

XEROX

PALO ALTO RESEARCH CENTER

Systems Science Laboratory

LSI Systems Area

March 6, 1978

For Xerox Internal Use Only

ICARUS2 (version 1.3)

An Interactive Integrated Circuit Layout Program

by Douglas Fairbairn and James Rowson

Introduction

ICARUS2 is a drawing program used for interactive layout of integrated circuits on a standard Alto computer. Check plots can be printed on the Dover, Versatec printers, or the 3100/SLOT printer. Pattern generator tapes can be made from the ICARUS2 data base for use in mask fabrication. The program includes no capability for simulation or design rule checking.

ICARUS2 is an enhanced version of the IC layout program written by Jim Rowson in 1976. ICARUS2 features an improved user interface, virtually unlimited storage space for circuit elements, and increased display redraw speed. The command set is essentially the same, but the mode of operation of these commands is considerably altered. The file structure for the two systems is upward compatible, which means that layouts created with ICARUS are readable by ICARUS2.

ICARUS2 is able to offer a good interactive environment and at the same time allow for the design of high complexity circuits by allowing the user to specify those sections of the circuit which need to be worked on at present and those which can be ignored for the time being. Unfortunately this requires that the user do some memory space management, the techniques of which are described at the end of this paper. The space management issues can be ignored while you are becoming familiar with ICARUS2, but you should understand those issues before you set out to design a moderate to large complexity circuit.

To use ICARUS2 retrieve the file [IVY] \<icarus>ICARUS2.DM using the *Load* command in FTP. You will get the files ICARUS2.RUN and ICARUS.AL. The latter is the font file used for text display in the program. There are no standard symbol libraries available although these may be developed over time.

Terminology

Before I get into the details of the interaction with the program, it is necessary to clarify the meaning of various terms.

The three buttons on the mouse are called RED (top or left-most button), YELLOW (middle button), and BLUE (bottom or right-most button). The functions of the buttons change depending on the operation being performed.

ICARUS2 layouts are done by drawing rectangles on the Alto screen. Each rectangle is called an *item*.

A *symbol definition*, or *symbol*, is a group of items stored as a single unit. These symbols are stored in what can be thought of as a symbol library. Presently there may be up to 175 symbols within this library. The symbols are the basic building blocks which are used in constructing a complete layout. Examples of things which might be symbol definitions are a memory cell, an ALU slice, or an output driver. When you want to include a symbol definition in your layout, you must draw a *symbol instance*. The symbol instance may describe a single copy of the specified symbol or it may describe an array of copies. When you create a symbol instance, you are asked to specify how big the instance array is to be. A symbol instance is treated as a single entity, just as an item is. You may not delete or modify one of the symbol definition copies in the instance without deleting or changing them all. New symbol definitions are created with the *Def Symbol* command, and the symbol instances are created with the *Draw Symbol* command (see below).

A symbol library is associated with a drawing and does not exist as a separate entity or file. The file which holds your items and instances also includes your symbol library.

A *selected* item or symbol instance is one which is outlined with a black line, with a double width line used in the case of a selected instance. This outlining is used to identify items which will be affected by commands such as *move*, *copy*, *delete*, etc.

The ICARUS2 Display

The ICARUS2 display features two windows which give a very flexible working view of the integrated circuit being laid out. The upper window is generally used for viewing a large piece of the layout at a small magnification. The lower window is generally used for looking at a smaller section of the circuit in more detail (high magnification). You may of course use the windows in the way that best suits your purposes.

In addition to the two windows there are various menus and status lines on the left side and in the middle of the display. The menu on the left is the *command menu*. The menu under the upper window is the *parameter menu*. Under the *parameter menu* is the *stipple menu*. The current coordinates and status of memory space available is to the right of the *stipple menu*.

Stipple Menu

The stipples are the various half-tone patterns which are used to represent the different mask layers. You can select which stipple you wish to draw with by pointing at the stipples and clicking RED. The black box which surrounds the number of the mask layer will move to reflect any new selections. In the standard N-channel MOS process we normally use, the stipple patterns correspond to the following masks:

Stipple	0 - Depletion-mode implant
	1 - N-type diffusion (source/drain)
	2 - Polysilicon
	3 - Contacts
	4 - Metal
	5 - Buried Contacts
	6 - Over-glass (passivation)
	7 - Undefined

More mask levels can be added for use with different processes.

