Document 1: Computer Interfacing /4th / CSE-UOT / 2019-2020 (Dr L.J.S.)

Control and systems Eng. Dept. – University of Technology
4th year-Computer Interfacing Course
1st term course – 2019/2020
Instructor: Dr Laith J. Saud

****

This document includes:

- Course aim.

- Course Syllabus.

- Introductory Abstracted Lecture Notes – Part 1

<u>SYLLABUS</u> (30 Hrs)

1. Introductory Part: (6 Hrs)
    - Basic Microcomputer Components and Architecture.
    - Buses.
    - Bus Architecture and Types of Buses.
      Parallel and Serial Ways of Exchanging Information.
    - Timing Diagram.
    - S/W and H/W Concerns.
2. Using Some Standard Buses: (4Hrs)

- ISA.
- PCI.

3. Designing Communication & Interfacing Circuits: (6Hrs.)
   (Using Microcomputer Buses)
4. Making Use of Already Existing I/O Ports in the (4Hrs)
   Microcomputer fo Communication and Dealing With
   Some Standard I/O Circuits:
5. Interfacing Peripherals to the Microcomputer. (4Hrs.)
6. Requirements For Using Microcomputer for Data (4Hrs.)
   Acquisition and Control.
7. Some Notes About Some Extra Related Matters: (2Hrs)

- Practical Considerations to Care in Interfacing.
- RTS Programming Language.

INTRODUCTORY ABSTRACTED LECTURE NOTES

Lecture 1

Interface definition:

Interfacing is the coupling between a system under consideration and another system or between devices of a system, through which information passes.

Applications of digital computers

Digital computers are used in a huge number of applications because of the different jobs they can perform. Some of the applications of digital computers are: monitoring and data acquisition, control, computer aided design, data bases, ... etc.

Advantages of digital computers over analogue ones

Among the basic advantages of digital computers, are:

1. Digital computers can implement complex algorithms with constant accuracy at high speed.

2. The digital signals are less affected by noise compared with analogue ones.

3. Digital computers are extremely versatile.

4. Digital computers size is continuously decreasing with recent developments in manufacturing and design technology. And so are their prices.

Digital computers in the control field

One of the important application of digital computers is in systems control. Most of the recent applications (if not all) use digital computers instead of analog ones. A digital computer deals with digital quantities (represents data in digital form) rather than analog quantities being dealt with in analog computers. Another thing to keep in mind when dealing with digital computers is that digital computers deal with discrete signals. This is due to the way in which the digital computers process signals. This way is as follows:

The digital computer takes a signal sample, processes it, gives the result, and goes for another sample and so on.

Requirements for Microcomputer based monitoring and control systems:

These requirements come into three major parts, theoretical, hardware, and software. These issues are interacting ones.

The subjects involved in these issues are many and depends on the application nature, and below are just few to mention:

Theory H/W S/W

Hardware requirements

Connecting a digital computer to a process or system being under control or monitoring, always, cannot be done directly. An interface circuit or system is usually used for this purpose. This interface circuit involves different hardware parts, like data latches, buffers, A/D and D/A converters, sample and hold units, real time clocks, decoders, transducers, signal modification circuits, ..., etc.

Software requirements

In digital control systems, almost every action is under control and timing with the aid of programs written for this purpose. These actions might be like: transfer of data between the computer and the outside system, doing certain mathematical or logical functions, controlling the hardware parts of the interfacing circuit....etc. These programs might be written in high-level or in low-level languages depending on the given problem or application. When the speed of execution is a critical point, of course we need the low-level language. The low-level language depends on the type of the microprocessor (μP). Each (μP) has its own assembly language.

Theoretical requirements

A) **Sampling rate:**

Due to the discrete way of digital computers in dealing with signals, as mentioned earlier, there will be a certain time between processing one sample and another. The minimum period of this time depends on the time required by the computer to process the sample. On the other hand, the maximum period can be controlled by us through hardware or software arrangements. But, of course there is a limitation on this maximum period which we call sampling period. Shannon criterion states, that the sampling rate of a given signal must be equal to, or larger than twice the maximum frequency in the signal. Many control books give the number ten instead of two in order to be on the safe side. If we are talking about systems, one rough and acceptable way of choosing the sampling period is, to take it ten times smaller than the smallest time constant for the system.

B) Difference equation:

Dynamic behavior of continuous control systems is described by differential equations, while in discrete control systems it is described by difference equations.

C) Z-Transformation

The role of this transformation in discrete time systems is similar to the role of Laplace transformation in continuous time systems. That is here, to convert difference equations into algebraic ones.

D) Quantization

This term means converting a signal from analog to digital form. Quantizing analog signals in magnitude, causes error. The error could be reduced by increasing the number of bits used for digital representation of signals.

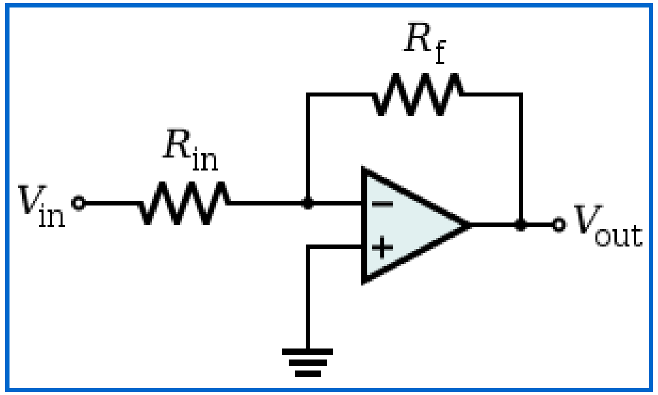
E) Data representation

Signals in digital control systems take different formats as it move from one point to another. And we have to trace and care for these changes (which the signals undergo) in our design procedure, otherwise, the control will fail for sure.

Lecture 2

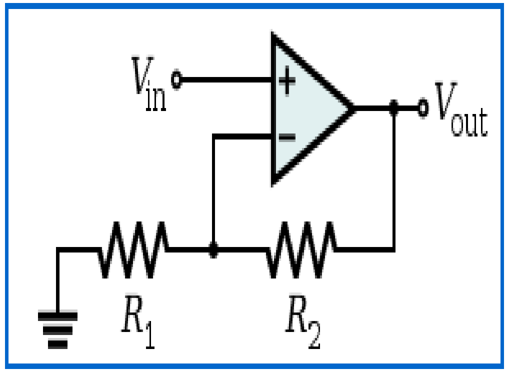
USING OP-AMP FOR SIGNAL MODIFICATION

Inverting amplifier



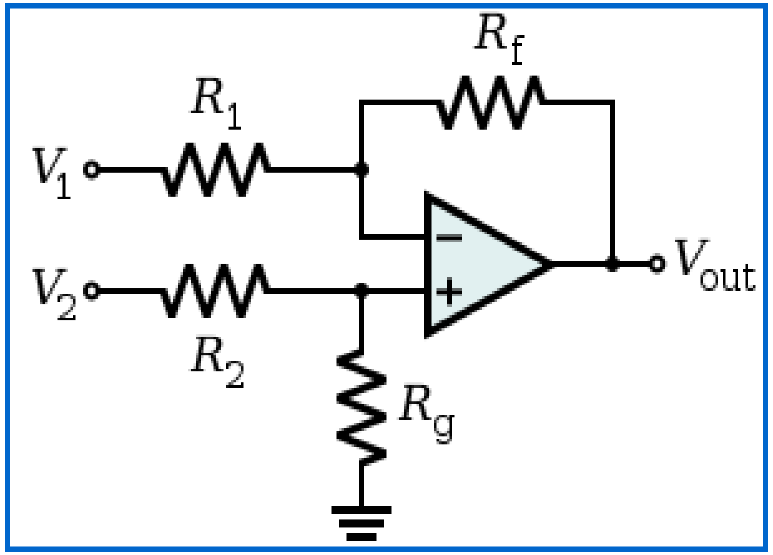
$$V_{out} = -\frac{R_f}{R_{in}} V_{in}$$

Non-inverting amplifier



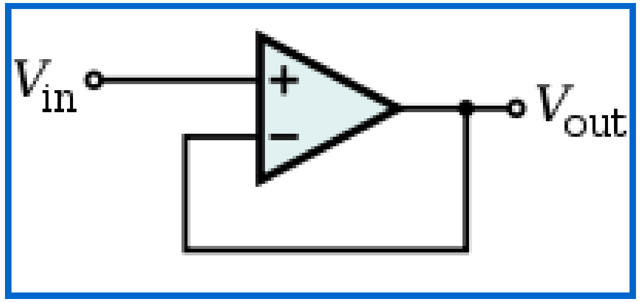
$$V_{out} = V_{in}\left(1 + \frac{R_2}{R_1}\right)$$

Differential amplifier



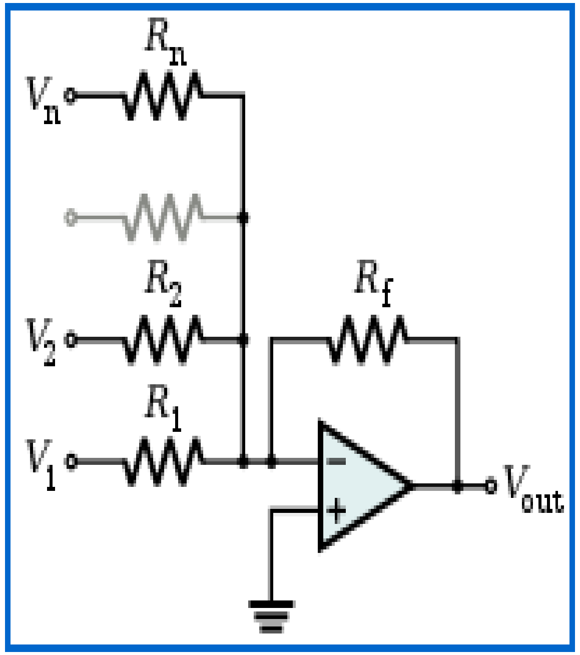
$$V_{out} = \frac{(R_f + R_1)\,R_g}{(R_g + R_2)\,R_1}V_2 - \frac{R_f}{R_1}V_1$$

Voltage follower (Unity Buffer Amplifier)



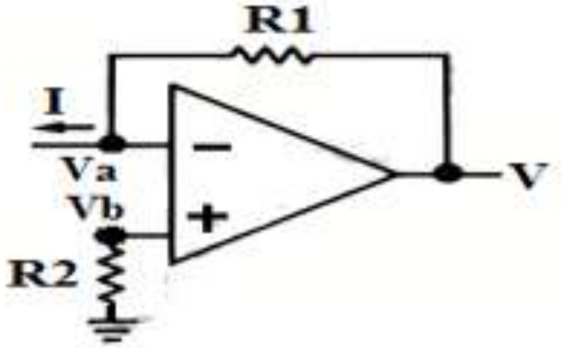
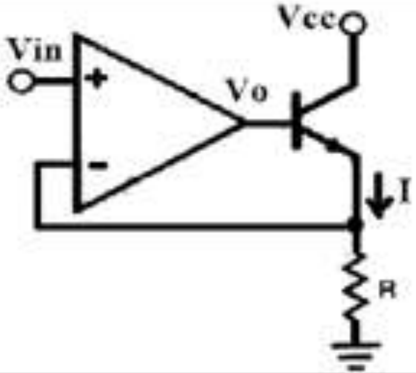
$$V_{out} = V_{in}$$

Summing amplifier



$$V_{out} = -R_f\left(\frac{V_1}{R_1} + \frac{V_2}{R_2} + \cdots + \frac{V_n}{R_n}\right)$$

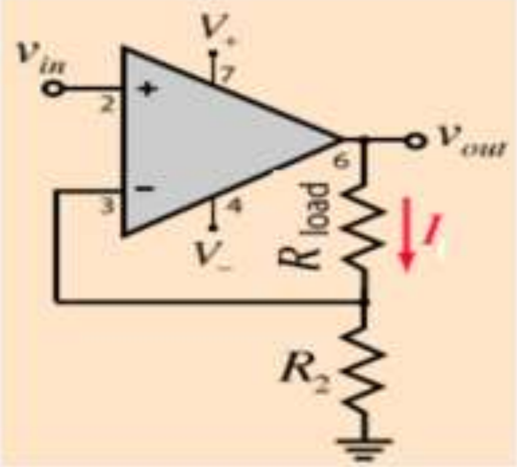
V to I and I to V ccts.

