International Journal of Computer Applications Technology and Research
Volume 5– Issue 3, 165 - 171, 2016, ISSN:- 2319–8656

Computer Interface for Electroluminescence (EL)

Ajay Kumar Mishra
Department. of Mathematics and Computer Science,
R.D. University,
Jabalpur, India

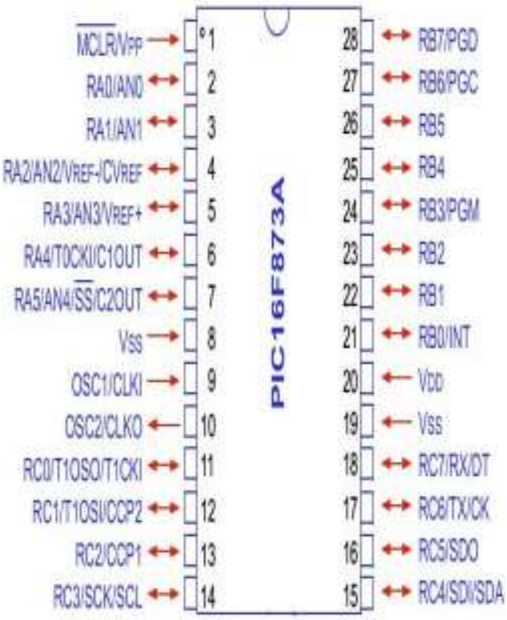
Abstract: The goal of Computer aided device start from the physical description of integrated circuit devices, considering both the physical configuration and related device properties and build the link between the broad range of physics and electrical behavior models that support circuit design. Physics-based modeling of devices, is distributed and lumped form is an essential part of the IC process development. It seeks to quantify the underlying understanding of the technology and abstract that knowledge to the device design level, including extraction of the key parameters that support circuit design and statistical metrology [1][2]. IC development for more than a quarter-century has been dominated by the MOS technology. In the 1970s and 1980s NMOS was favored owing to speed and area advantages, coupled with technology limitations and concerns related to isolation, parasitic effects and process complexity. During that era of NMOS-dominated LSI and the emergence of VLSI, the fundamental scaling laws of MOS technology were codified and broadly applied [3]. It was also during this period that Computer Aided Device reached maturity in terms of realizing robust process modeling (primarily one-dimensional) which then became an integral technology design tool, used universally across the industry [4]. At the same time device simulation, dominantly two-dimensional owing to the nature of MOS devices, became the work-horse of technologists in the design and scaling of devices [5]. The transition from NMOS to CMOS technology resulted in the necessity of tightly coupled and fully 2D simulators for process and device simulations [6][7].

Keywords: **Computer interface, interfacing, computer aided device.**

1. INTRODUCTION

In computer science, an interface is the point of interaction with software, or computer hardware, or with peripheral devices such as a computer monitor or a keyboard. Some computer interfaces such as a touch screen can send and receive data, while others such as a mouse or microphone can only send data [8]. A hardware interfaces exist in computing systems between many of the components such as the various buses, storage devices, other I/O devices, etc. A hardware interface is described by the mechanical, electrical and logical signals at the interface and the protocol for sequencing them (sometimes called signaling). A standard interface, such as SCSI decouples the design and introduction of computing hardware, such as I/O devices, from the design and introduction of other components of a computing system, thereby allowing users and manufacturer's greater flexibility in the implementation of computing systems. Hardware interfaces can be parallel where performance is important or serial where distance is important [9]. A computer network, also referred to as just a network consists of two or more computers and typically other devices as well (such as printers, external hard drives, modems and routers), that are linked together so that they can communicate with each other and thereby exchange commands and share data, hardware and other resources. The devices on a network are referred to as nodes. They are analogous to the knots in nets that have traditionally been used by fishermen and others. Nodes can be connected using any of various types of media, including twisted pair copper wire cable, optical fiber cable, coaxial cable and radio waves. And they can be arranged according to several basic topologies (i.e., layouts), including bus (in which all nodes are connected along a single cable), star (all nodes are connected to a central node), tree (nodes successively branch off from other nodes) and ring

2. DESCRIPTION OF HARDWARE USED



Pin Description of PIC16F873A microcontroller

Pin No.	Pin Name	Description
1	MCLR/VPP	Master Clear (input) or programming voltage