Basic Operations

The basic operations in ICARUS2 are those which involve creating, placing, and modifying items or symbol instances. The command language has been optimized around these operations, as they are by far the most frequent.

Drawing - This is the basic operation in ICARUS2. It is accomplished by selecting the appropriate stipple and then moving to either the upper or lower window. You mark a starting point for a line of default width (see *parameter menu* below) by clicking RED. A small cross will appear to temporarily mark the point. Now move to the other end-point of the desired line and click YELLOW. A line will appear. The system will only draw horizontal, vertical, and 45 degree lines. The system determines which of these angles your two points most closely approximate and draws the appropriate lines.

[ICARUS2 will not allow you to draw 45 degree lines which are longer than 300 units(microns). This is a limitation of the pattern generator program used to make mask files from the data base. In addition, ICARUS2 also forces the width of 45 degree lines to be less than 32 units because of internal limitations. In both these cases, you can abut several lines to get any width and length you desire]

Selecting - An item or symbol instance may be selected by pointing at it with the cursor and clicking BLUE. This operation will deselect any items which are already selected. If you wish to select an item without deselecting other items, hold down CTRL while clicking BLUE. If you want to select all the items in a given area, mark the upper left corner of the rectangular area by clicking RED and then, while holding down the SHIFT key, click BLUE at the lower right corner. Only those items which are contained completely within the defined rectangle will be selected. When you are drawing items, the system will automatically deselect any previously selected items and select the most recently drawn one.

Moving - You may move selected item(s) or instances by marking (click RED) a reference point, and then, while holding down the CTRL key, marking a destination point. Each selected item or instance will be moved the same distance and direction as the two points were separated. If no specific reference point is marked before the mark with the CTRL key down, the reference point will be assumed to be the coordinate of the last button click. In the case where you are drawing a line, the last button click is the second end-point specified. Therefore, if you draw a line in the wrong place, don't erase it and redraw it. Simply *move* it by pressing CTRL and clicking RED again at the new desired endpoint.

WARNING: If you are moving a large number of items in a large drawing, moving those items may take 15 seconds or more. During the operation, it will look like nothing is happening. Be patient.

Copying - The *copy* operation is very similar to the *move* operation. Reference points are selected in the same way and the CTRL key is used when specifying the destination. The difference is that you click YELLOW instead of RED at the destination. This will create a new copy of the selected items instead of moving the original ones.

A useful way to look at the last two operations is to consider the RED and YELLOW mouse buttons to be Mark and Draw in normal operation. The CTRL key is used as a "shift" key to change their meanings to Move and Copy respectively.

Flashing - To create a *flash* hold down the TAB key while you click YELLOW. This will create a square of the specified size (see *parameter menu*) centered at the current cursor coordinate.

Stretching - This is a very useful procedure which can be used to change the size of an item, without erasing it and recreating it. It has the additional advantage that the new item uses no additional memory space (see memory space management below). Stretching only works on selected items. Only one item may be selected when stretch mode is entered.

The stretch mode is entered by typing "S" and holding it down. If you want to make a selected item larger, hold the "S" key down and click YELLOW on the side of the object which you want stretched. The appropriate side will stretch to the specified coordinate. If you want to "stretch" a side inward (make the item smaller) you must hold the "S" key, click RED near the edge you want to stretch and then click YELLOW at the destination.

Scrolling - You may scroll around on your IC layout so you can see more of the chip by using the SPACE bar in conjunction with the RED and BLUE mouse buttons. To scroll the upper window, hold down SPACE and click RED (the top mouse button) while pointing at the coordinate you want centered in the upper window. You may point in either window. To scroll the bottom window, hold down SPACE and click BLUE (the bottom mouse button). The coordinate you click at will appear in the center of the bottom window. You may point in either window.

This operation may be a bit confusing at first. Try the following rule to help: To center

an object in the top window, point at it with the top mouse button. To center an object in the bottom window, point at it with the bottom mouse button. Where you do the pointing (top or bottom window) is incidental.