1- Derive the equation relating the o/p to the input in the I to V converter below.	2- Derive the equation relating I to Vin in the V to I converter below.
	

3- A- What is the main purpose of the cct. shown in Fig. below?

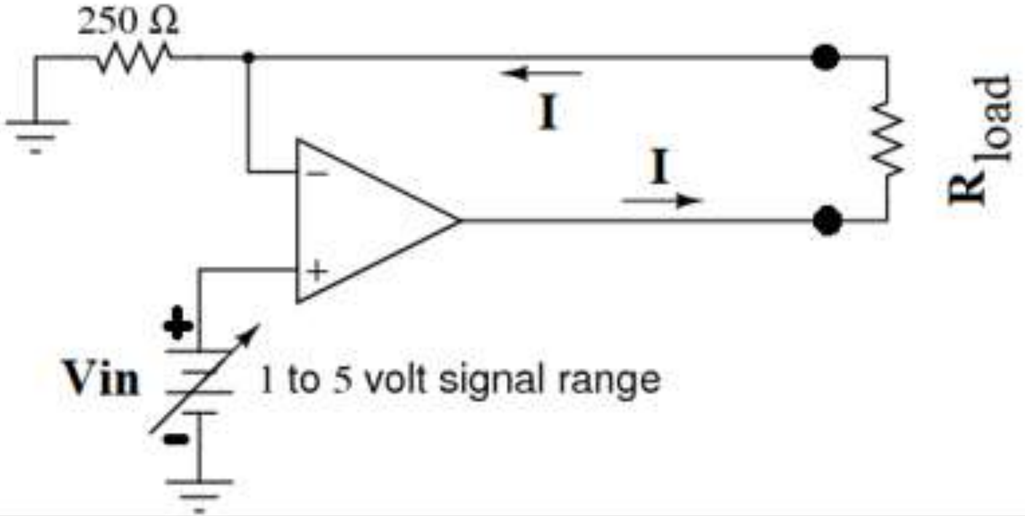
B- Give an equation relating I to Vin.

C- Give an equation relating I to Vout.



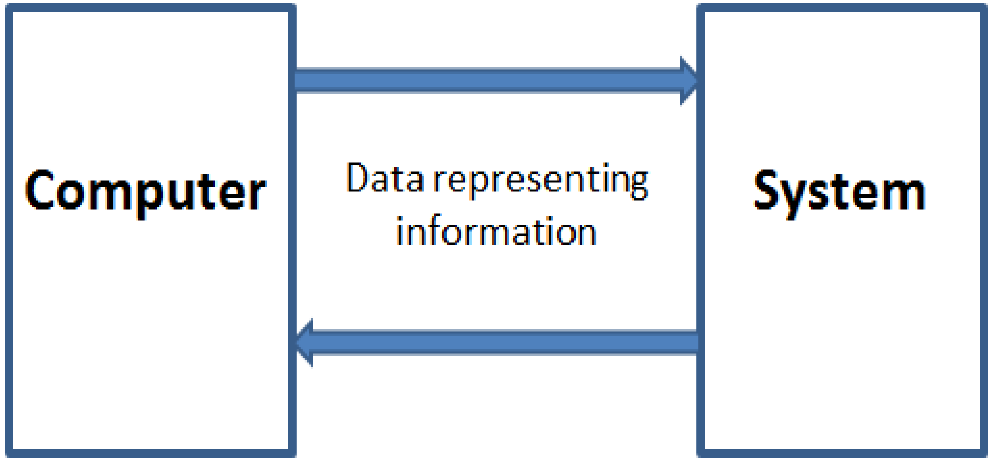
4- A- What is the main purpose of the cct. shown in Fig. below?

B- For the range of Vin shown, what will be the range of I?



COMMUNICATING WITH THE COMPUTER

When using the computer for monitoring and control of a given system, information will pass in both directions as shown. This requires the existence of IN and OUT ports in the computer, which is actually the case. There are many ways of communicating with the compute which come in two main categories, buses and ports. ISA bus, PCI bus, Parallel port, and Serial port are examples of such ways. Of course all these ways use connectors or slots for the actual physical connections. Some of these ways are universal and some are dedicated. Data are sent or received either in parallel or serial way.



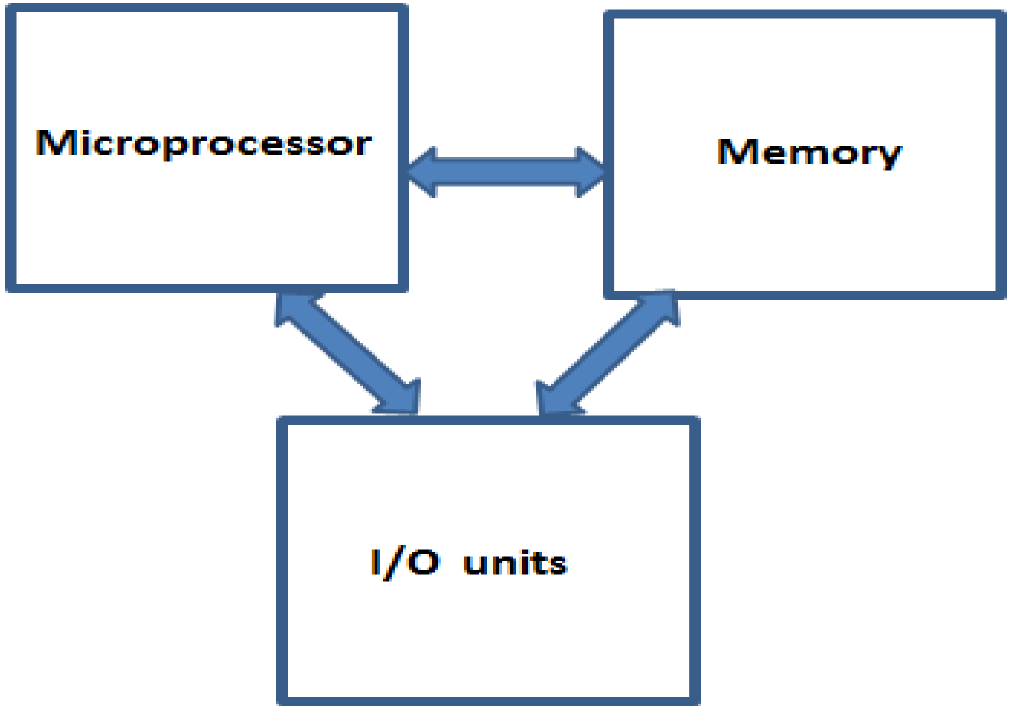
Computer

Data representing information

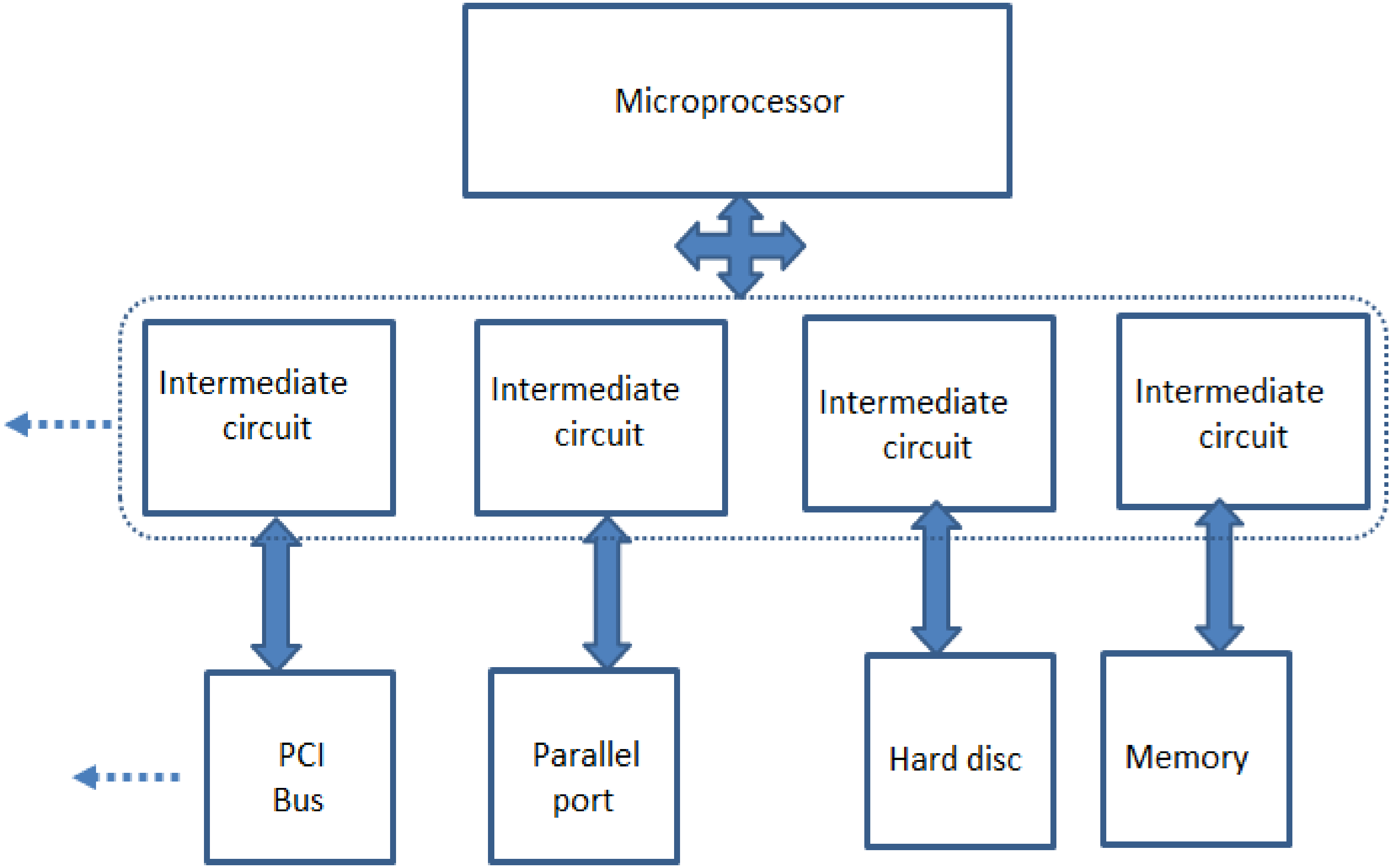
System

Computer architecture:

Generally the microcomputer consists of the microprocessor, memory, and input/output units.

