# Guide to Primitive Cell Layout
### A Cadence Help Document

**Document Contents**



## Introduction

Following proper layout guidelines for primitive cells will insure maximum layout density and simplify construction of higher level cells that instantiate your primitive cells. This document provides guidelines for primitive cell layout.

## General Layout Guidelines

All primitive cells should adhere to the following guidelines

1. All primitive cells should maintain a cell pitch of 21μm. Pitch is the cell height measured from top of VDD to bottom of GND rails.
2. With the exception of the *nwell* layer, no layer should extend left or right beyond the VDD and GND rails.
3. All intra-cell routing is to be kept within the cell boundaries established by the VDD and GND rails.
4. No *metal2* (or 3) should be used within a primitive cell, unless specifically told otherwise for a particular cell.
5. Set the origin of each cell to lower left corner of the GND rail to easy construction of multi-cell layouts, where cells can be placed side-by-side to form a continuous VDD/GND rail.

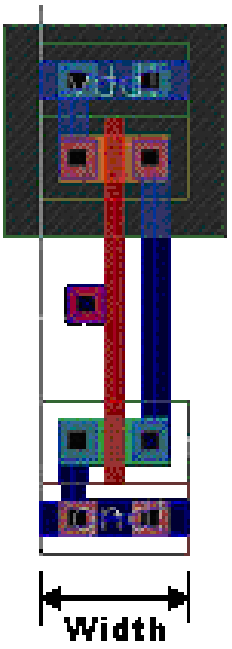


*Figure 1: INV layout.*

## Minimum Cell Width Guidelines

1. The width of a layout is defined as the distance from the left edge of VDD/GND rail to the right edge of that same rail, as identified for the inverter in Figure 1. Width = 4.8μm for this minimum-sized inverter.
2. VDD and GND rails should always have the same width.
3. VDD and GND rails must extend 2λ beyond the any *active, poly, or metal* feature within the cell. NOTE: 1.5λ may be acceptable also, but 2λ is safer.
4. *nwell* can extend beyond the cell boundaries.
5. *nselect/pselect* should generally extend to the left/right edge of the cell, as established by the VDD and GND rails. There can be exceptions to this rule, but multi-cell layouts will generally be simplified by following this rule.

## Port Placement

Ports specify where connections between cells will be made. The best location for ports won't become obvious until you've had some experience routing multiple cells to each other. Figure 2 illustrates how inter-cell routing will be achieved. Here, routing between cells using vertical jumpers of *metal2* tied to horizontal runs of *metal1* above or below the cells, which will route the signal over (left or right) to the

proper cell. Alternatively, in some cases we will be able to place cells so that the output of one feeds directly into the input of the next cell, as illustrated in Figure 2 using blue bars with yellow outlines. In this case, *metal1* can be used to route the signal within the cell boundaries, provided there is no *metal1* within the cell blocking such routing. For adjacent cells with ports close to each other, *poly* can also be used to connect ports, but *poly* traces have a lot of resistance and must be kept very short (roughly no longer than the size of the cell)

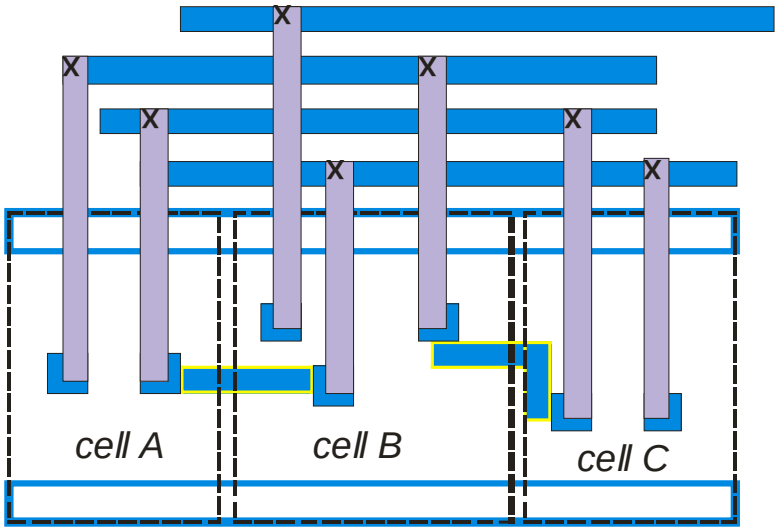


*Figure 2: Inter-cell routing using vertical metal2 and horizontal metal1 traces. The dash-lined boxes represent primitive cell layouts.*

To simplify inter-cell routing, adhere to the following rules for port placement.

1. Use *metal1* for ports: All ports in primitive cells should be made on *metal1* layer. Although ports can be made on *poly*, to follow the routing scheme shown in Figure 2, *metal1* will be needed so each signal can be routed through *metal2* (*metal2* can not make a direct connection to *poly*).
2. Do not vertically stack ports: Ports can be places anywhere within the cell boundary, but each port must have a clear path up and down that will not hit any other port. That is, you should avoid placing any ports directly above or below another port. Adequate space should be provided to route signals up or down in metal2 without crossing another port.

In addition, it is generally preferable to keep input ports toward one side (left or right) of the cell and output port(s) toward the opposite side. This helps to facilitate direct cell-to-cell routing as discussed above. Keep in mind that cells can be flipped or rotated so it is not necessary to distinguish left from right in placing input/output ports.