Command Menu

The command menu displays the most commonly used commands and allows you to select them with the mouse. The currently selected command is identified by a black box which surrounds the command name. Any of the commands may be executed by first selecting it (click RED), and then confirming the selection by typing ESC.

There is another set of commands relating to symbols which are not immediately accessible. To gain access to these commands, merely select *Symbols* on the menu and confirm the selection with ESC.

The following is a detailed description of functions of these commands:

Quit - Returns user to the Alto operating system following *two* confirmations. This command should be used with care. You must be sure to save your drawing before you *quit* or it will be lost. If you do execute this command without saving your drawing, try starting ICARUS2 again and *getting* the *drawing* "icarus.temp". This is a file maintained by the system and updated every eight minutes. It should reflect a fairly recent state of your drawing.

Restart - This command allows the user to delete all the items in the drawing, but leave the symbols, or to delete both data types. One might want to do the former operation to insure that there were no items on the drawing before starting to construct a layout from pre-defined symbols. You might want to delete both symbols and items after saving a drawing and before starting work on a new one. This command is also handy when one wants to reclaim the item space which is being taken by previously deleted items (see memory management below). In this case the user would *Save* his *Drawing*, *Restart*, and *Get Drawing* again.

Get Drawing - Used for reading a file from the disk. After confirmation, it will prompt for a file name. Once the filename has been typed once, the system will remember it. The next time you select this command, it will default to that same drawing name. If you want a different drawing, just type it over the default name. If you include no extension on your drawing name, the system will automatically insert the .ic extension.

When you first type "ICARUS2" to the executive to start the program, you can follow that command by a space and the desired input file name if any. The system will remember this name when you do a *Get Drawing*.

During read-in of the file, the system may find a symbol in the new file which has the same name as a symbol defined in the current drawing. It will then ask if you want the old version of the symbol (the one associated with the drawing), the new version (the one associated with the file being read), or if you want to rename the symbol coming in from the file. Type the first letter of your selection (O,N,R) and the system will go on.

Save Drawing - This command is used to save your drawing on the disk. After confirmation, the system will prompt for a file name. If you have previously read in a file, the system will default to that same name. Once you have specified an output file name the system will remember it. Use any name you choose. If you don't put an extension on your file name, the system will assume .ic.

Redraw - This will redraw the upper and lower windows. This has absolutely no effect on the data. From time to time, the system may not redraw some of the items properly after certain operations. This command allows you to see a fresh view. It should rarely be needed.

Ticks on/off - The user may choose to display tick or grid marks in the lower window to serve as a reference in his drawing. This command will alternately turn these marks on and off. The tick marks are not displayed in the upper window, nor are they printed on the checkplots. If the scale of the drawing is such that the ticks will be too dense to be usable, they are not displayed.

Print - This command is used for making checkplots. When this command is confirmed, it will ask which type of output device you wish to use. The current choices are SLOT and Versatec printers. You will then be prompted for the area you wish to have printed. The user is to mark the upper left and lower-right corners of the rectangular area which he wishes to print. These points may be marked in either window. The system will then outline the boundaries of the pages it will print on the selected device, be it a Versatec printer or some SLOT printer. If the page alignments are correct, proceed as it indicates. You can redefine the area to be printed at this time if you wish.

If you are printing on a SLOT device, you will be asked for the name of a press file to use for the output file. It will default to either the input or output file names with a .press extension if you have specified such filenames. If you are printing on a Versatec printer, there is no press file generated. Your machine must be connected to a Versatec printer to run this option.

In the case of a SLOT printer, after the rectangular area has been selected and confirmed, the system will draw on the display a sample of what each output page will look like. If you wish to print that page, indicate so by typing CR. If you wish to skip the printing of that page, you should type SPACE. If at anytime you wish to abort the operation, type DEL. After typing DEL, any pages which were previously confirmed with CR will exist on "icarus.press" and can be printed. Be careful to check your available disk space before starting this operation. Each page to be printed will use 120 disk pages.

In the case of a Versatec printer, once the printing area is confirmed, the display will go blank and the paper on the printer should start to move. It will print in spurts of about 1 1/3 inches/spurt.

Symbols - Select this command if you wish to define, draw, or otherwise manipulate symbols. The system will display another menu for that purpose. See *Symbol Menu* below.