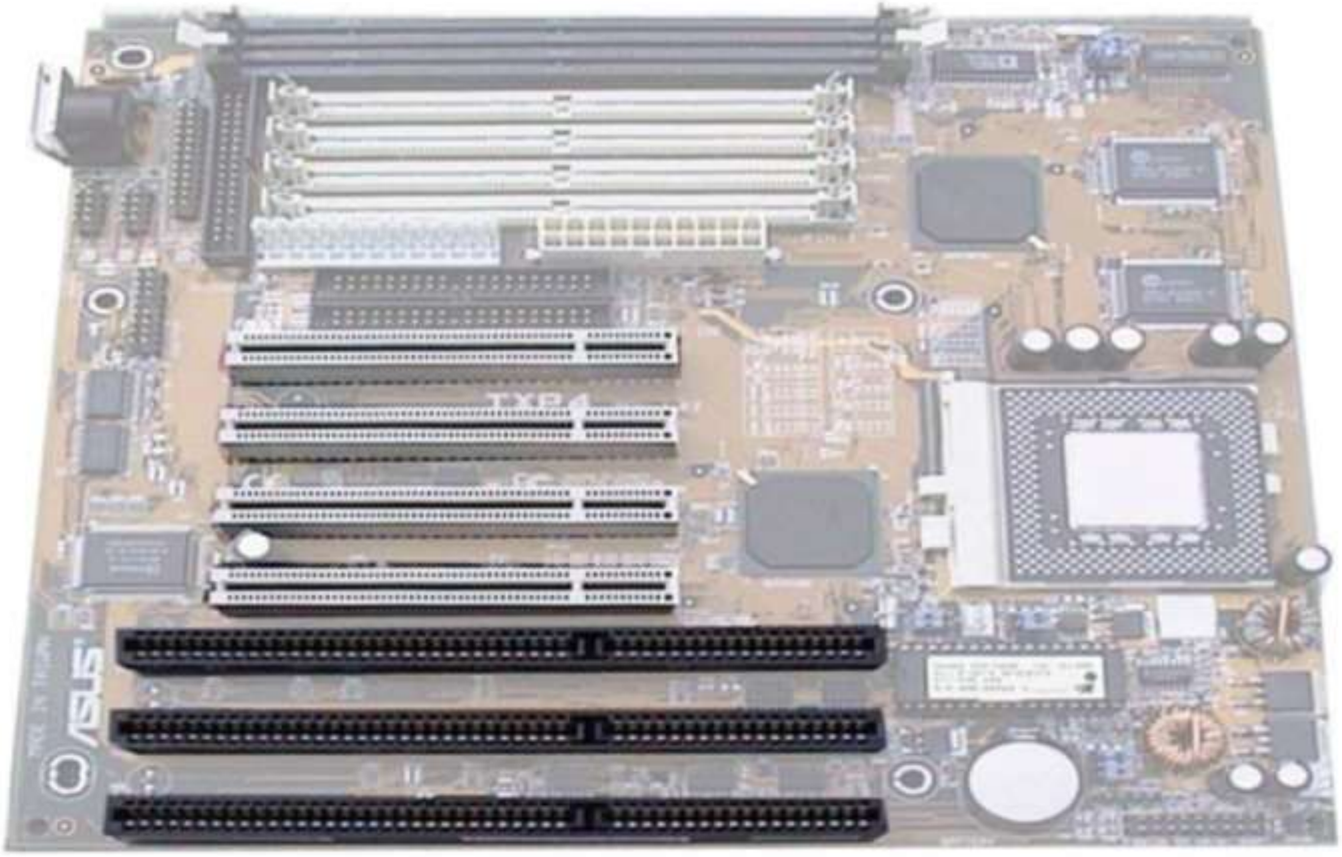


It is important to notice that there are many different computer architectures. Figure below shows just a simple partial clarifying block diagram.



The basic building unit of the μcomputer is the μprocessor. The μprocessor buses are used by intermediate circuits (to be designed) to communicate with the different computer units.

Figure below shows a sample motherboard with slots and connectors among which are the ISA and PCI.



Bus and I/O Standards:

The microcomputers buses are many, among which are the followings:

[[PCI Express (formerly 3GIO)]] [[InfiniBand (formerly System I/O, NGIO, Future I/O)]] [[HyperTransport (formerly LDT)]] [[RapidIO]] [[PCI Bus, PCI-X Bus]] [[ISA Bus (Industry Standard Architecture)]]

[[Plug and Play (PnP)]] [[EISA Bus (Extended Industry Standard Architecture)]][[

VL Bus (VESA Local Bus/Video Electronics Standards Association)

PCMCIA (Personal Computer Memory Card International Association)

[[CardBus (PCMCIA Bus Master)]] [[Micro Channel]]

[[[AGP (Accelerated Graphics Port)]] [[I2O, Intelligent-IO]]

[[USB (Universal Serial Bus)]] [[SMBus (System Management Bus), I^{2}C, ACCESS.bus]] [[[IrDA (infrared data link)]]

[[IDE, EIDE, ATA, ATA-2, ATAPI, Fast ATA, Ultra ATA, Ultra DMA, DMA/33, DMA/66, etc.]] [[SCSI (Small Computer Systems Interface)]]

[[RAID (Redundant Array of Inexpensive Disks)]] [[Fibre Channel]]

[[FireWire (IEEE 1394)]] [[HIPPI]] [[HSSI (High Speed Serial Interface)]] [[PC/104]] [[Multibus]] [[VME Bus]] [[[STD 32 Bus]]

[[CAN Bus (Controller Area Network)]]

Lecture 3 : ISA Bus (ISA = Industry Standard Architecture)

Description for some of the ISA Bus Signal:

SA19 to SA0 *System Address* bits 19:0 are used to address memory and I/O devices within the system.

AEN *Address Enable* is used to degate the system microprocessor and other devices from the bus during DMA transfers. When this signal is active the system DMA controller has control of the address, data, and read/write signals. This signal should be included as part of ISA board select decodes to prevent incorrect board selects during DMA cycles.

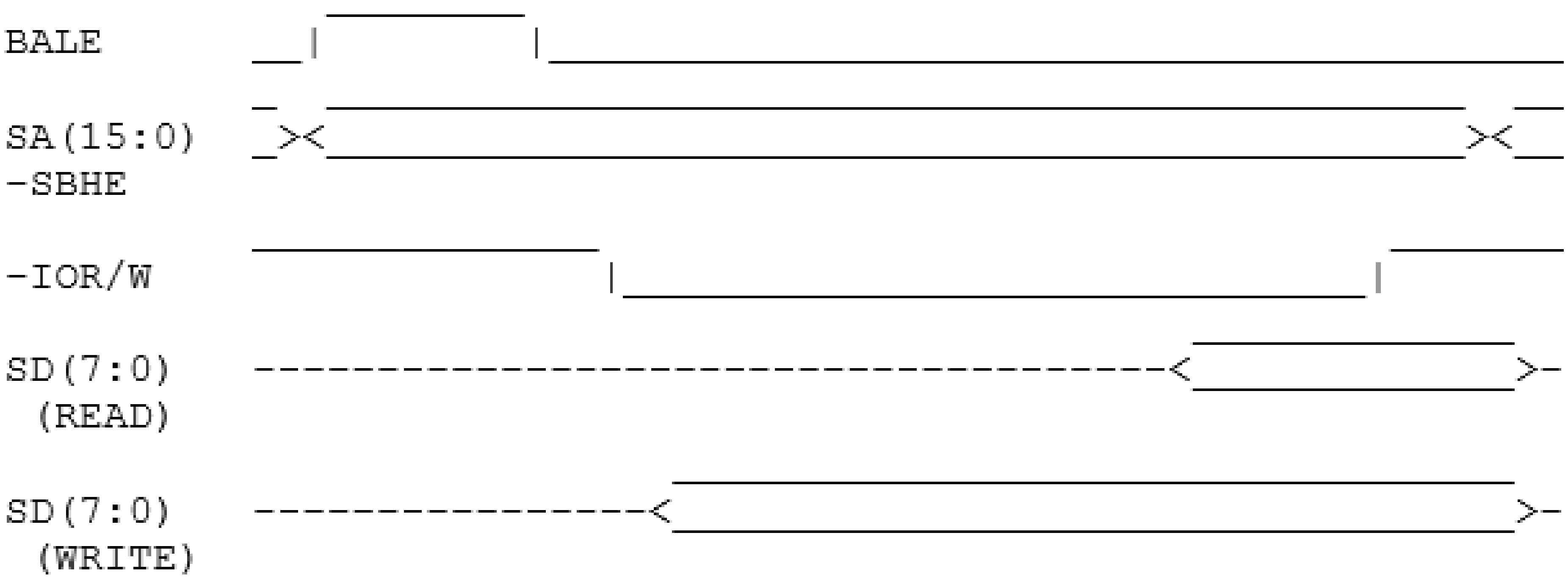
SD7 to SD0 *System Data* serves as the data bus bits for devices on the ISA bus. SD7 is the most significant bit. SD0 is the least significant bits. SD7 to SD0 are used for transfer of data with 8-bit devices.

-IOR *I/O Read* is driven by the owner of the bus and instructs the selected I/O device to drive read data onto the data bus.

-I**OW** *I/O Write* is driven by the owner of the bus and instructs the selected I/O device to capture the write data on the data bus.

ISA Bus Timing Diagrams

8-Bit I/O Bus Cycles



Note: For more details, refer to the data sheet given to you or any related reference.

