King Saud University
Collage of Applied Medical Sciences
Biomedical Technology Department

BMT485
Level9
Lab2

Interfacing to the standard parallel port using C language

Objectives

Upon completion of this lab you'll be able to understand the following points

1. **How to convert analog signal to digital representation using the Analog to Digital Converter ADC0808**
2. **How to convert a digital representation to analog signal using the Digital to analog Converter DAC0808**
3. **Understand the standard parallel port (SPP)**

Pin out

Port Addresses

I/O using DOS Debug

4. **An introduction to C language**
5. **How to input 8 bits using the status port of the SPP using C language**
6. **Controlling and reading ADC0808 using the SPP & C language**

Introduction

This lab gives an overview of how to design a data acquisition system that incorporate ADC and DAC to convert data from the analog form to the digital form and vice versa. Also it demonstrates how to use the PC to control these devices and processes the data in and out using one of the most popular programming languages "C". This lab introduces new terms such as sampling time and conversion time deep understanding of these terms are must.

Pre-requisite

"Subjects to be understood before launch this lab"

No special pre-requisites are needed however the measuring skills of current, voltage, and resistance using the multi-meter are must

Required equipments

- **Multi-meter**
- **Oscilloscope**
- **Function generator**
- **Logic Analyzer**

Experiment steps

We are going now to discuss the devices (ICs) that are capable of converting analog signal to digital form and vice versa

ADC0808

8-Bit 8-channel µP Compatible A/D Converters with Multiplexer

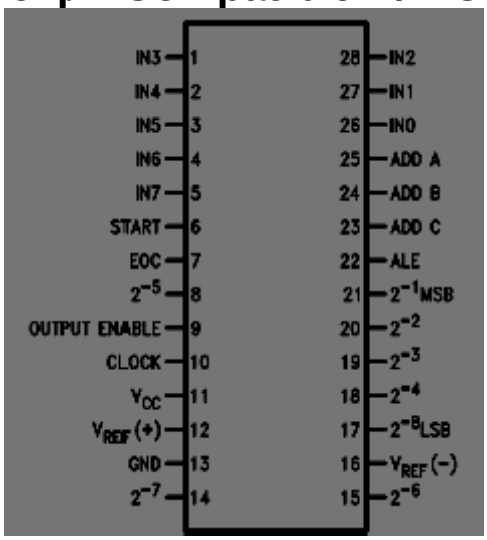
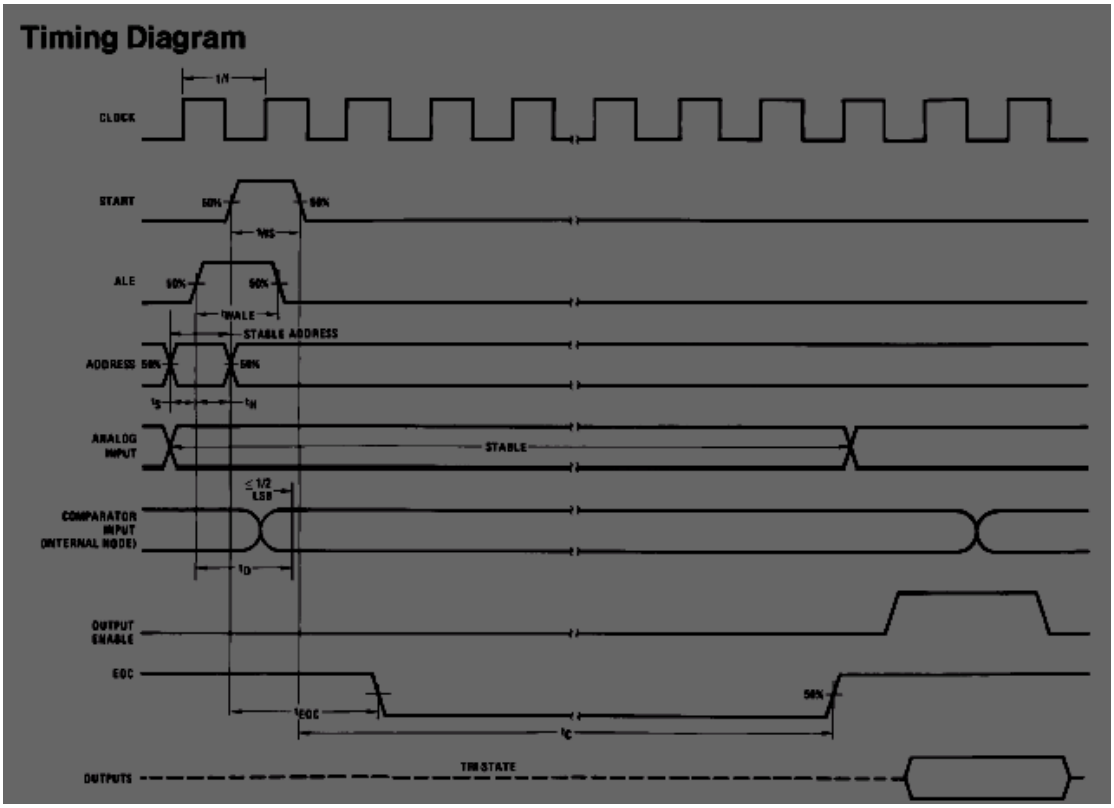


