GND   NT_PR   W='1.5*Abx#'
.ENDS
```

All of the examples above have transistors with width >= Abx. Since the minimum width is less than Abx, we need Macros for this as well. NT_ST and PT_ST can be used for minimum widths, but the Drain and Source sizes are not quite right. Fig. 11 shows a minimum width transistor. Macros NT and PT can be used for this type of transistor as they deal with the "notches" between the gate and the D/S areas. NT and PT can also be used for wider transistors, but they don't have the option of adding extra contacts for wider devices.

Figure 11 : A minimum width transistor.

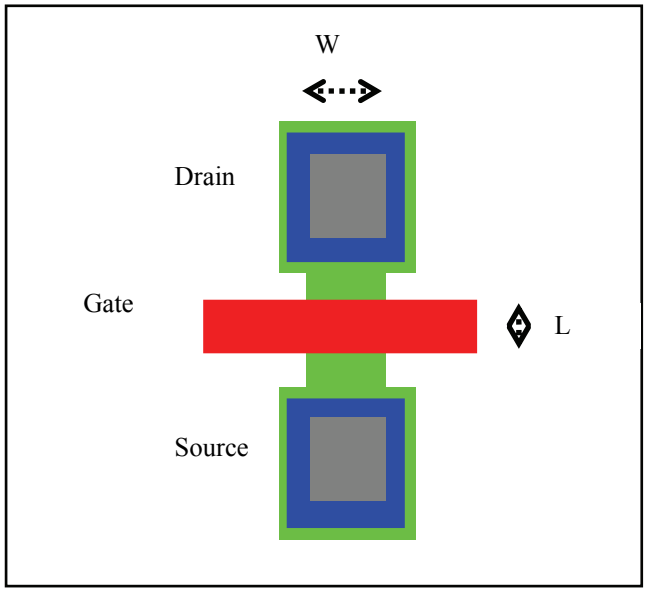


Figure 12 shows Hspice code for small size inverters. Note that Wm# is about 0.55Abx. If you want to stack transistors that are less than Abx in width, you can use NTCS, NTCD, NTCC (and similar PT). These macros can also be used for widths that are greater than Abx, but there is no feature for extra contacts. This is not recommended.

Figure 12 : Hspice code with small size transistors.

```
.SUBCKT  INV1  A  Z
X1  Z  A  VDD  PT    $ min size is default
X2  Z  A  GND  NT
.ENDS

.SUBCKT  INV2  A  Z
X1  Z  A  VDD  PT  W='0.8*Abx#'
X2  Z  A  GND  NT  W='Wm#'
.ENDS
```