Some of the ICs that will be used in designs

A/D : ADC0808 , ADC0804 , ((AD574))

D/A : TLC7226

PPI : 8255

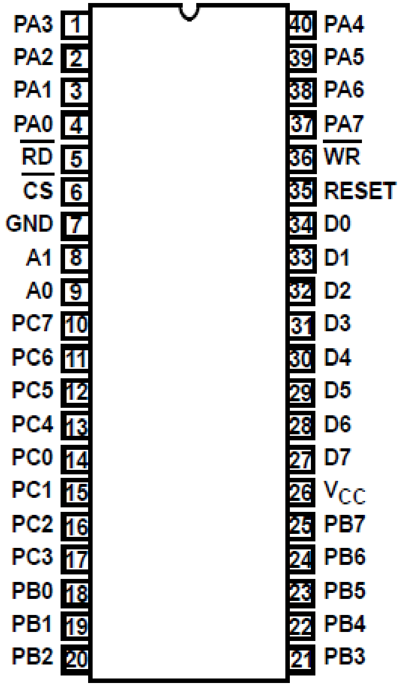
Latch/buffer: 74374

ICs:

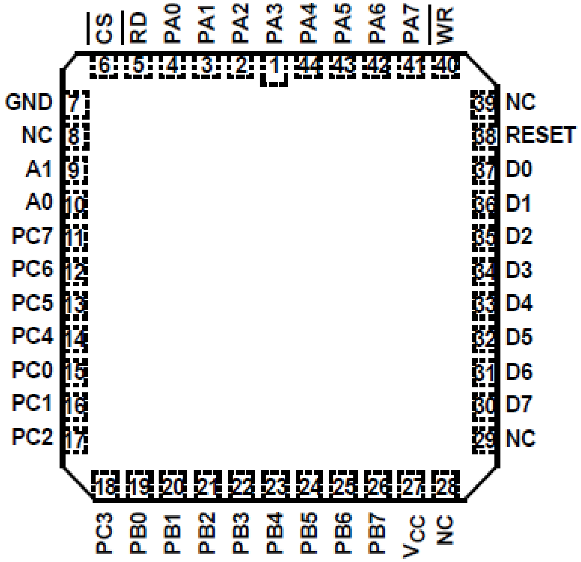
IC1: 8255 PPI (L-notes-ICs-1)(16-10-2017)

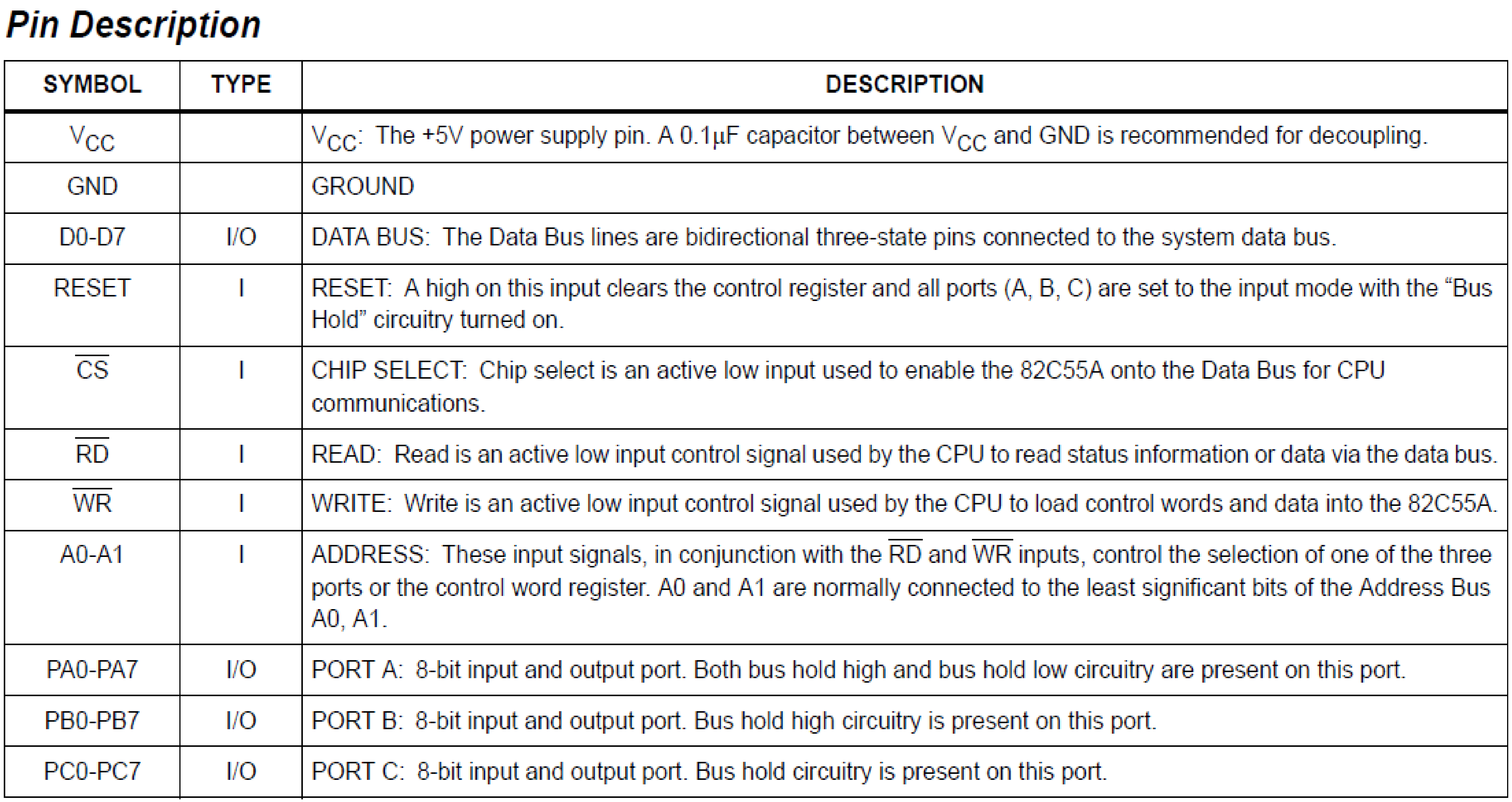
PPI 8255 pin configuration (Pinout)

82C55A (PDIP, CERDIP)
TOP VIEW



82C55A (CLCC)
TOP VIEW

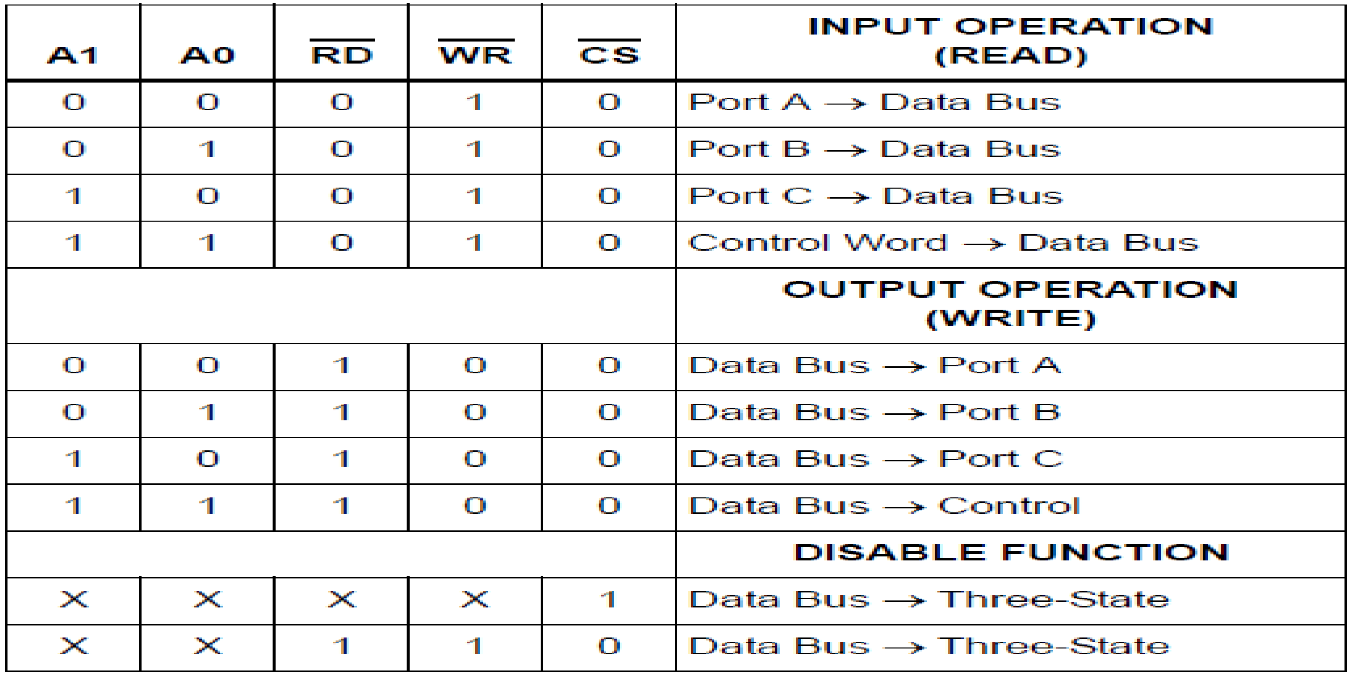




Pin Description

SYMBOL	TYPE	DESCRIPTION
V_{CC}		V_{CC}: The +5V power supply pin. A 0.1μF capacitor between V_{CC} and GND is recommended for decoupling.
GND		GROUND
D0-D7	I/O	DATA BUS: The Data Bus lines are bidirectional three-state pins connected to the system data bus.
RESET	I	RESET: A high on this input clears the control register and all ports (A, B, C) are set to the input mode with the “Bus Hold” circuitry turned on.
$\overline{CS}$	I	CHIP SELECT: Chip select is an active low input used to enable the 82C55A onto the Data Bus for CPU communications.
$\overline{RD}$	I	READ: Read is an active low input control signal used by the CPU to read status information or data via the data bus.
$\overline{WR}$	I	WRITE: Write is an active low input control signal used by the CPU to load control words and data into the 82C55A.
A0-A1	I	ADDRESS: These input signals, in conjunction with the $\overline{RD}$ and $\overline{WR}$ inputs, control the selection of one of the three ports or the control word register. A0 and A1 are normally connected to the least significant bits of the Address Bus A0, A1.
PA0-PA7	I/O	PORT A: 8-bit input and output port. Both bus hold high and bus hold low circuitry are present on this port.
PB0-PB7	I/O	PORT B: 8-bit input and output port. Bus hold high circuitry is present on this port.
PC0-PC7	I/O	PORT C: 8-bit input and output port. Bus hold circuitry is present on this port.

