Additional Object Types:

Two new types of object have been added to chipmonk. One is a regular array of wires, called a "bus"; the other is a new type of transistor: an "angled" transistor.

Busses:

A bus is a regular array of wires. Its parameters are: the layer of the wires, the wire width, the number of wires, the length of the first wire, the spacing of the wires (signed), the amount by which a wire's length differs from that of the previous (signed), and the offset of the "top" end of a wire from the previous (signed). Consequently, a bus can be made only for wires where these things are constant. The wires must be all on the same layer, all the same width, and evenly spaced. Furthermore, the ends of the wires must be in a straight line (at both ends). There are two says to make a bus. The first way is to draw the wires as ordinary wires, then select them and turn them into an equivalent bus (see **<Ctrl>B** and **<Ctrl><TAB>B**, below). The second way is to draw the bus directly, in much the same way as you would draw wires (see **B-Draw**, below). Busses can be turned back into wires with the **expand** commands, like cells.

Angled Transistors:

An angled transistor is a transistor with one right angle in its gate. These may be stuck together to form "button" transistors, "Z-shaped" transistors, and "U-shaped" transistors, etc. (The poly-extension of one transistor will overlap the gate of the other, but this is legal). Angled transistors can be created in much the same way as regular transistors with **Z-Draw** (below). When created, the "X-ratio" is used to set the ratio, as with regular transistors. The gate width is taken as the sum of the inside edges of the two "arms" of the angled gate (i.e. the "corner" square counts as zero).

There are some slight changes in the way the "widen", "shorten", "narrow", and "lengthen" commands work, when applied to angled transistors. The widen and narrow commands change the width of the bounding box, and hence the width of the **horizontal** arm (assuming standard orientation) of the transistor, while the lengthen and shorten commands change the height of the bounding box, and hence the "width" of the **vertical** arm. Consequently, all of these commands affect the "gate width", and none of them affects the "gate length". There is currently no command which will change the gate length of an angled transistor. Angled transistors may be created as either enhancement or depletion (implanted) transistors. Further, there is a set of new commands which allow the implantation of existing transistors to be changed (see the set of **I** commands, below). This is handy if you have created a complicatedly shaped transistor with angles in it out of several transistors, but forgot to turn on implant.

New Commands:

A-Draw Alter. This command applies to either busses, or transistors (of any type, including pullups) and its effect depends on which. The command searches for the first selected bus or transistor which has the mark on it.

If the first such thing is a transistor, the command applies to all selected transistors, and the effect is to change the poly or diffusion **extension** (the amount by which that layer extends beyond the gate). If the mark is on the end of the poly extension, then that extension will be increased or decreased by the difference between the mark and cursor. Similarly for the diffusion extension. This is logically like the stretch command. If the mark is not directly on the end of the extension (i.e. on the bounding box) the command will have no effect. The effect will be applied to all selected transistors. **Changing the diffusion extension of pullups does not have a useful effect**.

If the thing found is a bus, the effect is to change the relative offset of the ends of the wires. The command applies only to the bus which is selected and which has the mark **on** one of its wires. This is one of those things which is in fact simple, but which is complicated to explain, but here goes: The general idea of this command is to "stretch" the end of the wire that the mark is on by the difference between the mark and the cursor, while holding another of the wires fixed, and changing all the other wires to keep the "slope" constant. This applies to the end which the mark is closer to. There are two things about this. First, since the offset between adjacent wires must be an integer, it is not always possible to put the end in question exactly where you specify. Second is the matter of which wire is taken as the fixed one. You can specifiy which is the fixed wire, and each bus remembers (independently for the two ends) which is the fixed wire (it starts out as wire "0", but you can't tell which one that is). A given wire is marked as the fixed one by placing the mark on it (near the end in question) and doing an Alter (A-Draw) with the cursor on top of the mark (i.e. a zero length Alter). This marks that wire as the fixed wire (for this end) until another is specified. As a side effect of all this, if you give an Alter with non-zero length, for the fixed wire, all the wires in the bus will stretch by that amount (but the offset of the ends remains unchanged). **In summary**: there are three cases when an alter is given. If the cursor and mark coincide then mark that wire as the fixed one; else if the mark is on the fixed wire, then stretch all the wires; else stretch the wire with the mark, keeping the fixed one fixed (and all the others adjusted appropriately).

