COL216 Computer Architecture

Lab Assignments 13 and 14

ARM CPU with Input/Output

1. Objective

The objective of these assignment is to learn how to interface the 7-segment display on BASYS 3 board and a 4x4 key-pad on PmodKYPD to the ARM processor designed in the previous assignments. These assignments may be done after doing assignment 12 (full version or lite version) or directly after assignment 11.

2. Scope

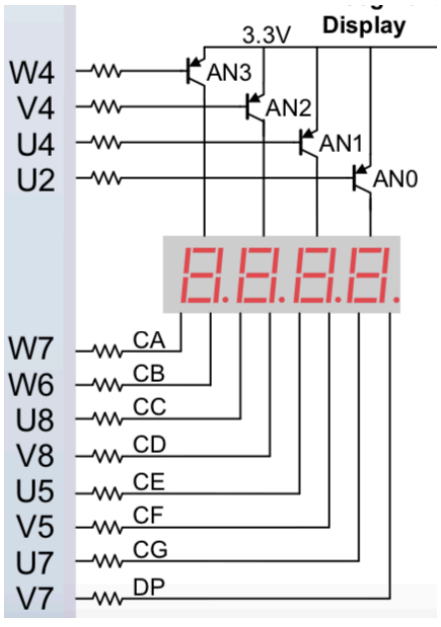
The scope of these assignments includes the following activities.

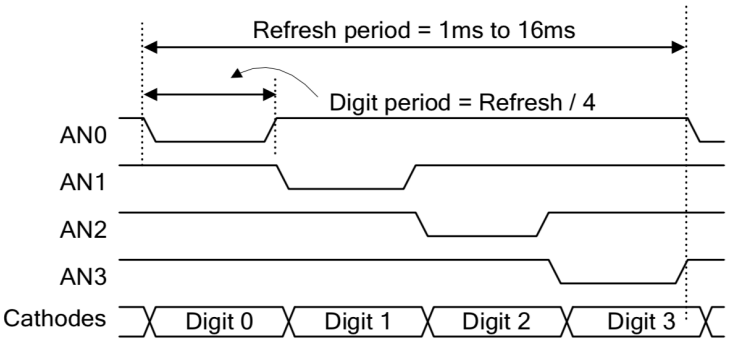
Assignment 13:
- Create hardware interface raw 7-segment display 4x4 keypad.
- Create SWI handlers or subroutines to perform I/O with these devices.

Assignment 14:
- Demonstrate use of these devices.

3. Display Interface

The hardware interface for display consists of two output ports – one to provide the anode pattern and the other to provide the cathode pattern. The anode pattern consists of three 1's and one 0. The position of 0 decides which one of the four digits is selected. The cathode pattern defines which LEDs are to be turned on for the selected digit. A '0' in this pattern turns the corresponding LED on and '1' turns it off.





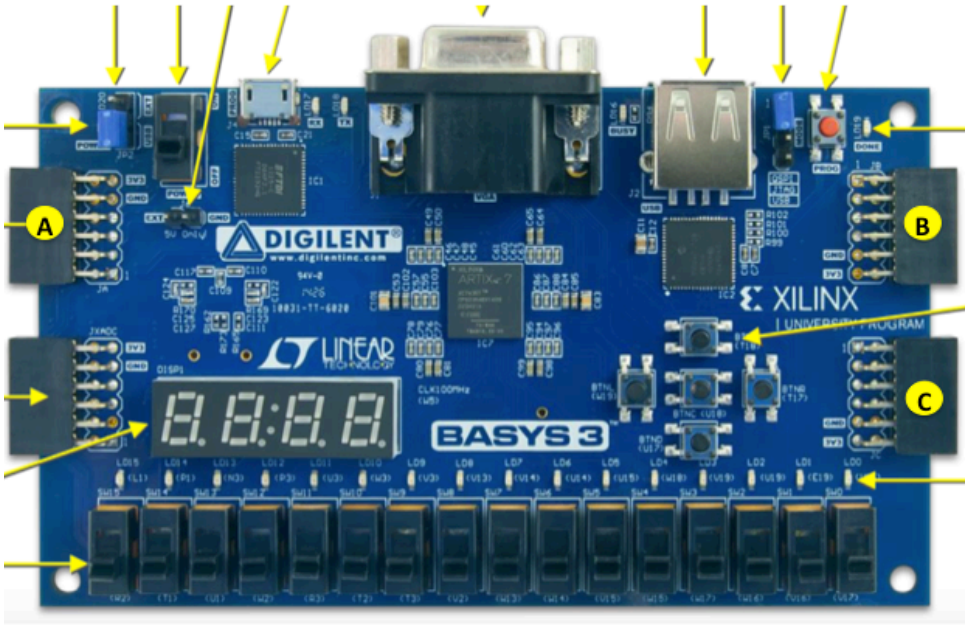
4. Key pad interface

PmodKYPD has 16 keys (0 – 9 and A – F), arranged as a 4x4 matrix as shown in the picture. These are scanned column-wise, applying a '0' to one of the columns and keeping the other three

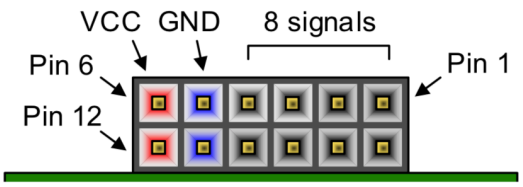
at ‘1’. The outputs observed at the rows are all 1’s by default. If a key on the selected column is pressed, the output for the corresponding row becomes ‘0’. Signals for this Pmod are shown in the table below. The columns are to be connected to an output port and the rows to an input port.