8255 Basic operation



A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	**INPUT OPERATION (READ)**
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					DISABLE FUNCTION
X	X	X	X	1	Data Bus → Three-State
X	X	1	1	0	Data Bus → Three-State

Operational Description

Mode Selection

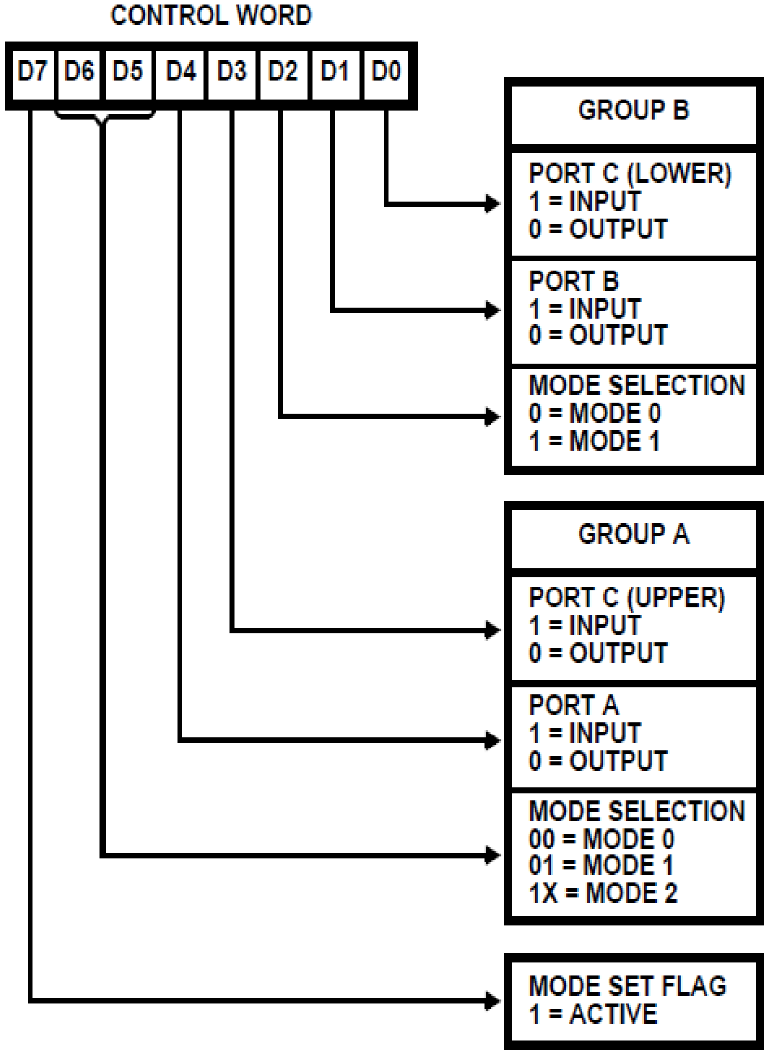
There are three basic modes of operation than can be selected by the system software:

Mode 0 - Basic Input/Output

Mode 1 - Strobed Input/Output

Mode 2 - Bidirectional Bus

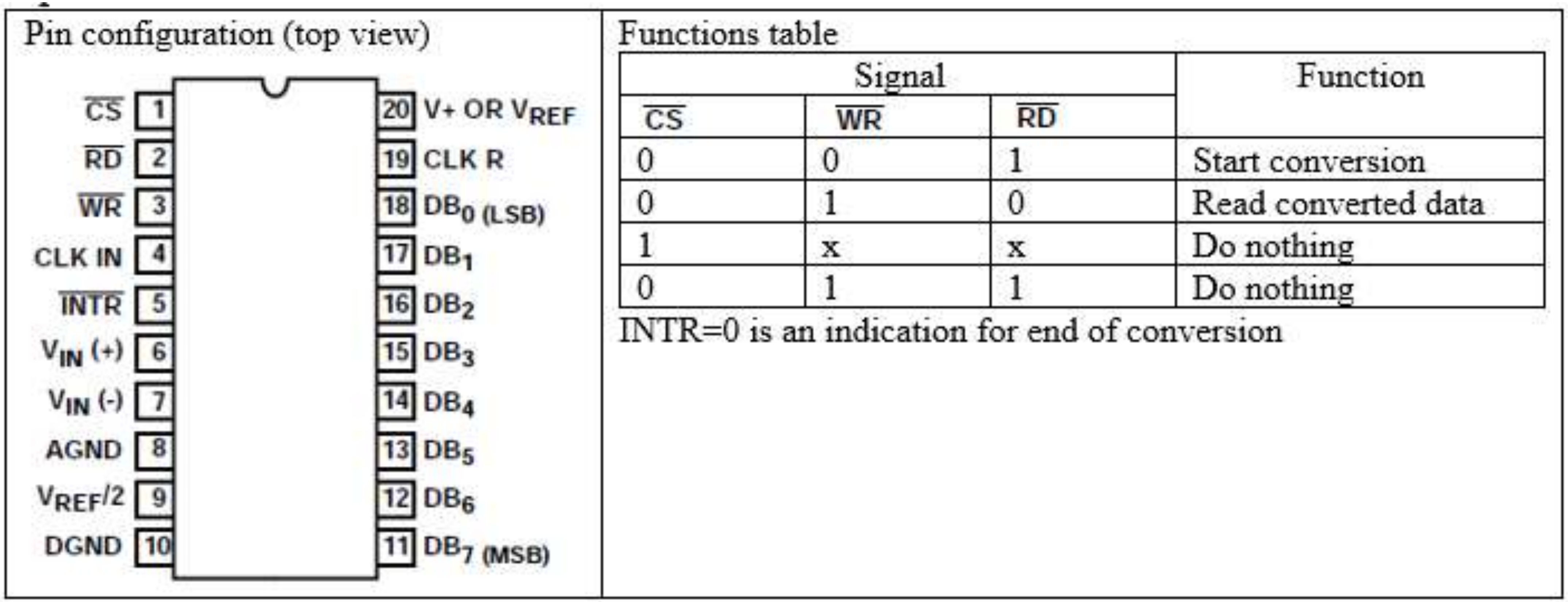
When the reset input goes "high", all ports will be set to the input mode with all 24 port lines held at a logic "one" level by internal bus hold devices. After the reset is removed, the 82C55A can remain in the input mode with no additional initialization required. This eliminates the need to pull-up or pull-down resistors in all-CMOS designs. The control word register will contain 9Bh. During the execution of the system program, any of the other modes may be selected using a single output instruction. This allows a single 82C55A to service a variety of peripheral devices with a simple software maintenance routine. Any port programmed as an output port is initialized to all zeros when the control word is written.



Note: For more details, refer to the data sheet given to you or any related reference.

Lecture 4 : ICs: IC2: ADC0804

ADC0804 is an 8-bit successive approximation analog to digital converter with single analog input.



Pin configuration (top view)

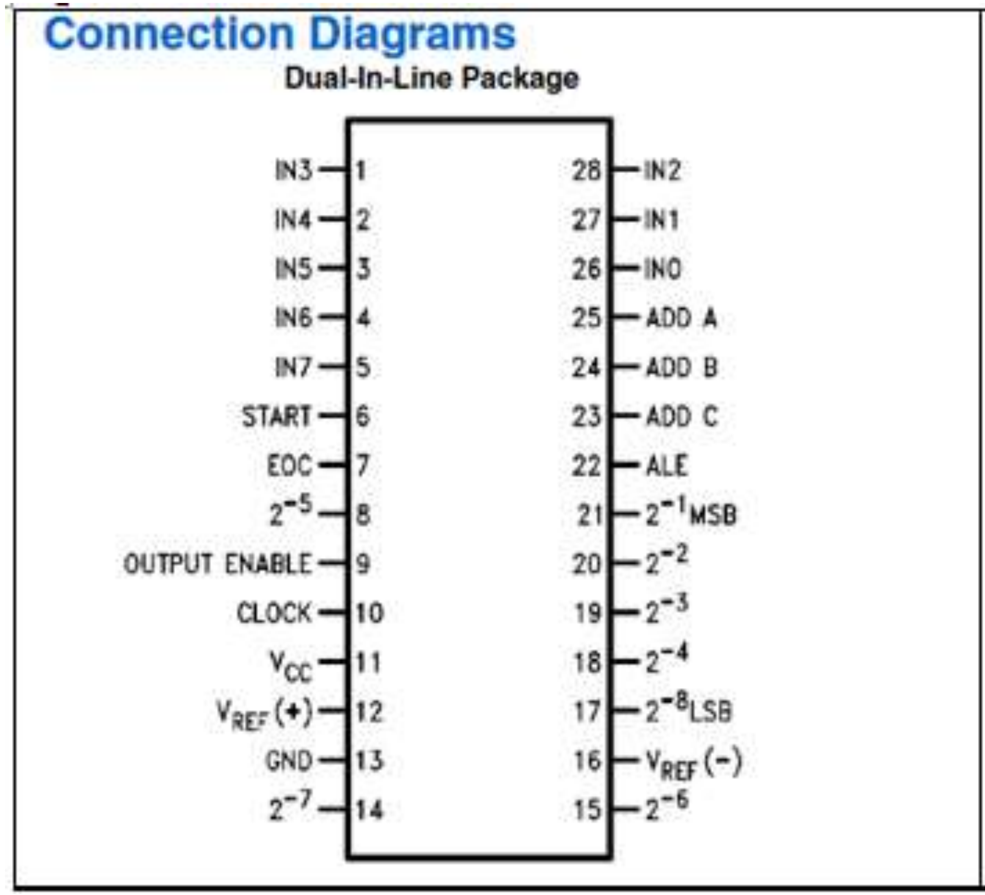
Functions table

Signal			Function
$\overline{CS}$	$\overline{WR}$	$\overline{RD}$	
0	0	1	Start conversion
0	1	0	Read converted data
1	x	x	Do nothing
0	1	1	Do nothing

INTR=0 is an indication for end of conversion

ICs: IC3: ADC0808

ADC0808 is an 8-bit successive approximation analog to digital converter with 8 analog inputs.



