A system that can selectively store and erase any pattern of dots in a 64 x 64 matrix is designed in this project. The stored pattern is displayed on an oscilloscope.

The object of this project is to design and build a system that can create and display any pattern of dots in a 64 x 64 matrix. A block diagram of the system is given in Figure D3-1. The heart of the system is a 4096-bit shift register and a 12-bit counter. At each clock pulse, the shift register is shifted one position and the counter is incremented. The lower six bits of the counter are interpreted as an X position and the high six bits are interpreted as a Y position. These bits are connected to two digital-to-analog converters (DACs); the resulting X and Y voltages are connected to a display oscilloscope. Hence, the counting action of the counter causes a beam to scan horizontally across the display, moving vertically one line after each scan. At each count, a new bit appears at the output of the shift register. This bit is fed to the Z-axis of the scope to

![Block diagram](image-url)
determine whether or not the corresponding position of the display is to be intensified (i.e., have a dot appear) or to be left blank. The clock rate is chosen so that a complete shift register and counter cycle is made about 60 times per second, so that a flicker-free display is seen by the user.

A few features are needed to make the system practical. The user needs a way of clearing the entire display and of selectively drawing or erasing dots at any position on the screen. Clearing the screen can easily be accomplished by a "clear screen" control that causes the shift register to be loaded with all zeroes (after 4096 shifts). To draw or erase a particular dot the user must first specify which dot is to be modified. For this purpose a cursor is provided. The X and Y positions of the cursor are set by the user via the "cursor position control." When the user requests a "draw," the current scope XY position is compared with the cursor position. When the positions become equal, the corresponding dot position is available at the shift register and the "draw/erase control" and "recirculate control" cause a 1 to be written instead of the old contents. Erases are similar, with a 0 being written.

The user must have some means of knowing where the cursor is positioned. The easiest scheme to implement is to require the user to enter the cursor position as a 12-bit number and let the user interpret the number as a position. However, this is not too convenient for the user. It is better to provide some means for the user to see the cursor directly on the display. The suggested way of accomplishing this is shown in Figure D3-1. Two counters are used to store the cursor X and Y coordinates. The user, through the "cursor position control," can change either of these counters to move the cursor.

The current cursor position is displayed by inhibiting the clock for four cycles when the scope beam is at the cursor position. The corresponding dot is intensified for four cycles, making it brighter than other dots on the screen.

ASSIGNMENT

You are to design and build a scope-a-sketch system with the capabilities outlined here. You are welcome to deviate from the suggested block diagram if you can come up with a better system. The suggested system has three outputs, to the X, Y, and Z axes of the display scope, and the following controls:

CLEAR - This switch, when set, clears the entire screen.

UP, DOWN, LEFT, RIGHT - These switches are used to move the cursor. As long as a switch is set, the cursor should move in the corresponding direction at a reasonable rate, say 8 to 16 dots (1/8 to 1/4 of the screen) per second.

DRAW - When this switch is set, a dot is drawn at the current cursor position.

ERASE - When this switch is set, a dot is erased at the current cursor position. When both DRAW and ERASE are set, the dot should be complemented.
Note that a DRAW or ERASE command can be given while the cursor is moving, and so the matrix must be updated on the fly. What does this imply about the relationship between the maximum cursor speed and the refresh rate?

PARTS

In addition to parts in the standard kit, the following ICs will be needed:

1 - 4K shift register. Recommended: 2 × Intel 2401 or 4 × Intel 2405.

1 - 12-bit position counter for scan. Recommended: 3 × any 4-bit counter.

2 - 6-bit up/down counters for cursor. Recommended: 4 × 74193 or 74191.

1 - 12-bit comparator. Recommended: 3 × 7485. Alternate: 2 × LM311 or other analog comparator.

HINTS

The Z-axis of most oscilloscopes is DC coupled, but some are AC coupled. With an AC-coupled Z-axis, problems may be encountered when trying to display a completely blank dot matrix, and you will have to design circuitry to overcome this.

One of the most critical parts of the design is the clock inhibit for the cursor display. Be careful not to introduce any hazards here.

OPTIONS

There are other ways to implement the cursor display, for example an intensified caret can be displayed instead of a dot. A blinking cursor is very easy to display (perhaps too easy since it makes the project complexity closer to C-series).

The cursor position control can be implemented with two potentiometers for X and Y instead of two counters and push buttons. The potentiometers produce analog output voltages that can be compared with the scanning position using analog comparators. This approach requires less circuitry than the completely digital approach.

The size of the system can be reduced by displaying a smaller dot matrix. For example, a 32 × 32 matrix requires only 1K of memory and 10 bits of position information.

It is not necessary to use shift registers to store the dot matrix. Static random-access memories such as 2102s (1K by 1 bit) can be used instead, by connecting the output of the position counter to the address inputs of the memory. A dynamic 4K memory such as the 2107 can also be used if its refresh requirements are met.