Figure 1

Specs

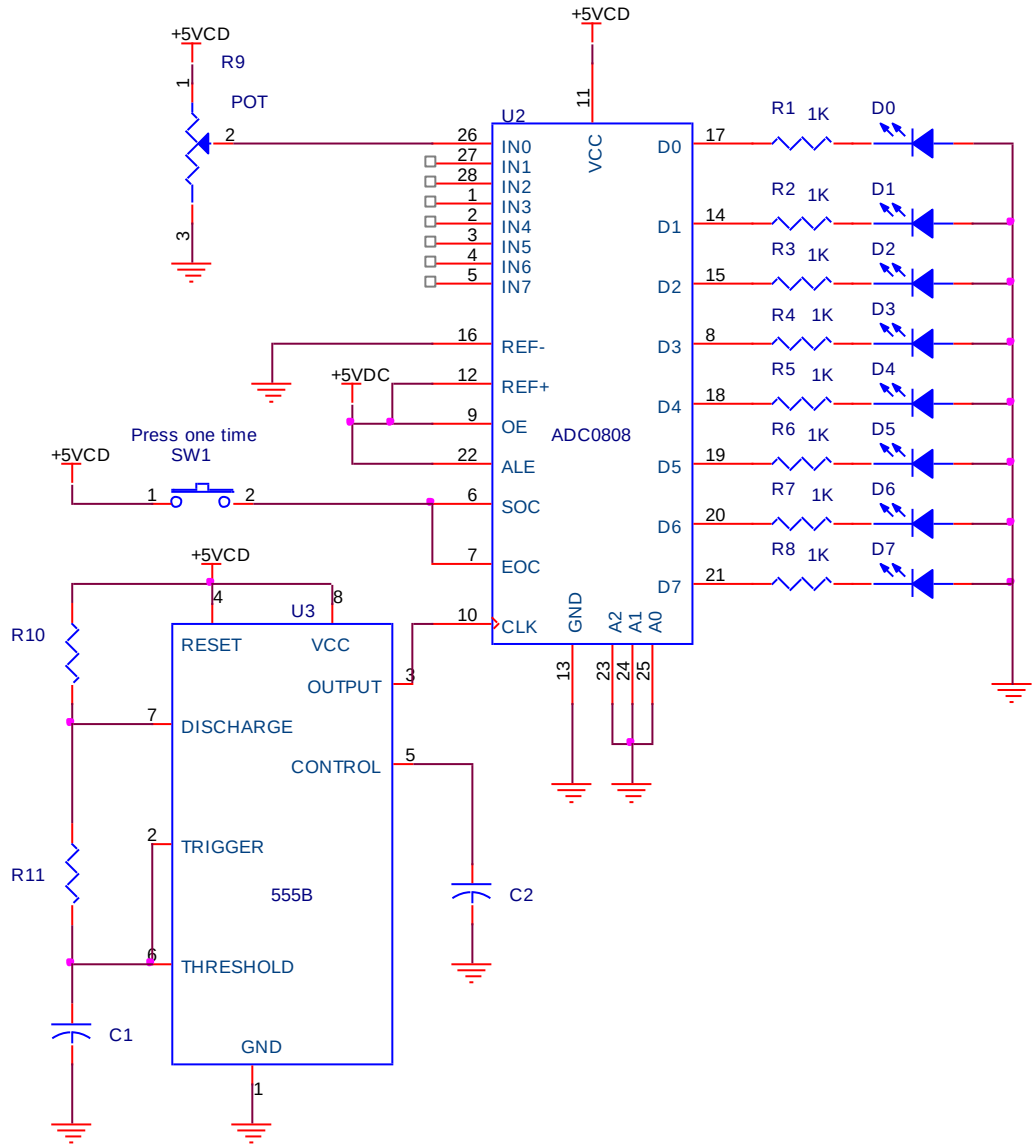
- **Easy interface to all microprocessors**
- **Operates ratio metrically or with 5 VDC voltage reference**
- **No zero or full-scale adjust required**
- **8-channel multiplexer with address logic**
- **0V to 5V input range with single 5V power supply**
- **Resolution: 8 Bits**
- **Total Unadjusted Error: ± 1 LSB**
- **Single Supply: 5 VDC**
- **Low Power: 15 mW**
- **Conversion Time: 100 µs**