		(output). This pin is active low
2	RAO/ANO	Digital I/O. Analog input 0.
3	RA1/AN1	Digital I/O. Analog input 1.
4	RA2/AN2/VREF-/CVREF	Digital I/O. Analog input 2. A/D reference voltage (Low) input. Comparator VREF output.
5	RA3/AN3/VREF+	Digital I/O. Analog input 3. A/D reference voltage (High) input.
6	RA4/TOCKI/C1OUT	Digital I/O- Open –Drain when configured as output.
7	RA5/AN4/SS/C2OUT	Digital I/O. Analog Input 4. SPI Slave selects input. Comparator 2 output.
8	VSS	Ground reference for logic and I/O Pins.
9	OSC1/CLK1	Oscillator Crystal or external clock input. Oscillator crystal input or external clock source input
10	OSC2/CLKO	Oscillator Crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode.
11	RCO/T1OSO/T1CKI	Digital I/O. Timer1 oscillator output. Timer1 external clock input.
12	RC1/T1OSI/CCP2	Digital I/O. Timer1 oscillator input. Capture2 input, compare2 output, PWM2 output.
13	RC2/CCP1	Digital I/O. Capture1 input, compare 1 output, PWM 1 output.
14	RC3/SCK/SCL	Digital I/O. Synchronous serial Clock input/output for SPI mode.
		Synchronous serial Clock input/output for I^2C mode.
15	RC4/SDI/SDA	Digital I/O. SPI data in. I^2C data I/O.
16	RC5/SDO	Digital I/O. SPI data out.
17	RC6/TX/CK	Digital I/O. USART asynchronous transmit. USART1 synchronous clock.
18	RC7/RX/DT	Digital I/O. USART asynchronous receive. USART synchronous data.
19	VSS	Ground reference for logic and I/O Pins.
20	VDD	Positive Supply for logic and I/O Pins.
21	RBO/INT	Digital I/O. External Interrupts
22	RB1	Digital I/O.
23	RB2	Digital I/O.
24	RB3/PGM	Digital I/O. Low- voltage (single-supply) ICSP programming enable pin.
25	RB4	Digital I/O.
26	RB5	Digital I/O.
27	RB6/PGC	Digital I/O. In-circuit debugger and ICSP programming clock.
28	RB7/PGD	Digital I/O. In-circuit debugger and ICSP programming data.

MICROCONTROLLER PROGRAMING

PIC16F873A microcontroller used program code as given below

………………………………………………………………… ……………………………

LIST P=PIC16F873A

INCLUDE "P16F873A.INC"