Mirror - If you wish to mirror some items or symbol instances, they must first be selected. After confirmation, this command will ask you to specify the origin around which to mirror and then it asks whether you want to mirror about the X or Y axis. Mark the origin with the cursor (click RED), and then type X or Y as appropriate. It will then ask whether you want to move or copy it. If you choose (m)ove, it will mirror the selected items and erase the unmirrored versions. If you choose (c)opy, it will leave the original items and create the mirrored ones in addition.

Rotate - This command is very similar to *mirror* described above. It prompts for an origin (click RED), and then asks for amount of rotation in increments of 90 degrees. Choosing 1 means 90 degrees, 2 means 180, etc. You then get the choice whether the operation will move the current items or make an additional copy of them. As with *Mirror* only selected items are affected.

Input Text - ICARUS2 allows you to enter text at any point in the layout. The text is used only by the user for reasons of notation or clarity, and has no meaning within the system. The text will be reproduced on check-plots but will be ignored at pattern generation time. If you wish to include some notation on your final masks, use the font set defined in the metal layer described later in this document.

Appear/Disappear - It is often useful, especially for checking purposes, to temporarily delete one or more of the layers from the drawing. To make a layer disappear, select the desired layer and confirm this command. To make the layer reappear, select the layer and confirm this command again. The disappeared layers will not appear on check plots as long as they are in that state, but will be written out to any files created during *Save Drawing*. The layers are never deleted from the drawing by this command.

Delete - This is the command which is used most frequently, so the system selects it by default. *Delete* only works on selected items. This means that as you are drawing, if you draw an item the wrong size or in the wrong place, you can easily *delete* it by typing ESC.

When an item or group of items is deleted, the command display changes slightly and the command *Delete* is replaced by *Undelete*. Confirming this command will bring back the most recently deleted items to the display. This is merely a safety measure. Once you draw a new item, however, the command reverts to *Delete* and the recently deleted items can no longer be reclaimed.

Symbols Menu

The symbols menu displays those commands which are associated with the definition, drawing, and modifying of symbol definitions and symbol instances. You can get access to these commands by selecting *Symbols* on the command menu and confirming the selection with ESC. The following additional commands will be displayed for you.

Menu Return - Select this command if you wish to return to the normal command menu.

Def Symbol - This command is invoked when you wish to create a new symbol definition to be added to your symbol library. The program will lead you through a series of steps necessary to fully specify the new definition. The following comments will clarify those steps.

- The system will first ask you to define an origin or reference point for the new symbol. This reference point is often in the upper left corner of the symbol, but it may in fact be anywhere. The center of the symbol is another common choice. You specify this origin by clicking RED.

- The next request is for the upper left and lower right corners of the rectangle which will bound the symbol. Specify these points by clicking RED as well.

- Connection points may now be specified. Do so by clicking RED at the desired point. If you make an error and click in the wrong place, type BS on the keyboard. This will cause the system to ignore the last CP but it will not immediately erase it from the screen.

Connection points are merely graphical reminders of where signals cross the symbol boundary. They are useful because the individual items which make up a symbol are not normally displayed in the instance as it appears on the screen. The connection points then serve as reminders to the user of where exactly to connect signals which come from outside the symbol. The system makes no attempt to verify that external connections really do line up.

- You now will be asked to identify a symbol name, which may be any collection of characters or numbers.

- After confirmation of the name the system draws the symbol for the user, with all appropriate lines clipped at the rectangle boundaries, etc. If the everything is correct, the user confirms with ESC and the new definition is added to the library. If there is something wrong at this point, the user may abort the whole definition process by typing DEL.

If the definition is confirmed, the symbol itself and any line which crossed the symbol boundary will be erased from the screen. This is the appropriate procedure in that the user typically defines the symbol in a scratch area and then replicates it in some sort of array structure (creates an instance) at a different location in the layout.

Delete Symbol - If at some point you no longer want a particular symbol definition in your symbol library, you may use this command to delete it. It will prompt for the name of the symbol and do the appropriate things to remove it from the library. If you have any instances of that symbol drawn in your layout, it will also delete those instances.

Draw Symbol - This command is used for creating instances of symbols whose definitions appear in the symbol library. Like *Def Symbol* above, this command requires that you specify several parameters.