Timing Diagram

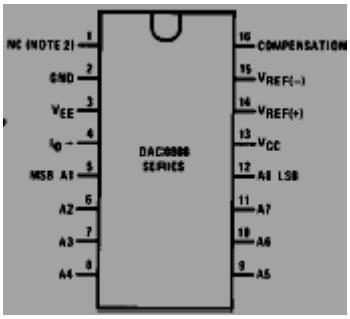


Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{WS}	Minimum Start Pulse Width	*(Figure 5)*		100	200	ns
t_{WALE}	Minimum ALE Pulse Width	*(Figure 5)*		100	200	ns
t_S	Minimum Address Set-Up Time	*(Figure 5)*		25	50	ns
t_H	Minimum Address Hold Time	*(Figure 5)*		25	50	ns
t_D	Analog MUX Delay Time From ALE	$R_S = 0\Omega$ *(Figure 5)*		1	2.5	μS
t_{H1}, t_{H0}	OE Control to Q Logic State	$C_L = 50$ pF, $R_L = 10$k *(Figure 8)*		125	250	ns
t_{1H}, t_{0H}	OE Control to Hi-Z	$C_L = 10$ pF, $R_L = 10$k *(Figure 8)*		125	250	ns
t_C	Conversion Time	$f_C = 640$ kHz, *(Figure 5)* (Note 7)	90	100	116	μS
f_C	Clock Frequency		10	640	1280	kHz
t_{EOC}	EOC Delay Time	*(Figure 5)*	0		8 + 2 μS	Clock Periods

