Mini projects 2019

Text encryption/decryption

Design a system for encrypting/decrypting text files. The system would operate in two modes (selectable by a switch) – encrypt and decrypt. The files are to be read/written using previously designed serial receiver/transmitter. Sequences of operations in the two modes would be as follows.

In encrypt mode: read a plain text file into memory, perform encryption, write encrypted text into a file.

In decrypt mode: read encrypted text from a file into memory, perform decryption to get back plain text, write plain text into a file.

Use any simple stream cipher or block cipher scheme. Key used may be specified through switches.

Image filter

Design an image filter that uses a 3x3 sliding window. The serial receiver downloads the image to be filtered from a file into memory. The serial transmitter uploads the filtered image into another file. Filter coefficients are kept in another small memory. The filtering operation is defined as follows.

$Y[I,J] = \sum_{i=-1}^{1} \sum_{j=-1}^{1} X[I+i, J+j] * C[i,j]$, $1 \le I \le m\text{-}2,\ 1 \le J \le n\text{-}2$
where X is the m x n matrix denoting the image to be filtered, Y is the m-2 x n-2 matrix denoting the filtered image, and C is the 3 x 3 filter coefficient matrix.

Make provision for at least two sets of filter coefficients shown below. The coefficients can be scaled up by a suitable power of 2 and rounded off to integers. The final results could be scaled down by the same factor, simply by doing a right shift. Use 8 bits for pixel values in the range 0 to 255.

Smoothening filter

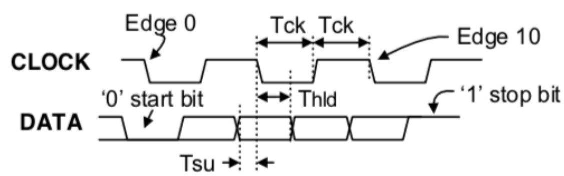
$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$

Sharpening filter

$$\begin{bmatrix} -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \\ -\frac{1}{9} & 1\frac{8}{9} & -\frac{1}{9} \\ -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \end{bmatrix}$$

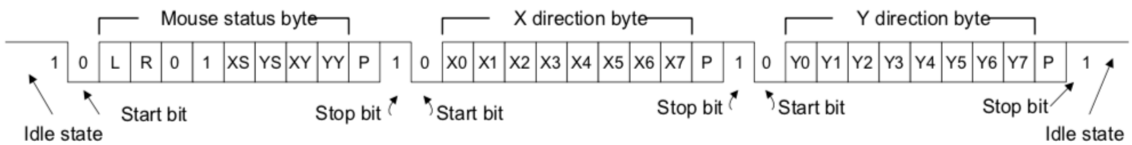
Mouse controlled display

BASYS 3 board has a built-in key-board and mouse interface. Standard USB key-board and mouse can be connected to the USB Host port (J2) on the board. This project is concerned only with mouse interface and use of mouse to control LED and 7-Seg displays on the board. The board provides logic to emulate PS/2 interface signals that are connected to FPGA. The data representing mouse movement is sent serially through this interface in the form of three 11 bit words, each including a start bit, data byte (LSB first), odd parity, and stop bit, as shown below.



Symbol	Parameter	Min	Max
T_{CK}	Clock time	30us	50us
T_{SU}	Data-to-clock setup time	5us	25us
T_{HLD}	Clock-to-data hold time	5us	25us

These three words are encoded as shown below.



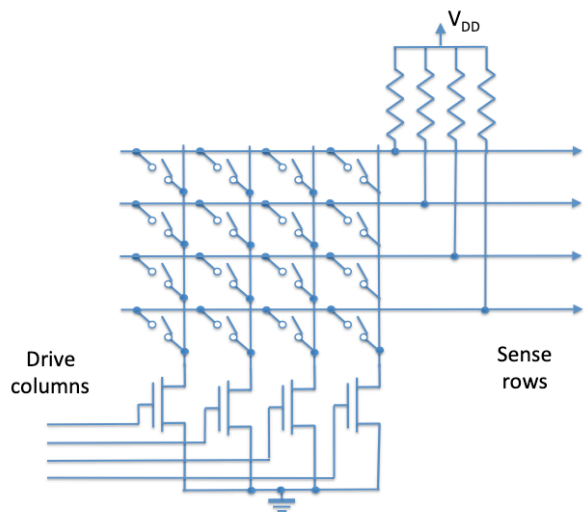
X0 .. X7 and Y0 .. Y7 indicate rate of mouse movement in X and Y directions. XS and YS indicate direction of movement. XV and YV are overflow indicators. L and R indicate pressing of left and right buttons.

Display some information on LEDs and 7-Seg displays and use mouse movement to control these. Left-right movement may be used to scroll the information and up-down movement may be used to control brightness.