Connection Diagrams

Dual-In-Line Package

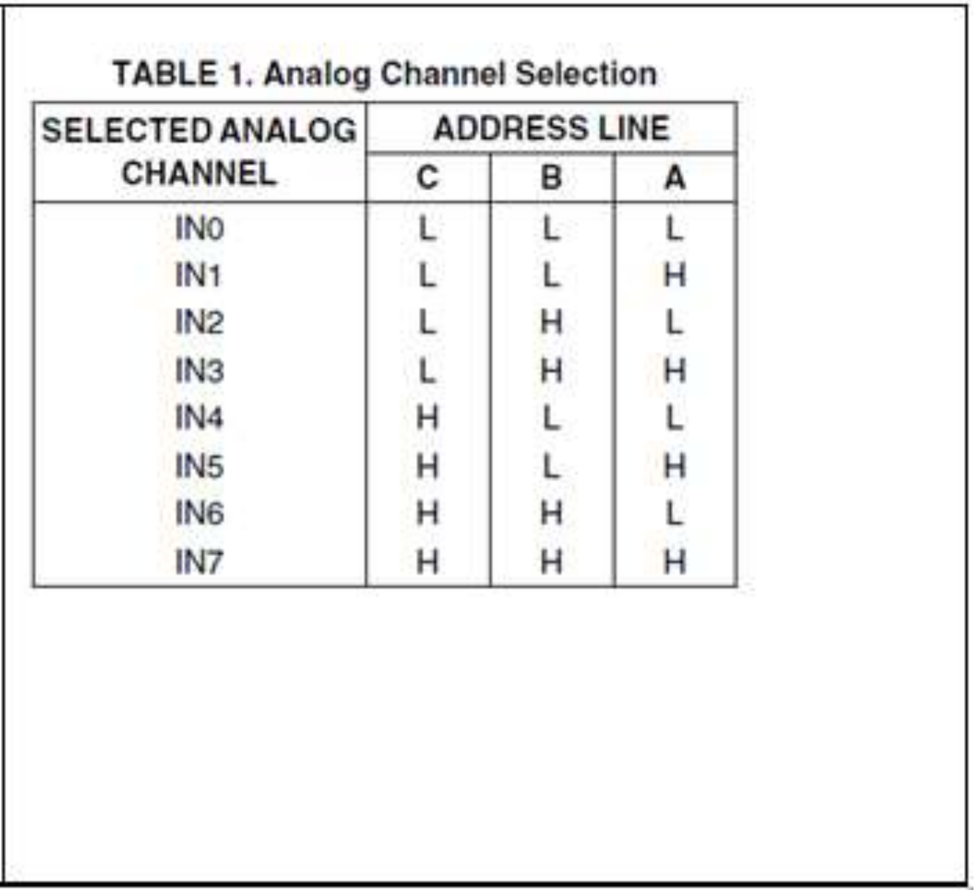
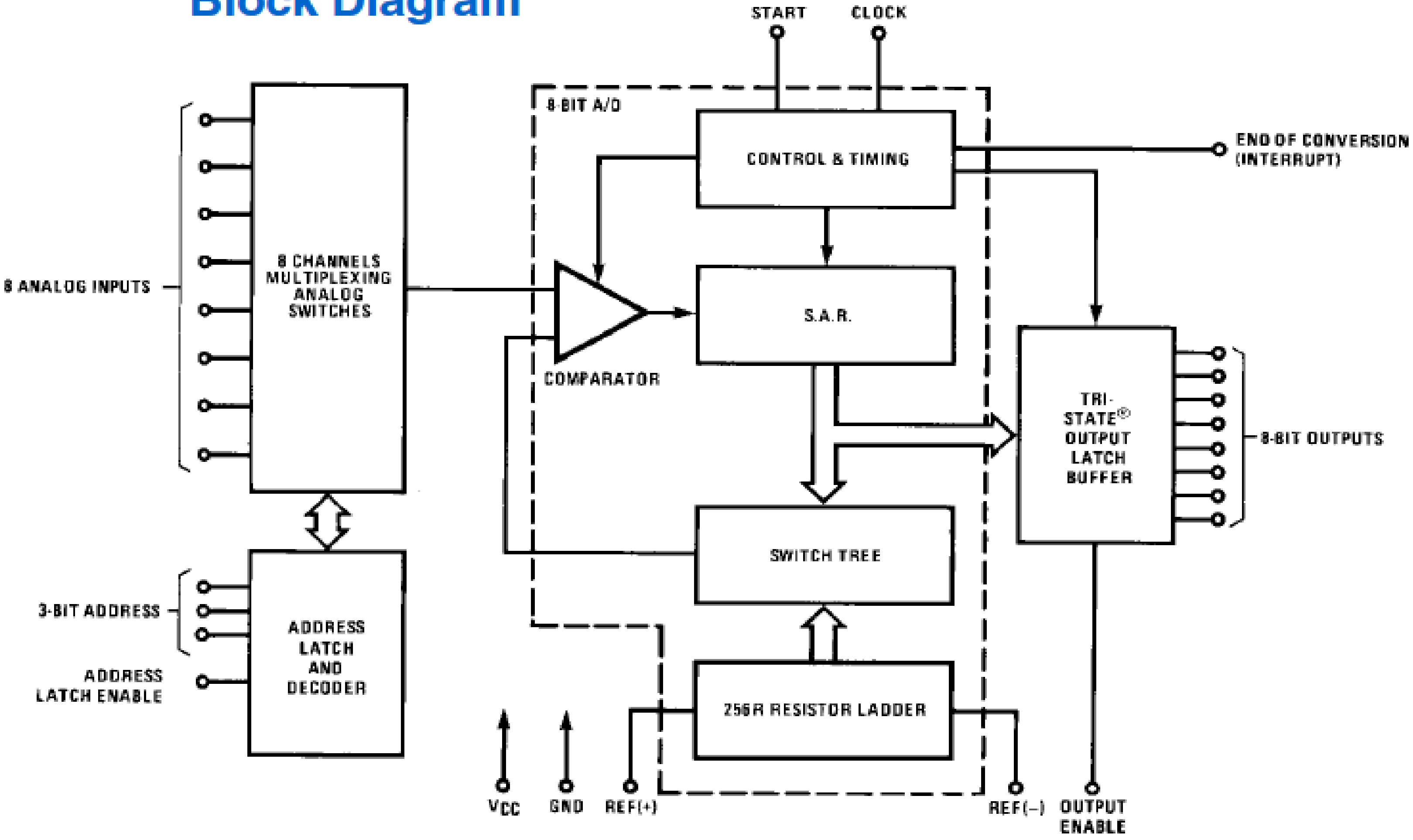


TABLE 1. Analog Channel Selection

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

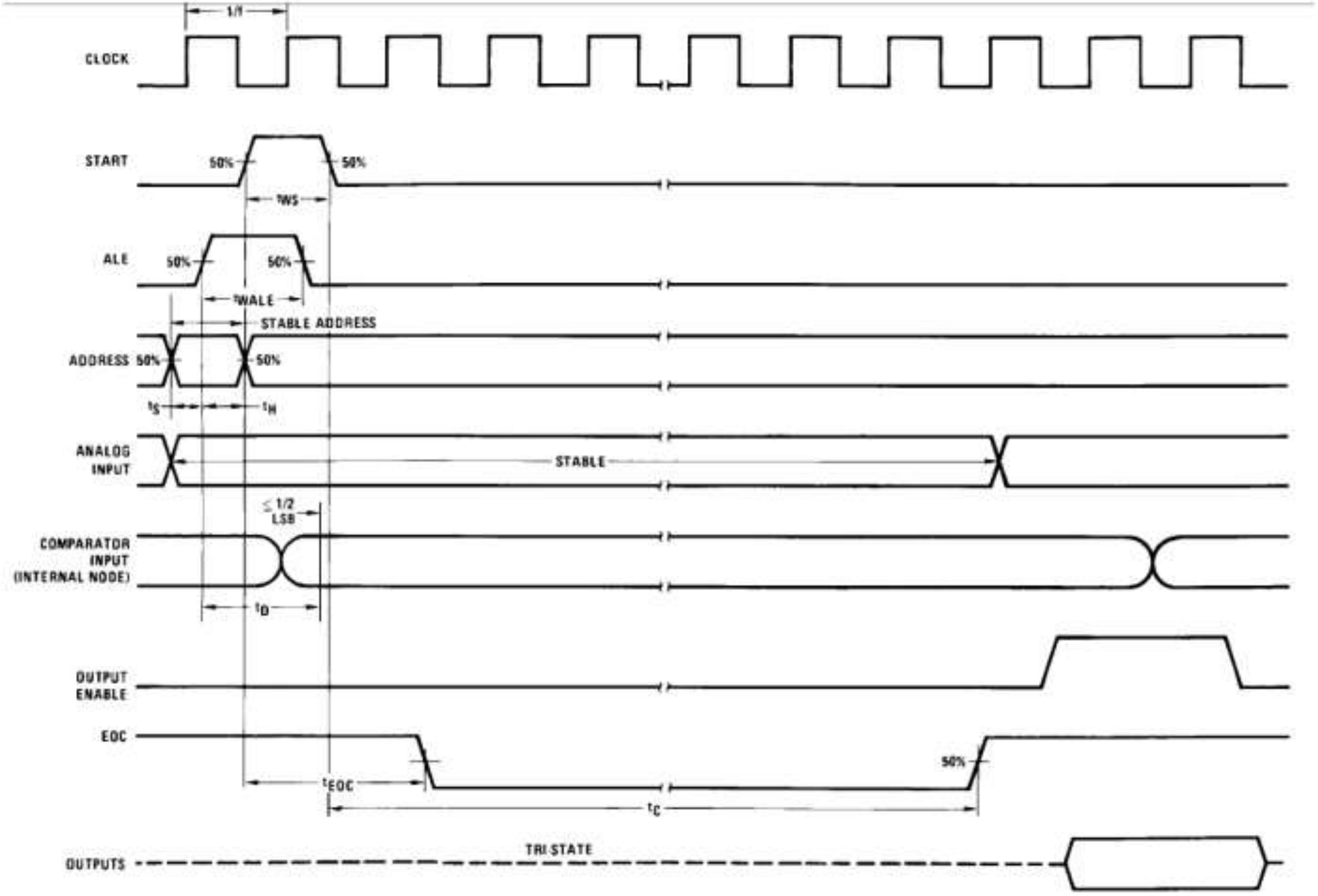
(ALE) and (SC) and (OE) are effective high. EOC = 1 indicates end of conversion.

Block Diagram



See Ordering Information

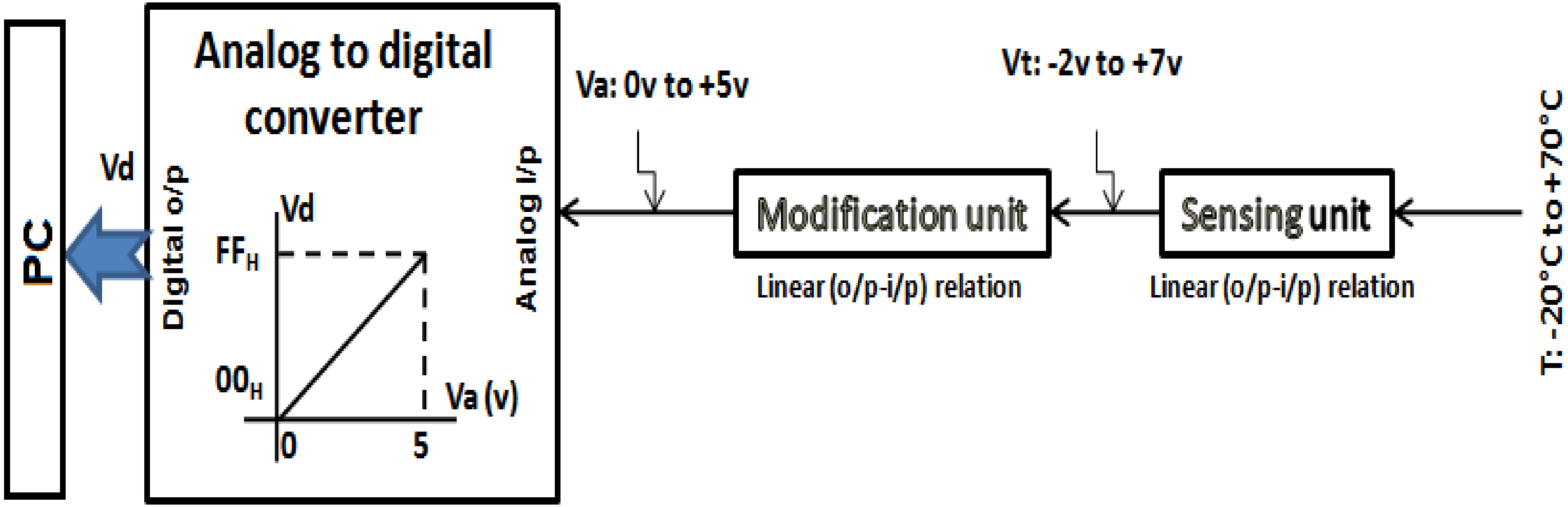
Timing diagram



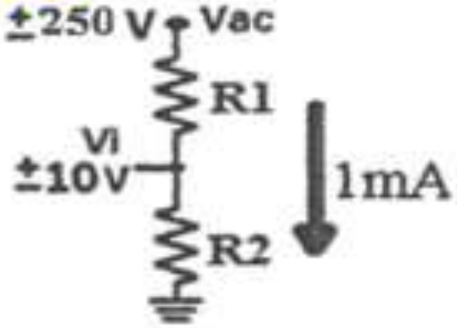
QUESTIONS – Group 1

Q1- Consider the figure below:

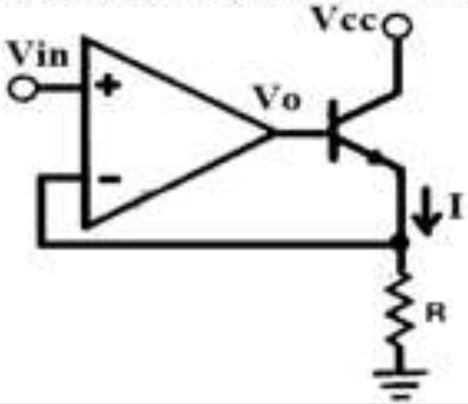
- In Figure below, if V_d is 80_H then what is the value of T in °C?
- Design the modification unit using operational amplifiers.



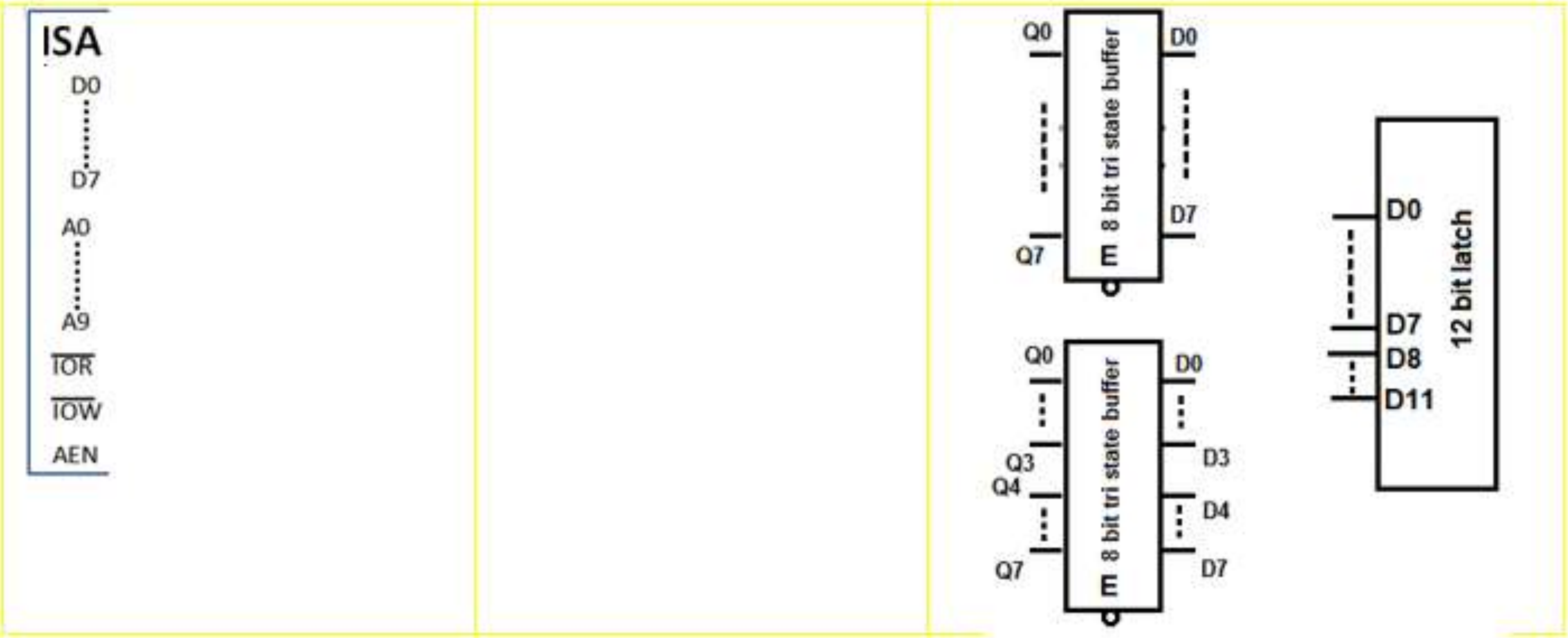
Q2- Find R1 and R2 in circuit below.



Q3- Derive the equation relating I to Vin in the V to I converter below.

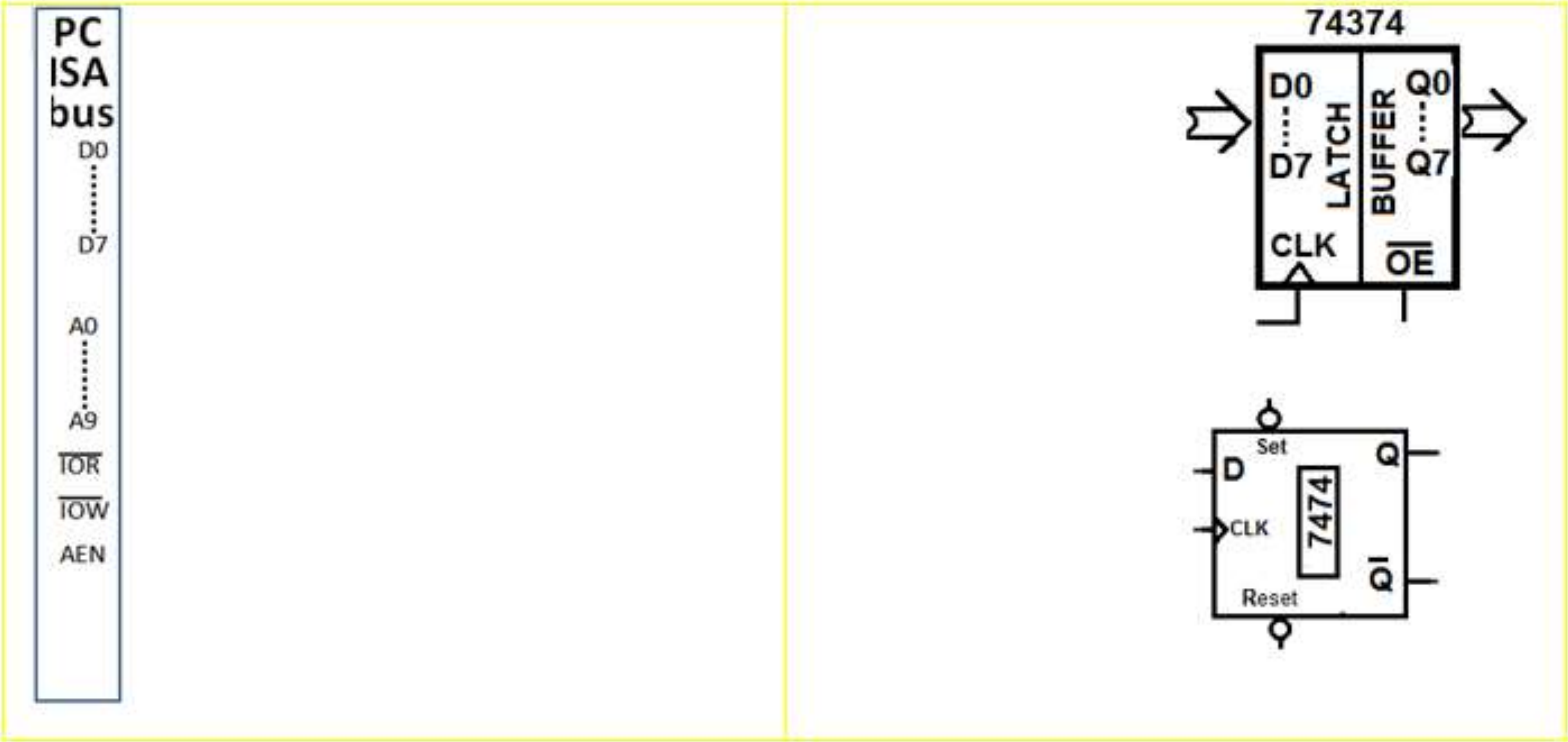


Q-4- In Figure below, it is intended to input the 12 bits output of the 12 bit latch to the PC using the 8bit ISA bus with the help of buffers. Complete the design given below, and write a program to input one data group from the latch to the computer and save it in memory location “TWLV”. (Note: use addresses between 300H and 30FH and as needed).

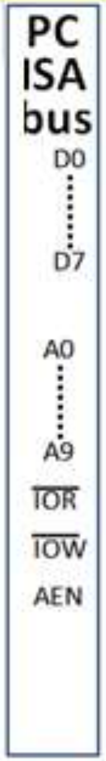

ISA
D0
D7
A0
A9
$\overline{IOR}$
$\overline{IOW}$
AEN
Q0
Q7
8 bit tri state buffer
E
D0
D7
Q0
Q3
Q4
Q7
8 bit tri state buffer
E
D0
D3
D4
D7
D0
D7
D8
D11
12 bit latch

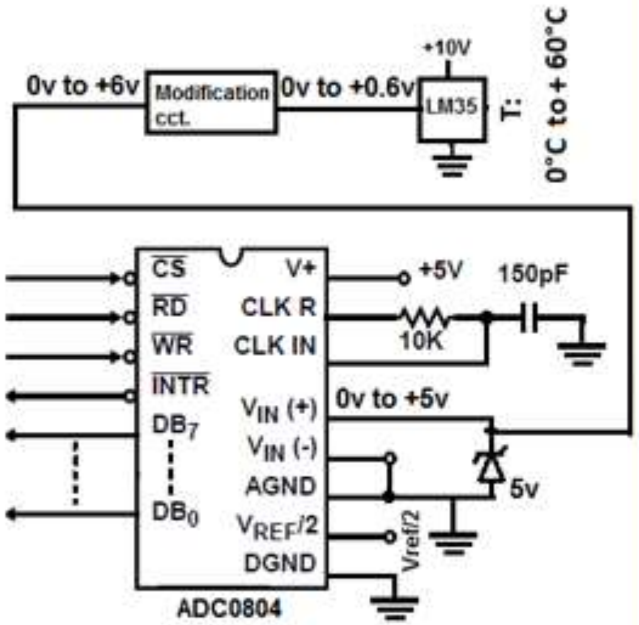
Lecture 5

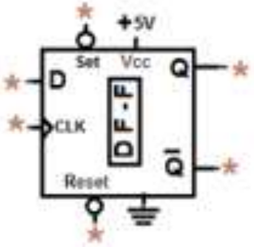
Q5- Interface the 74374 in the figure below to the ISA bus as follows:
-The 74374 is to be interfaced to the ISA bus as an output port with address 378_H .
- A 7474 D F-F is used to enable or disable the 74374 buffer.(Hint: use address 379_H to set the D F-F and address $37A_H$ to reset the D F-F)
- Write a program in pseudo language to output the value EE_H from the PC to the 74374 latch, and to set Q to one.