- First specify the name of the symbol which you want to create an instance of.
- Specify an origin. The instance origin is the point where the origin of the first copy of the symbol will be placed. All other copies will appear to the right and below this one. The instance origin may be specified by clicking RED or by typing from the keyboard. You may click RED several times and the program will merely take the last point specified as being the origin. If you enter the coordinates from the keyboard, you type the X coordinate, terminate that entry with ESC and then do the same for the Y coordinate. If you don't type anything or click the mouse button, and merely confirm with ESC, the program will take the default of 0,0 as the origin.
- Nx,Ny: specifying the array size. Here you are asked to specify how many copies of the symbol will be drawn in this instance. It will always create a rectangular array with Nx symbols in the X direction and Ny symbols in the Y direction. As with the origin above, you type the value for Nx and Ny and terminate each number with ESC. If you want the default of 1,1 a single ESC will suffice.
- Dx,Dy: specifying the symbol spacing. You may space the symbol copies in the array at any distance in the X and Y direction. The default values are those which should be used if you want the copies to directly abut. If you want them spaced at some larger or smaller increment, you should type the appropriate number. If you want to change both values, type the new values, terminating each with an ESC. If you want to change only the Dx value, type the new Dx and two ESCs. If you want to change only the Dy value, type a space to leave the Dx at its default value and type the new Dy value terminated with ESC.

When you have finished specifying the above parameters, the system will draw the new symbol instance at the point specified. It will draw only the outlines of each symbol copy along with any connection points which were specified during *Def Symbol*. If you want to see the internal items of the symbol, see the command *Visible* described below.

List Symbols - Use this command to list the names of the symbols you have stored in your symbol library. The list will appear in the lower window. It will temporarily erase that window to show you the list. The list will also include the names of the symbols which are defined within other symbols. If all symbols won't fit on one window, the system will prompt you when you are ready to see the rest of the names. When you type ESC the

bottom window will revert to its original display.

Outlines Off - It may sometime become desirable to erase the outline which surrounds the copies of the symbols in the instances. This command will accomplish that function. It will only erase the outline of those symbols which are declared visible (see below)

Outlines On - The complement of the above command.

Visible - As mentioned earlier, symbol instances are normally drawn with only their outlines showing. If it becomes necessary to see the inside of a symbol instance, you should use this command. After confirmation, it will prompt you for a symbol name, or alternately ask you to point at the desired symbol. If you choose the latter method, move the cursor to the desired symbol instance and click RED. The program will then go to the disk, where it has the details of the symbol definition stored, and display the insides of the appropriate symbol instance. If you want to make all the symbols visible, type ESC when the program prompts you.

Note that the visible property is associated with a symbol name, so all instances of that symbol will be affected. Other symbols will not be made visible unless explicitly requested. Because of the need to put more items on the screen and because the data is stored on disk rather than in main memory, it is wise to only make symbols visible when there is a specific requirement to do so.

Invisible - This merely undoes the above command.

Expand - Once a symbol definition is created, the items which make it up are no longer directly accessible. When an instance of the definition is created, you can only manipulate the instance as a single unit, you can't add or delete single items from it. The *Expand* command allows you to break a symbol definition into its constituent parts again so it can be modified.

After confirming this command, it will ask you to identify the symbol. You can type the name or point at it and click RED. The program then prompts for a location to put the expanded symbol. You can specify this location by clicking RED at the desired point. If you want the symbol expanded in place, you do not have to specify a location at which to draw it. Merely type ESC.

If there is only one copy of the symbol in the instance, the system will delete the instance from its data list and leave you with only the expanded version of the symbol. If the symbol instance involved an array of any kind, the array will remain in addition to the expanded version of the symbol. This feature was included on the assumption that most single-copy symbol instances were defined to take advantage of the memory-saving features of symbols (see memory management below). It is assumed that the reason you are expanding the symbol is to work on that section of the circuit and that you may or may not redefine it as a symbol again. As pointed out below, symbol expansion does not remove a symbol definition from the library. So if you later want to redefine that group of items as a symbol you will probably want to use the original symbol name. If you do

not redefine it, you may want to *Delete* the *Symbol* definition from the library.