FREE RUNNING ADC0808

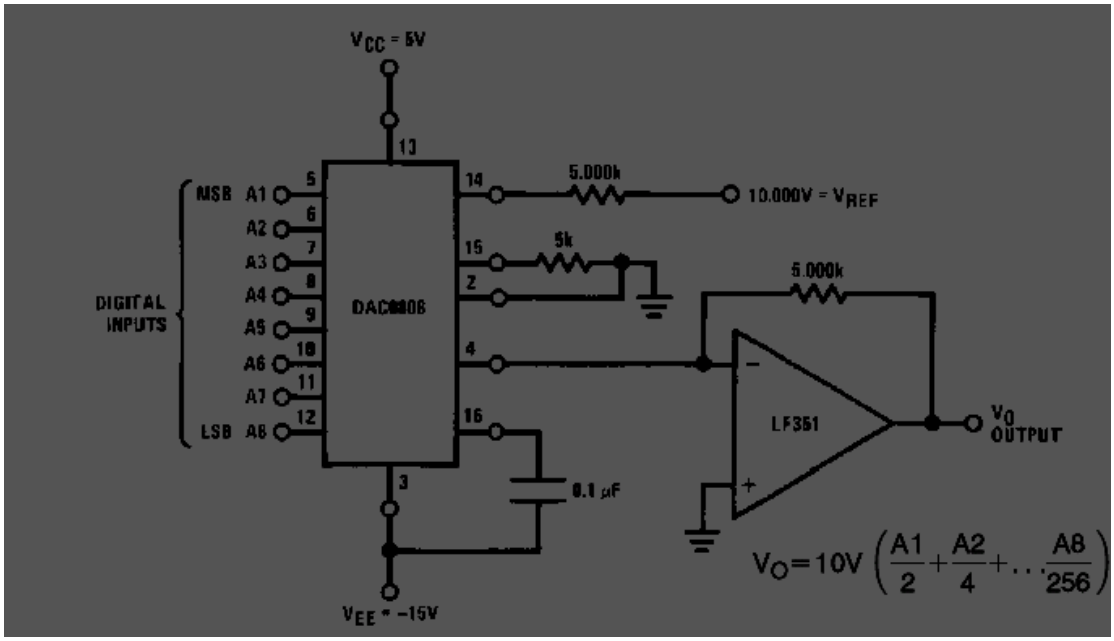


DAC0808



Specs

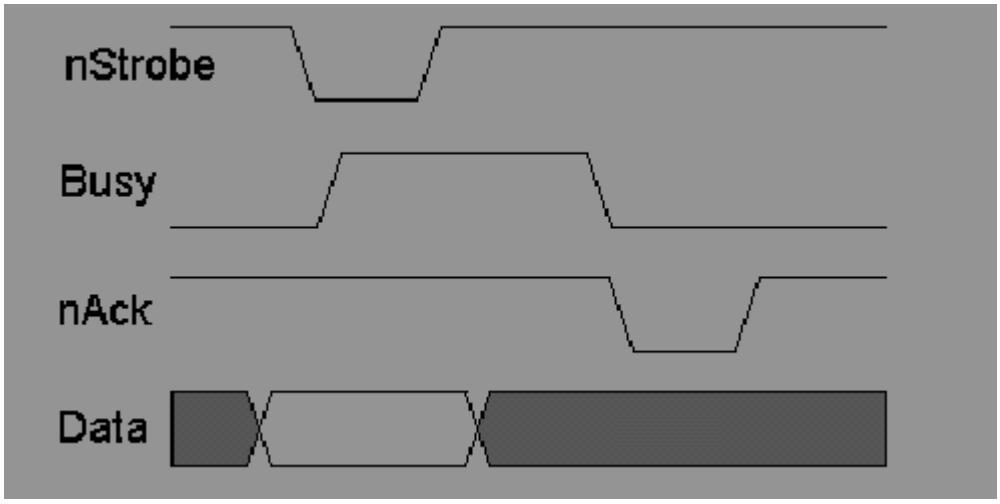
- **Full scale current match: ±1 LSB**
- **Fast settling time: 150 ns**
- **Power supply voltage range: ±4.5V to ±18V**
- **Low power consumption: 33 mW 5V**



Understand the standard parallel port (SPP)
Pin out

Pin No (D-Type 25)	Pin No (Centronics)	SPP Signal	Direction In/out	Register	Hardware Inverted
1	1	nStrobe	In/Out	Control	Yes
2	2	Data 0	Out	Data	
3	3	Data 1	Out	Data	
4	4	Data 2	Out	Data	
5	5	Data 3	Out	Data	
6	6	Data 4	Out	Data	
7	7	Data 5	Out	Data	
8	8	Data 6	Out	Data	
9	9	Data 7	Out	Data	
10	10	nAck	In	Status	
11	11	Busy	In	Status	Yes
12	12	Paper-Out PaperEnd	In	Status	
13	13	Select	In	Status	
14	14	nAuto-Linefeed	In/Out	Control	Yes
15	32	nError / nFault	In	Status	
16	31	nInitialize	In/Out	Control	
17	36	nSelect-Printer nSelect-In	In/Out	Control	Yes
18 - 25	19-30	Ground	Gnd		

PC to printer interfacing



Port Addresses

Address	Notes:
3BCh - 3BFh	Used for Parallel Ports which were incorporated in to Video Cards and now, commonly an option for Ports controlled by BIOS. - Doesn't support ECP addresses.
378h - 37Fh	Usual Address For LPT 1
278h - 27Fh	Usual Address For LPT 2

Port addresses in BIOS

Start Address	Function
0000:0408	LPT1's Base Address
0000:040A	LPT2's Base Address
0000:040C	LPT3's Base Address
0000:040E	LPT4's Base Address (Note 1)