BANK0 EQU 20H

```
        CBLOCK BANK0
        LOWERLSB
        HIGHER
        UNIT
        TEN
        HUND
        R1
        R2
        REQUEST
        HEX
        THOU
        TEMP
        DIGIT_SEL
        DIGIT_DISP
        DIGIT_OUT
        REPEAT
        ENDC
        ORG 0X000
RVRESET
        GOTO START
        ORG 0X004
RVINT
        BTFSS PIR1, ADIF
        GOTO $-1
        BCF PIR1, ADIF
        CALL ADC_INT
        GOTO PROGRAM
        RETFIE

START
        CALL SET_PORTS
        MOVLW 0X0F
        MOVWF REPEAT
        GOTO PROGRAM
SET_PORTS
        ;CALL SET_PORTA
        CALL SET_PORTB
        CALL SET_PORTC
        RETURN
SET_PORTA
        BANKSEL ADCON1
        MOVLW 0X06
        MOVWF ADCON1
        BANKSEL TRISA
        MOVLW 0X3F
        MOVWF TRISA
        BANKSEL PORTA
        CLRF PORTA
        RETURN
SET_PORTB
        BANKSEL TRISB
        CLRF TRISB
        BANKSEL PORTB
        CLRF PORTB
        RETURN
SET_PORTC
        BANKSEL TRISC
        CLRF TRISC
        BANKSEL PORTC
        CLRF PORTC
        RETURN
PROGRAM
        CALL SET_PORTB
        CALL SET_PORTC
        CALL HEX_TO_BCD
        CALL NUMBER
        CALL DISPLAY
        DECFSZ REPEAT, F
```

```
        GOTO $-2
        MOVLW 0X0F
        MOVWF REPEAT
        CALL REQUEST_1
        CALL ADC_1
        GOTO PROGRAM
REQUEST_1
        MOVLW 0X01
        MOVWF REQUEST
        RETURN
REQUEST_2
        MOVLW 0X09
        MOVWF REQUEST
        RETURN
REQUEST_3
        MOVLW 0X11
        MOVWF REQUEST
        RETURN
REQUEST_4
        MOVLW 0X19
        MOVWF REQUEST
        RETURN
ADC_1
        MOVLW 0XC0
        MOVWF INTCON
        MOVF REQUEST, W
        MOVWF ADCON0
        CLRF PORTA
        CLRF ADRESH
        BANKSEL PIE1
        CLRF PIE1
        BSF PIE1, ADIE
        MOVLW 0X40
        MOVWF ADCON1
        CLRF TRISA
        COMF TRISA, F
        CLRF ADRESL
        NOP
        NOP
        NOP
        NOP
        BANKSEL ADCON0
        MOVLW 0X05
        MOVWF ADCON0
        GOTO $

ADC_INT
        CALL LOW_1
        CALL HIGH_1
        RETURN
LOW_1
        BANKSEL ADRESL
        MOVF ADRESL, W
        MOVWF LOWERLSB
        RETURN
HIGH_1
        BANKSEL ADRESH
        MOVF ADRESH, W
        MOVWF HIGHER
        MOVWF HEX
        RETURN
OUTPUT
        BANKSEL TRISB
        CLRF TRISB
        BANKSEL PORTB
        MOVF HIGHER, W
        MOVWF PORTB
        RETURN
```

```
NUMBER
        SWAPF UNIT, W
        ANDLW 0X0F
        MOVWF TEN
        MOVLW 0X0F
        ANDWF UNIT, F
        RETURN
HEX_TO_BCD
        BANKSEL STATUS
        BCF STATUS, 0   ;1 shift
        RLF HEX, F
        RLF UNIT, F
        BCF STATUS, 0           ;2 shift
        RLF HEX, F
        RLF UNIT, F
        BCF STATUS, 0           ;3 shift
        RLF HEX, F
        RLF UNIT, F
        CALL UNIT_5
        BCF STATUS, 0           ;4 shift
        RLF HEX, F
        RLF UNIT, F
        CALL UNIT_5
        BCF STATUS, 0           ;5 shift
        RLF HEX, F
        RLF UNIT, F
        CALL UNIT_5
        BCF STATUS, 0           ;6 shift
        RLF HEX, F
        RLF UNIT, F
        CALL UNIT_5
        CALL TEN_5

        BCF STATUS, 0           ;7 shift
        RLF HEX, F
        RLF UNIT, F
        RLF HUND, F
        CALL UNIT_5
        CALL TEN_5
        BCF STATUS, 0           ;8 shift
        RLF HEX, F
        RLF UNIT, F
        RLF HUND, F
        RETURN
UNIT_5
        MOVF UNIT, W
        ANDLW 0X0F
        MOVWF TEMP
        MOVLW 0X05
        SUBWF TEMP, W
        BANKSEL STATUS
        BTFSS STATUS, 0
        RETURN
        MOVLW 0X03
        ADDWF UNIT, F
        RETURN
TEN_5
        MOVF UNIT, W
        ANDLW 0XF0
        MOVWF TEMP
        MOVLW 0X50
        SUBWF TEMP, W
        BANKSEL STATUS
        BTFSS STATUS, 0
        RETURN
        MOVLW 0X30
        ADDWF UNIT, F
        RETURN
```

```
DISPLAY
        CLRF DIGIT_SEL
        BCF DIGIT_SEL, 7
        BSF DIGIT_SEL, 4
        MOVF UNIT, W
        MOVWF DIGIT_DISP
        CALL S_S_DECODER
        MOVWF DIGIT_OUT
        CALL DIGIT_SELECT
        CALL OUT_TO_FND
        CALL DELAY
        BCF DIGIT_SEL, 4
        BSF DIGIT_SEL, 5
        MOVF TEN, W
        MOVWF DIGIT_DISP
        CALL S_S_DECODER
        MOVWF DIGIT_OUT
        CALL DIGIT_SELECT
        CALL OUT_TO_FND
        CALL DELAY

        BCF DIGIT_SEL, 5
        BSF DIGIT_SEL, 6
        MOVF HUND, W
        MOVWF DIGIT_DISP
        CALL S_S_DECODER
        MOVWF DIGIT_OUT
        CALL DIGIT_SELECT
        CALL OUT_TO_FND
        CALL DELAY
        BCF DIGIT_SEL, 6
        BSF DIGIT_SEL, 7
        MOVF THOU, W
        MOVWF DIGIT_DISP
        CALL S_S_DECODER
        MOVWF DIGIT_OUT
        CALL DIGIT_SELECT
        CALL OUT_TO_FND
        CALL DELAY
        GOTO DISPLAY
        RETURN
S_S_DECODER
        ;Display code table......................................
        MOVF DIGIT_DISP, W ; Get key count
        ADDWF PCL ; and calculate jump
        ;NOP ; into table
        RETLW B'11011110' ; Code for '0'
        RETLW B'01000010' ; Code for '1'
        RETLW B'11101100' ; Code for '2'
        RETLW B'11100110' ; Code for '3'
        RETLW B'01110010' ; Code for '4'
        RETLW B'10110110' ; Code for '5'
        RETLW B'10111110' ; Code for '6'
        RETLW B'11000010' ; Code for '7'
        RETLW B'11111110' ; Code for '8'
        RETLW B'11110110' ; Code for '9'
        ;Output display code.....................................
        RETURN
OUT_TO_FND
        MOVF DIGIT_OUT, W
        MOVWF PORTB
        RETURN
DIGIT_SELECT
        CLRF PORTB
        MOVF DIGIT_SEL, W
        MOVWF PORTC
        RETURN
DELAY
```

```
        MOVLW 0X0F
        MOVWF R1
        MOVWF R2
        DECFSZ R2, F
        GOTO $-1
        DECFSZ R1, F
        GOTO $-4
        RETURN
DELAY_L
        MOVLW 0XFF
        MOVWF R1
        MOVWF R2
        DECFSZ R2,F
        GOTO $-1
        DECFSZ R1,F
        GOTO $-4
        RETURN
                END
```