Once a symbol has been expanded and modified, you will normally want to redefine it in the library. Expanding a symbol does not remove it from the library, it merely creates an expanded copy on the screen, which can then be modified. When you are ready to redefine the symbol, you can give it the original name or a new name. If you give it the original name, it will erase the old version of the symbol and substitute the new version. All symbol instances which pointed to the old one will now point to the new one. If you give it a new name, the system will just create a new symbol definition in the library.

Parameter Menu

The parameter menu appears across the middle of the display and allows you to modify any of the most commonly used system parameters. As in the command menu, one of the parameters is always selected. You can change the selected parameter merely by typing a number on the keyboard and terminating it with ESC. The number you type can be positive, negative or relative. Relative numbers are added to or subtracted from the current value of the selected parameter. This feature is most useful for typing in cursor coordinates. (The entry of cursor coordinates from the keyboard is planned but currently unimplemented). A relative number is entered by typing "R" and a positive or negative number.

Line - This indicates the current default line width for the selected layer. Line widths are associated with layers and are "sticky", meaning that if you change it for a certain layer, it will assume the new value until explicitly changed. Line widths for other layers are unaffected.

Flash - A *Flash* is a standard-sized square patch of the current layer. It is used principally for making pads around contacts or for contacts themselves. Like line sizes, flash sizes are associated with a layer, and may be changed in the standard way.

Top - The value following this is the current magnification of the upper window. The magnification may have a value between 0 and 20, with the most detail visible at the highest magnifications. If the magnification is set at 9 or below, individual items will not be displayed, only outlines of symbols. This is because the detail is not visible at those times anyway and it merely takes longer to redraw the window.

The magnification numbers relate to the ratio between a dot on the screen and one ICARUS2 unit. ICARUS2 units are normally considered to be microns. At mag 0, there is one screen dot for each 64 units. At mag 20 there are 16 screen dots for each unit. The settings in between these two extremes increase the magnification at a rate of about 4/3 per step.

Bottom - The value following this is the magnification in the bottom window. See *Top* for details.

Grid - You will notice that the cursor moves on a discrete grid when it is in the upper or lower window. This is to facilitate the matching of line end points and the like. The units of this grid are multiples of the basic ICARUS2 units. The number associated with this parameter sets that grid spacing. The grid must be integer multiples of the basic unit, which is usually considered to be microns.

Ticks - This parameter represents the spacing of the tick marks (small dots) which can optionally be displayed on the screen. This parameter may be reset to any integer value. The tick spacing will change to reflect the new value if they are being displayed at that time.

Cursor Coordinates

The position of the last mouse button click is recorded to the right of the stipple menu. Along with the current values (X:, Y:) are displayed the distance in x and y between the current cursor coordinates and the previous ones (DX:, DY:). This display makes a convenient "ruler" for measuring distances on the chip.

Managing Memory Space

As with ICARUS, you will have to be somewhat aware of the space your design is using within the computer's memory. In order to maintain the interactive feel which we felt was essential for this system, we were forced to keep as much of the data describing the circuit in main memory as possible. Unfortunately, this is an extremely limited resource. The alternative to keeping data in main memory is to put the data on the Alto's disk. We have decided to keep the symbol library with all its definitions on the disk. All items and instances not included within symbols are in main memory. In addition, there is a small amount of main memory used for storing a limited amount of data about the symbol library.

As you add items and symbol instances to your drawing, these items use up more space in the memory of the computer. Each item uses 6 words of computer memory for its representation. Each symbol instance uses 14 words of memory. When you start, you have about 5100 words of Item space. This number is shown to the right of the letter "I" in the coordinate window. You will notice that as you add to your drawing this number keeps shrinking. If you delete items from your drawing, it will free up space in the computer memory but unfortunately this space is not immediately usable.

If as you work, you notice that the "I number" is below 200 or so, you should think about defining part of your circuit as a symbol. Elements which are included in a symbol are stored on the disk and thus do not consume main memory space. Thus you can use the symbol definition capability not only to create structures which are to be replicated in arrays, but also to help get around the main memory space limitations. When you are deciding what piece of your circuit to define as a symbol in order to free memory space, try to choose a segment which you don't want to work on for awhile, and one whose parts have a related function. After *Defining Symbol*, you should *Save Drawing*, *Restart*, and *Get Drawing* again. By going through this process, you force the system to reclaim the space devoted to deleted items and to items which were subsequently included into symbols.