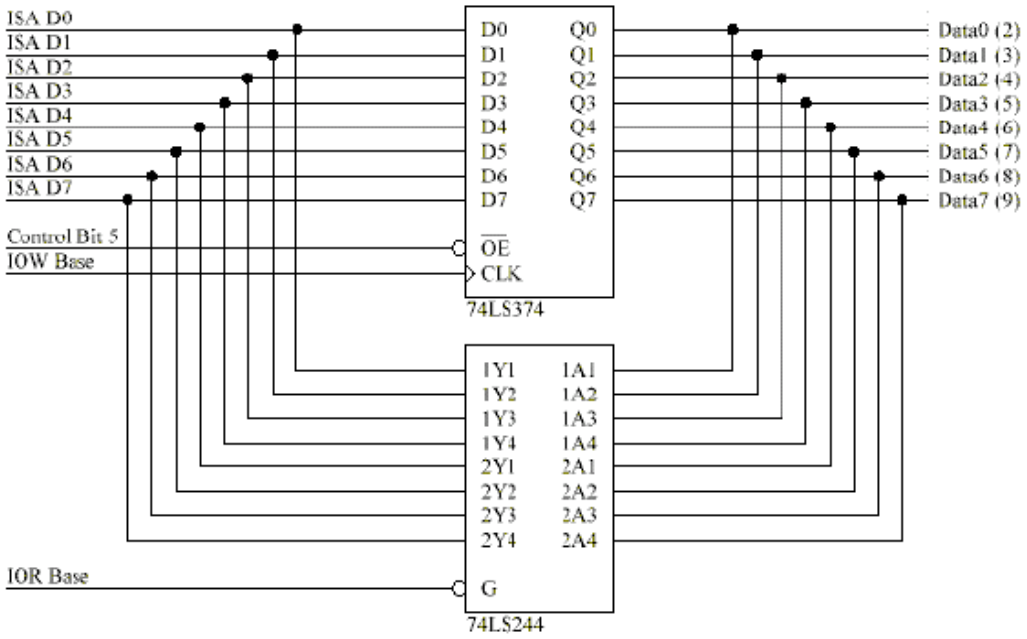
Read the parallel port addresses from the BOIS data area

```
#include <stdio.h>
#include <dos.h>
Void main (void)
{
unsigned int far *ptraddr;/* Pointer to location of Port Addresses */
unsigned int address; /* Address of Port */
int a;
ptraddr=(unsigned int far *)0x00000408;
for (a = 0; a < 3; a++)
{
address = *ptraddr;
if (address == 0)
printf("No port found for LPT%d \n",a+1);
else
printf("Address assigned to LPT%d is %Xh\n",a+1,address);
*ptraddr++;
}
}
```

Software Registers - Standard Parallel Port (SPP)

Offset	Name	Read/Write	Bit No.	Properties
Base + 0	Data Port	Write (Note-1)	Bit 7	Data 7 (Pin 9)
			Bit 6	Data 6 (Pin 8)
			Bit 5	Data 5 (Pin 7)
			Bit 4	Data 4 (Pin 6)
			Bit 3	Data 3 (Pin 5)
			Bit 2	Data 2 (Pin 4)
			Bit 1	Data 1 (Pin 3)
			Bit 0	Data 0 (Pin 2)
Base + 1	Status Port	Read Only	Bit 7	Busy
			Bit 6	Ack
			Bit 5	Paper Out
			Bit 4	Select In
			Bit 3	Error
			Bit 2	IRQ (Not)
			Bit 1	Reserved
			Bit 0	Reserved
Base + 2	Control Port	Read/Write	Bit 7	Unused
			Bit 6	Unused
			Bit 5	Enable bi-directional Port
			Bit 4	Enable IRQ Via Ack Line
			Bit 3	Select Printer
			Bit 2	Initialize Printer (Reset)
			Bit 1	Auto Linefeed
			Bit 0	Strobe

Bidirectional Standard Parallel Port



Controlling and reading ADC0808 using the SPP & C language

"EXPLAIN IT STEP BY STEP BY STEP"

“USE THE ORCAD OUTPUT FOR MORE CLEAR DEMONSTRATION”