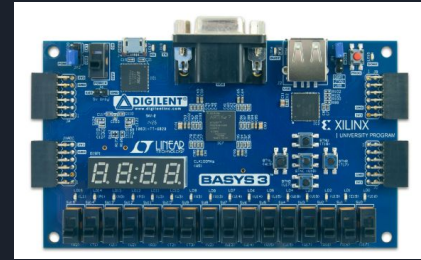


A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

# ECE 583 Final Project

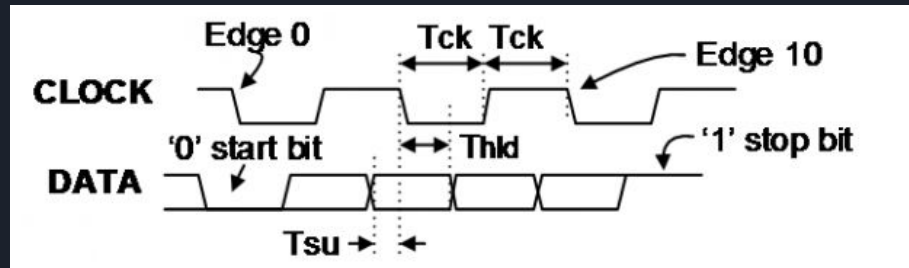
Ryan Conte and Robert Lee

# What does it do?



- We used the Basys 3 to create a small memory system, which the user interfaces with via the switches and any USB keyboard.
- The memory has 4-bit addresses, and 16-bit words. Each word is two ASCII characters input by the user from the keyboard.
- The rightmost 4 switches are used to specify the address of the memory contents on the 7-segment display.
- The contents of the 7-segment display can be saved to the selected address by pressing the ENTER key on the keyboard.
- The center button can be pressed to “pop” (value  $\geq 8$ )

# How does it work?



-USB HID (PS/2) ports “PS2Data” and “PS2Clk”

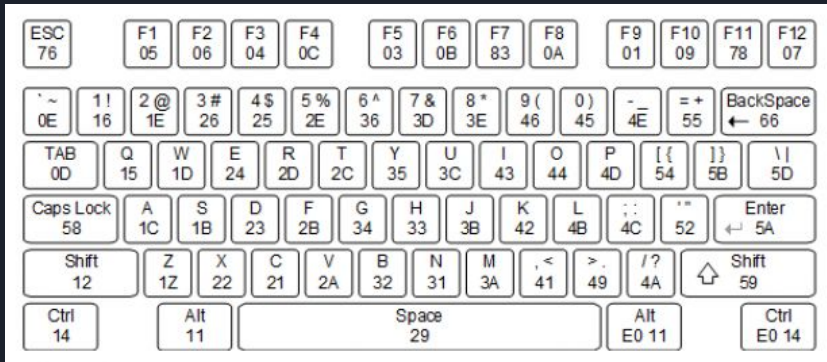
-Since we have a serial connection, only 1-bit is transmitted on each clock pulse.

-The data line is HIGH when not transmitting, and LOW signals the start of a transmission of an 11-bit word. These words contain 1 start bit, 8-bit keycode, an odd parity bit, and a stop bit.

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


# How does it work? (continued)

-When the key is released, 'F0' followed by the original key scan is given.



When the 'a' key is released, a scan code of "F0 1C" is sent over UART.





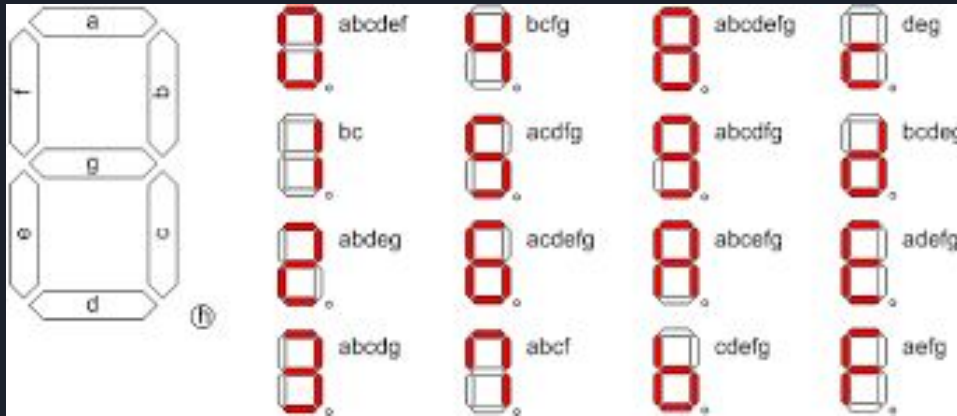
## How does it work? (continued)

- Since it re-sends the key scan code, and we only want one byte of data from each key press and release, the program ignores the first two word transmissions on each press.

- This was implemented by simply incrementing a word count variable whenever a stop bit is detected, and transmit only when `wordcount%3 == 0`.

# How does it work? (continued)

-The contents are displayed on the 7-segment display. The words fill in by shifting left by a byte to make space for the newly transmitted byte on the rightmost 2 displays. The “pop” performs the inverse, shifting to the right and filling in with a zero byte on the left.



Questions?