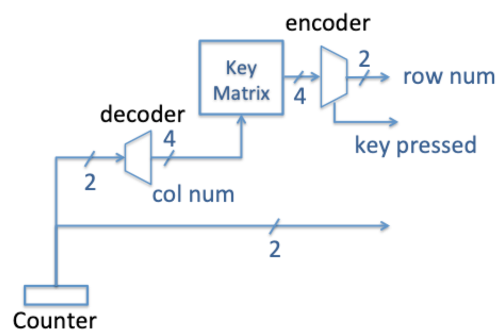
Currency exchange machine and Vending machine

These projects model the real machines where insertion of money is represented by key pressing on a 4x4 Key pad and the dispensed currency or goods are represented by 7-seg displays. While 7-seg display are built-in on the BASYS 3 board, the key pad is an external device available as a Pmod (see the picture on right).

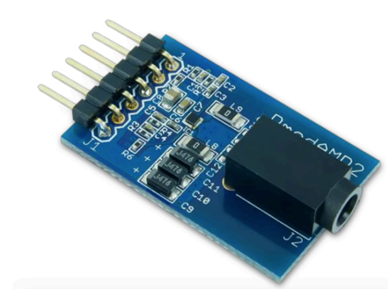
The 16 keys are arranged in the form of a 4x4 matrix as shown below. When a key is pressed, one column line gets connected to one row line.



To scan the entire key pad, the columns are driven to '0' one by one and the rows are read, as shown in the figure below. If no key is pressed, all rows are at '1' level. If a key is pressed, one of the rows is pulled down to '0' when the appropriate column is driven to '0'. The row number and the column number together identify the key.



Speech synthesis and Music synthesis



These projects aim at producing audio output using the audio amplifier Pmod (shown on right) connected to the BASYS 3 board. Input to this could be an analog signal or a pulse width modulated digital signal. It provides two levels of amplification.

For speech synthesis, store pre-recorded sound samples of spoken words in memory and construct phrases/sentences by playing these in a sequence, in response to some input on 4x4 Key pad (described above). For example, when keys 1, 2, 3 and 4 are pressed in sequence, sound for "one thousand two hundred thirty four" can be created by playing pre-stored sounds of the individual words – "one", "thousand", "two", "hundred", "thirty" and "four".

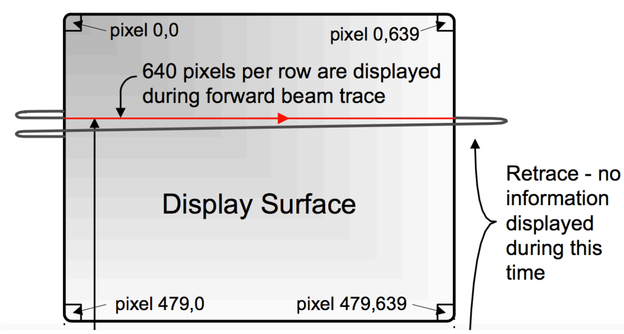
For music synthesis, designate the 16 keys as different musical notes. When a key is pressed, output samples of a sine wave of the frequency of the note corresponding to that key. You may experiment with addition of harmonics to get different types of sound, possibly sounds of different instruments.

Analyze tone/volume of sound

This project is being dropped because of its complexity. Allocation chart will be updated soon to show its replacements.

Bus/metro route display, Display graphics patterns

These projects focus use of a computer monitor as a display device for the FPGA board. BASYS 3 board has a VGA port for connecting to display devices compatible with VGA (Video Graphics Array) interface. This interface was designed for cathode ray tube (CRT) displays that use electron beam for scanning the display surface with a resolution of 640x480 pixels.



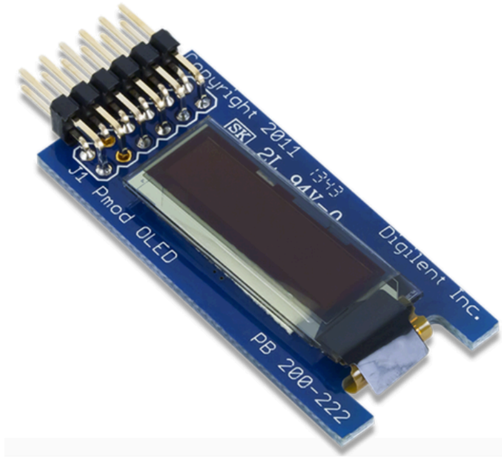
The information to be displayed can be transformed into a 640x480 pixel array and output to the VGA interface along with horizontal and vertical synchronizing signals. For details see the BASYS 3 reference manual.

Displaying text in air

The idea is to use persistence of vision to make the trail of a light visible even after it has changed. A row of LEDs mounted on a board can be moved at a suitable speed to create an effect of a matrix of LEDs. The scrolling of LEDs can be captured using a camera with long exposure settings. Examples of such experiments are available on the internet.

Ping-pong / Flappy bird / Shooting game

These are only some examples of simple games you can build using a display and an FPGA. We have OLED display Pmod providing 128x32 pixel monochrome display. You can use it to display some graphics or 4 lines of text, or some combination of graphics and text.



The module include a pixel buffer and a controller. The information to be displayed needs to be put in the form of an array of pixels and loaded into the pixel buffer. The controller takes care of refreshing the display. The controller supports horizontal and vertical scrolling of the display, but any other change in the display has to be achieved by explicitly modifying the pixel array and re-writing into the pixel buffer of Pmod.