3. MEASUREMENT OF ELECTROLUMINESCENCE (EL)

For the measurement of EL brightness EL cell is prepared by selected method and it is ready to use for measurement of luminescence. In prepared cell, there are two electrodes, one from conducting glass plate and other from phosphor sample side. Frequency is applied to amplifier at the desired level and it increases voltage up to required level, when frequency and voltage are reached at certain level, then emission of EL is takes place the emission of light is connected to Photomultiplier tube (PMT) and it converts the light in the form of current and then converted to voltage by voltage to current converter. The output voltage is apply to PIC16F873A microcontroller and it convert this analog voltage to Digital voltage. The digital voltage to be interfaced with the computer by parallel port. The data of parallel port is read by VB.NET software and finally luminescence of EL cell is converted in voltage, and is obtained at the screen of the computer.

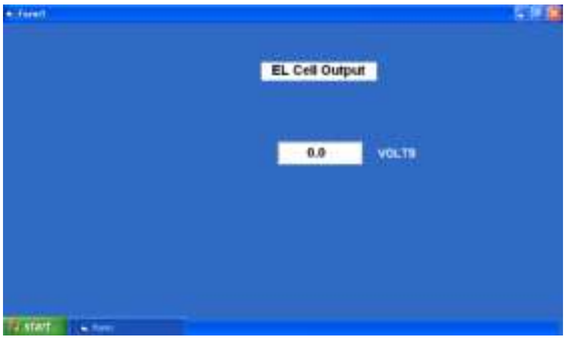


Figure 1 Sample of Interfacing Window

4. REFERENCES

[1] H.J. DeMan and R. Mertens, SITCAP--A simulator of bipolar transistors for computer-aided circuit analysis programs, International Solid-State Circuits Conference (ISSCC), Technical Digest, pp. 104-5, February, 1973

[2] R.W. Dutton and D.A. Antoniadis, Process simulation for device design and control, International Solid-State Circuits Conference (ISSCC), Technical Digest, pp. 244-245, February, 1979

[3] R.H. Dennard, F.H. Gaensslen, H.N. Yu, V.L. Rodeout, E. Bassous and A.R. LeBlanc, Design of ion-implanted MOSFETs with very small physical dimensions, IEEE Jour. Solid-State Circuits, vol. SC-9, pp.256-268, October, 1974.

[4] R.W. Dutton and S.E. Hansen, Process modeling of integrated circuit device technology, Proceeding IEEE, vol. 69, no. 10, pp. 1305-1320, October, 1981.

[5] P.E. Cottrell and E.M. Buturla, "Two-dimensional static and transient simulation of mobile carrier transport in a semiconductor," Proceedings NASECODE I (Numerical Analysis of Semiconductor Devices), pp. 31-64, Boole Press, 1979.

[6] C.S. Rafferty, M.R. Pinto, and R.W. Dutton, Iterative methods in semiconductor device simulation, IEEE Trans. Elec. Dev., vol. ED-32, no.10, pp.2018-2027, October, 1985.

[7] M.R. Pinto and R.W. Dutton, Accurate trigger condition analysis for CMOS latchup, IEEE Electron Device Letters, vol. EDL-6, no. 2, February, 1985.

[8] IEEE 100 - The Authoritative Dictionary Of IEEE Standards Terms. NYC, NY, USA: IEEE Press. 2000. pp. 574–575. ISBN 0-7381-2601-2.

[9] Blaauw, Gerritt A.; Brooks, Jr., Frederick P. (1997), "Chapter 8.6, Device Interfaces", Computer Architecture-Concepts and Evolution, Addison-Wesley, pp. 489–493, ISBN 0-201-10557-8 See also:

Patterson, David A.; Hennessey, John L. (2005), "Chapter 8.5, Interfacing I/O Devices to the Processor, Memory and Operating System", Computer Organization and Design - The Hardware/Software Interface, Third Edition, Morgan Kaufmann, pp. 588–596, ISBN 1-55860-604-1