Header J1					
Pin	Signal	Description	Pin	Signal	Description
1	COL4	Column 4	7	ROW4	Row 4
2	COL3	Column 3	8	ROW3	Row 3
3	COL2	Column 2	9	ROW2	Row 2
4	COL1	Column 1	10	ROW1	Row 1
5	GND	Power Supply Ground	11	GND	Power Supply Ground
6	VCC	Power Supply (3.3V/5V)	12	VCC	Power Supply (3.3V/5V)

This module can be connected to any of the BASYS 3 board ports A, B or C of shown in the figure below.



Pin numbering convention on BASYS 3 board is shown in the figure below.

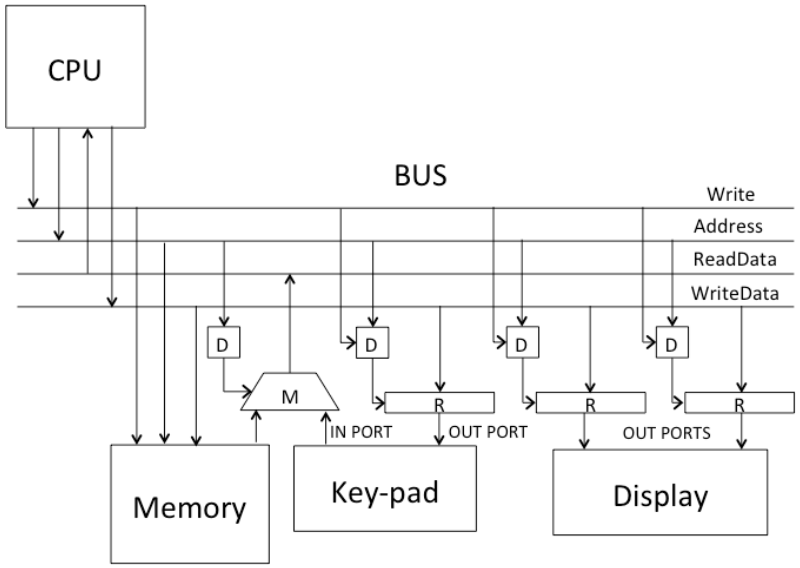


Signal names for various pins on the three ports, as required to be used in the xdc file are shown in the table on right.

Pmod JA	Pmod JB	Pmod JC
JA1: J1	JB1: A14	JC1: K17
JA2: L2	JB2: A16	JC2: M18
JA3: J2	JB3: B15	JC3: N17
JA4: G2	JB4: B16	JC4: P18
JA7: H1	JB7: A15	JC7: L17
JA8: K2	JB8: A17	JC8: M19
JA9: H2	JB9: C15	JC9: P17
JA10: G3	JB10: C16	JC10: R18

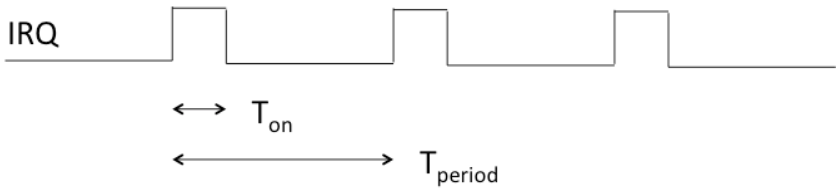
5. Creating I/O ports

Block diagram of the system including the two devices and I/O ports is shown below. The Key-pad requires a 4-bit input port to read column pattern and a 4-bit output port to drive columns. The Display requires a 4-bit output port to drive anodes and an 8-bit output port to drive cathodes. Blocks labeled R are registers (these are clock edge triggered, though clock is not shown) implementing the output ports. M is the multiplexer combining ReadData from memory and an input port. Blocks labeled D are address decoders used for selecting ports and memory. Keep aside some area in memory address space for I/O and assign addresses to the I/O ports in in this area. If protection has been implemented in assignment 12, keep this area within the supervisor area.



6. Display refresh and Key pad scan

Drive IRQ interrupt with a periodic signal and use IRQ handler for display refresh and key-pad scan. We assume that refreshing and scanning is done at the same rate. The figure below shows the waveform of this signal.



The T_{period} should be between 0.25 ms and 4.0 ms (see waveforms in section 3 above). T_{on} should be large enough so that no pulse is missed out and small enough so that no pulse causes multiple interrupts. Recall that IRQ is disabled when any exception handler is entered and remains disabled until it is explicitly enabled again by the handler. If a pulse falls entirely in this interval, it will not cause any interrupt. On the other hand, if the pulse is still there after exiting from IRQ handler, it will cause an interrupt again. One solution is to have IRQ driven by a flip-flop that is set by a signal with the right period and arbitrary T_{on}, and reset by a signal I_{ack} (interrupt acknowledge) which is '1' during the cycle in which IRQ is recognized.

If assignment 12 has not been implemented, IRQ needs to be polled through an additional (1-bit) input port.

7. I/O functions

Implement the following I/O functions as SWI calls, if assignment 12 has been implemented or as normal subroutines, otherwise.

Read_key: wait until a key is pressed and return 4-bit code of the key pressed

Input parameters: none

Output parameters: 4-bit key code

Display_1: display a BCD or hex digit in specified position

Input parameters: digit to be displayed, position

Output parameters: none

Display_4 digits: display a 4-digit BCD or hex number

Input parameters: 4-digit BCD or hex number

Output parameters: none

The Read_key function described above performs a *blocking read*. An alternative to this is a pair of functions – one checks whether a key has been pressed and the other performs a *non-blocking read*.

8. Demonstration program

Use a suitable program to demonstrate use of the key-pad and display.