W-Draw Alter width. This command applies only to busses and is used to change the spacing of the wires. The command searches for the first selected bus which has the mark on one of its wires. It then changes the wire spacing so that the wire with the mark will move by the difference between the mark and the cursor, with the fixed wire not moving. The fixed wire is the one that has been marked as fixed (see Alter, above) for the end which the mark is nearer to. Since the spacing must be an integer, it may not be possible to place the wire exactly where you specify.

<ctrl>B Convert to bus. This command looks at all selected wires and determines if they can be replaced by a bus. For them to be replacable by a bus they must: 1) be evenly spaced, 2) all be either horizontal or vertical, 3) all be on the same layer, 4) all be the same width, and 5) have a constant offset of the positions of both ends (i.e. the ends must be in a straight line). If the wires meet all these requirements, they are deleted and replaced by a bus (note: there must be at least two wires). Otherwise nothing happens.

<ctrl><TAB>B Convert to bus, dammit. This command looks replaces all selected wires by a bus, without checking for consistency. Consequently, the bus will not necessarily match the wires, and should be checked and fixed up. Note that expand (<Ctrl>E) applied to a bus will replace it with its wires (though not with the wires it replaced).

B-Draw Start drawing bus. This command starts a bus in much the same way as "Draw" starts a wire. When the command is given, the cursor should be distinctly above/below or to the right/left of the mark, as this determines whether the bus will be horrizontal or vertical (as with wires). When the command is given, it first asks you to mark the origin of the second wire of the bus (the "mark" is the origin of the first wire). Move the cursor to the second wire origin and click the mark button.

You are then asked to type the number of wires in the bus. Type the number, terminated by <CR> or <ESC>. You are now in bus-drawing mode. Two temporary busses are drawn from the mark to the cursor (as with wires), each consisting of three temp wires, one in the location of the first wire, one in the location of the second wire, and one in the location of the last (or Nth) wire. From here on, things are much as for wires. Each time you click Draw, the first temp-bus is made permanent. <ESC> will get you out. If you click Mark, the last two temp-busses are made permanent (the terminal one will have its terminal ends in a vertical or horizontal line) and you get out of drawing mode.

Special commands with special meanings during bus-drawing: While you are in bus-drawing mode, there are several things which you can type to modify the busses you are drawing. <Ctrl>S will increment the spacing between wires of the second temp-bus (which affects the ends of the wires of the first temp-bus) while <Ctrl><Shift>S will decrement the spacing. <Ctrl>I will increment the wire-count of the first (and subsequent) temp-bus, while <Ctrl><Shift>I will decrement it. If this is unclear, just try it.

Z-Draw Make angled transistor. This command places an angled transistor, just as T-Draw (and X-Draw) do for "straight" transistors. The gate ratio is taken from the "X-Ratio" parameters (in the feed-back area) as for "straight" transistors. The gate width is the sum of the (short) width of the two arms (the corner part isn't counted). See the discussion of angled transistors above. Note: these may be either implanted or not. Note: if the X-Ratio length is larger than 1, the transistors that are created will be peculiar.

The following "I" commands apply to busses or transistors, with the shift key controlling the "direction" of the change. For transistors they turn Implant on (if no shift) and off (if shift). For busses they increment (if no shift) or decrement (if shift) the wire count.

<Ctrl>I Increment wire count (busses) or turn on Implant (transistors). This command applies to all selected busses and transistors.

<Ctrl><Shift>I Decrement wire count (busses) or turn off Implant (transistors). This command applies to all selected busses and transistors.

I-Mark Increment/decrement wire count of bus pointed to, or turn Implant on/off for transistor pointed to. The Shift key controls the direction. No shift is increment/turn on; Shift is decrement/turn off.

<Ctrl>T Enter text. This command prompts for a text string (terminated by <CR> or <ESC>) and gives that name to the selected object (if more than one object is selected, it gives it to one of them). Any instance of any type of object can now have a name. If the object is a wire or a contact, the name is taken as the signal-name of that node. For other types of objects, the name has other uses. The name of an object gets displayed in the feed-back area under certain circumstances (see "New feed-back stuff", below).

New Feed-back Stuff:

There are two new things in the feed-back area, in the lower right. One is "Generic Cell Name" and the other is "Instance Name". These two fields get updated with the information from the newly selected object whenever you do a **Select** or a **<Ctrl>Select** (they do **not** get updated at any other time, even when the object gets deleted). "Generic Cell Name" displays the "Cell name" if the object in question is a cell, otherwise it tells you what type of object it is (wire, contact, etc.). "Instance Name" displays the text string, if

any, which was given to that instance with the T-Mark or <Ctrl>-T command.