When it is time to work on a section of a circuit which was earlier defined as a symbol, you can use the *Expand* command to transform the definition into its constituent items and instances. The description of the *Expand* command details how you can ask for the symbol to be expanded in place while at the same time deleting the original instance. You must be careful that when you request that a symbol be expanded that there is enough room for all the items in main memory. In order to maintain some flexibility in terms of what can be in main memory at one time, it is probably wise to include no more than 1500 words worth of items in any single symbol definition if possible.

If you do not need to operate on the items within a symbol, but merely need to see them, you can ask that they be displayed with the *Visible* command. This operation does not require the use of any main memory space.

There is a small amount of main memory space used for each symbol definition, however. The bounding box of the symbol, its name, and its connection points are maintained in main memory. The number of words of space required for this information is

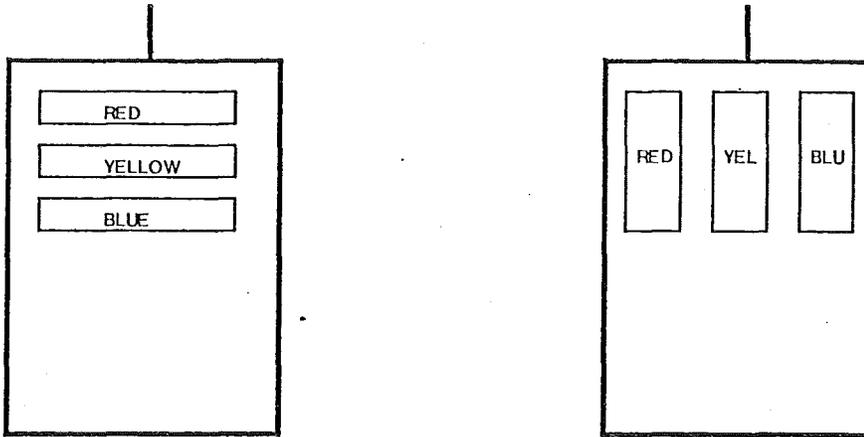
$$\# \text{ of words of space} = 10 + 1/2(\# \text{ of chars in name}) + 1 + 2(\# \text{ of connection points})$$

The space needed for the above information is allocated out of (S)ymbol library space. You have the option of setting the total size of this space when you start up the program. The default value is 400. This is enough space for approximately 15-20 symbols, depending on name lengths and numbers of connection points. The amount of available symbol space is shown next to the letter "S" under the (I)tem space notation. As with the Item space, you should be careful not to exceed the space allocated. If this number gets below 20, you should *Save your Drawing, Quit*, restart ICARUS2, specify a larger symbol space when it prompts you, and *Get your Drawing* again. The system will give you a warning when this number gets below 20.

If you should crash . . .

If your Alto should die, or you *Quit* without saving your drawing, or some other calamity should befall you, take heart. Leave ICARUS2 immediately (if you haven't already), and restart the program. *Get the Drawing "icarus.temp"*. This file should reflect a very recent state of your work. Icarus.temp is updated every fifteen minutes automatically while you're running. Remember to get this file immediately, however, as in fifteen minutes it will be written over with your current work.

ICARUS2 COMMAND SUMMARY



DRAWING:

To draw lines:

Click red at first endpoint and then yellow at second endpoint

To draw a flash:

Click TAB- yellow

To stretch a line:

Click S- yellow to stretch out; Click S- red, S- yellow to stretch inward

SELECTING:

To select an item:

Click blue on desired item

To select items in an area:

Click red at upper left corner and SHIFT- blue at lower right corner

To select items without deselecting old items:

Click CTRL- blue

MOVE/COPY COMMANDS:

To move an item or items:

Select item(s), click red at reference point and CTRL- red at destination

To copy an item or items:

Select item(s), click red at reference point and CTRL- yellow at destination

To scroll around on the chip:

Scroll top window by clicking SPACE- red

Scroll bottom window by clicking SPACE- blue

MENU COMMANDS:

Confirm all menu selections with ESC

Abort any menu command with DEL