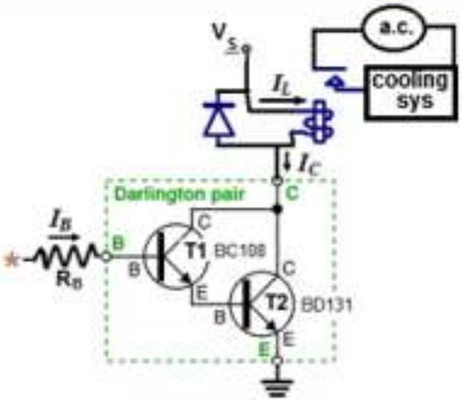

PC ISA bus
D0
D7
A0
A9
$\overline{\text{IOR}}$
$\overline{\text{IOW}}$
AEN
74374
D0
D7
LATCH
BUFFER
Q0
Q7
CLK
$\overline{\text{OE}}$
Set
D
CLK
7474
Q
$\overline{\text{Q}}$
Reset

Q6- The system shown in Fig. below is used to activate a 220v a.c. cooling system if the temperature T increases above 30°C and turn the system off if the temperature decreases below 20°C. Complete the system design. Use addresses between 300_H and $3FF_H$. Then write a loop program which checks temperature and takes the right action.


PC
ISA
bus
D0
D7
A0
A9
$\overline{IOR}$
$\overline{IOW}$
AEN


0v to +6v
Modification cct.
0v to +0.6v
+10V
LM35
T: 0°C to + 60°C
$\overline{CS}$
$\overline{RD}$
$\overline{WR}$
$\overline{INTR}$
DB_7
DB_0
V+
CLK R
CLK IN
V_{IN} (+)
V_{IN} (-)
AGND
$V_{REF}/2$
DGND
+5V
150pF
10K
0v to +5v
5v
Vref/2
ADC0804


*
+5V
Set
Vcc
D
Q
CLK
DF-F
$\overline{Q}$
Reset


V_S
a.c.
cooling sys
I_L
I_C
Darlington pair
C
I_B
R_B
B
T1 BC108
E
T2 BD131
E

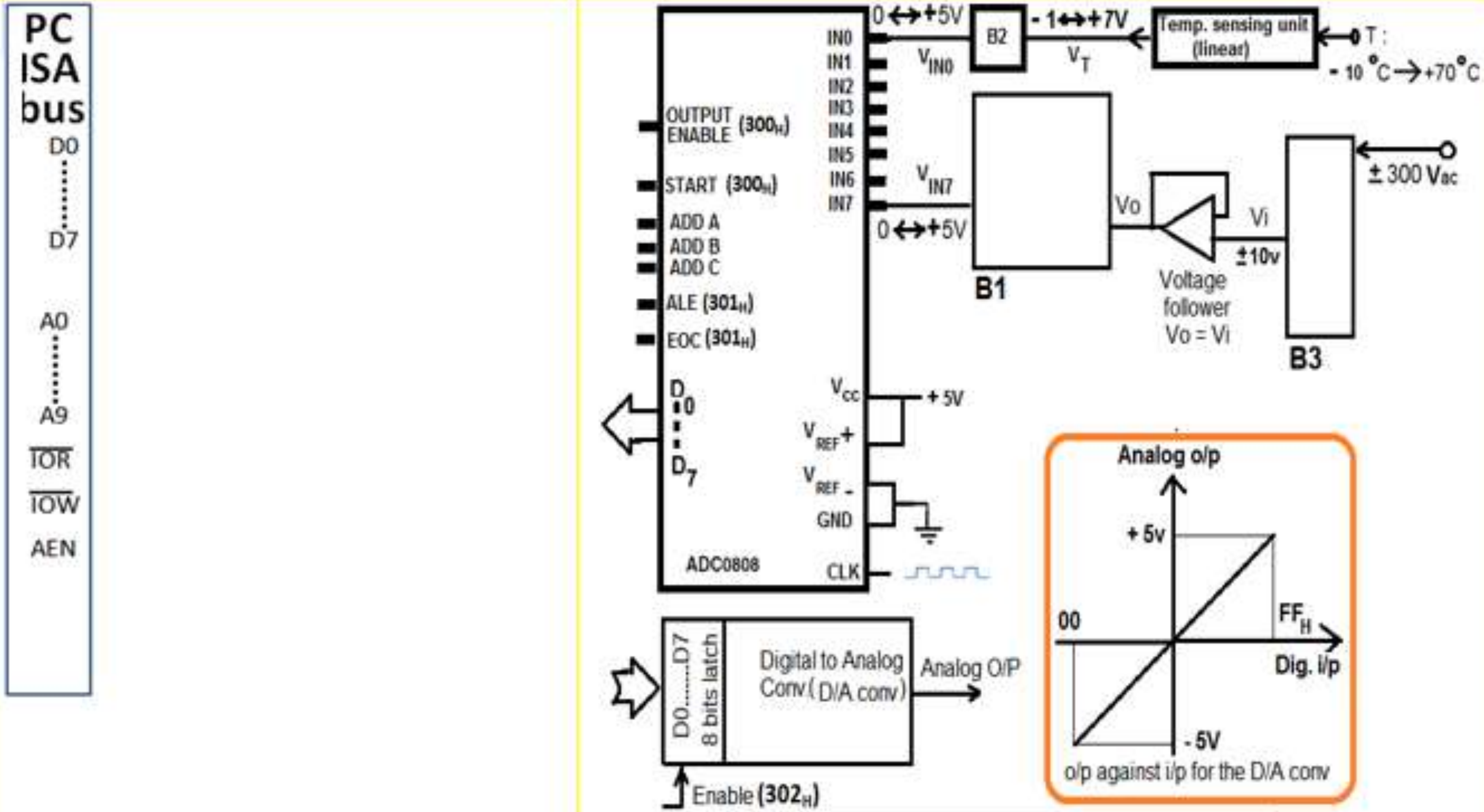
Q7-

A. Interface the cct. in Fig. below to the PC using the ISA bus .

B. Design B1 and B3.

C. Write a program in Pseudo Language which does the following procedure:

```
Inputs a sample of T.
  Compares T (in °C) with $T_d$ (the desired T) (in °C) to find the error $T_e$ (in °C).
  Finds $T_c$ which is equal to $T_e$ multiplied by 2.
  Applies routine ROT1 to find $CTRL_{dig}$ .
  Outputs $CTRL_{dig}$ to the D/A conv.
```

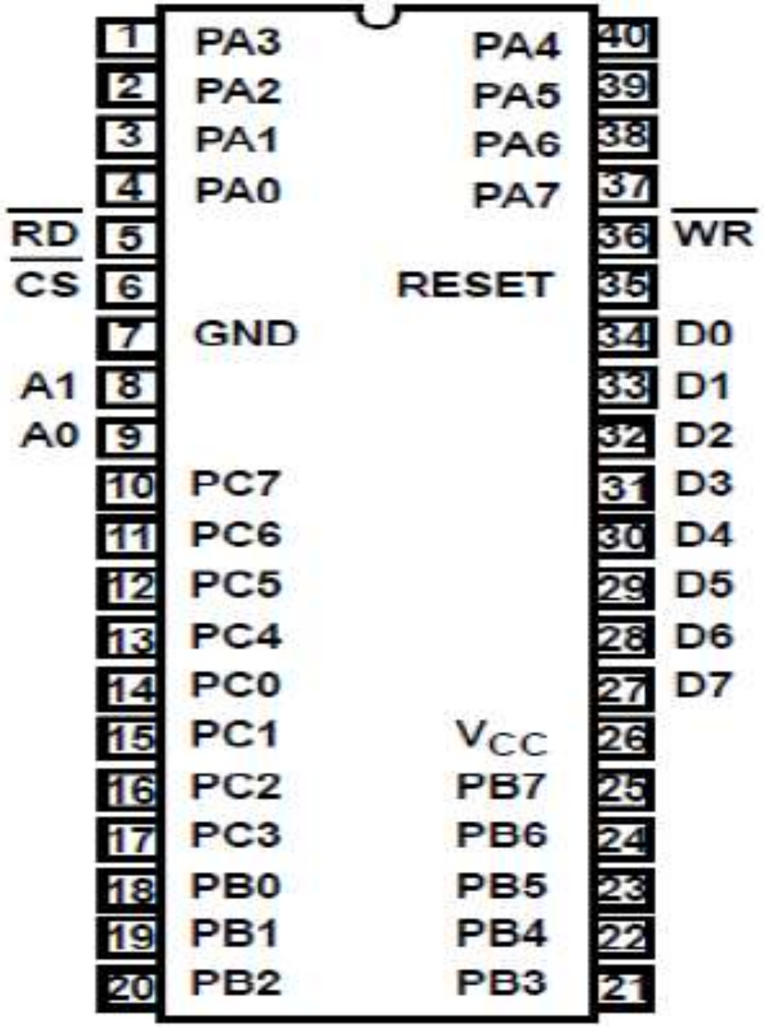


((Note: A/D Convs. normally have high input impedance analog input channels))

Q8- Interface the 8255 PPi to the PC through the ISA bus. Use the addresses 304H – 307H.

And write a program to read data through port A and send it to port B (Considering that the PPI is already programmed in mode 0, and so that: (port A: I/P) (port B: O/P), and (port C: o/p).

82C55A


1 PA3
2 PA2
3 PA1
4 PA0
RD 5
CS 6
7 GND
A1 8
A0 9
10 PC7
11 PC6
12 PC5
13 PC4
14 PC0
15 PC1
16 PC2
17 PC3
18 PB0
19 PB1
20 PB2
PA4 40
PA5 39
PA6 38
PA7 37
36 WR
RESET 35
34 D0
33 D1
32 D2
31 D3
30 D4
29 D5
28 D6
27 D7
V_{CC} 26
PB7 25
PB6 24
PB5 23
PB4 22
PB3 21

- Interface the cct. in Fig. below to the PC using the ISA bus .

A. Design B1 using resistors only.

B. Write a program in Pseudo Language which does the following procedure:

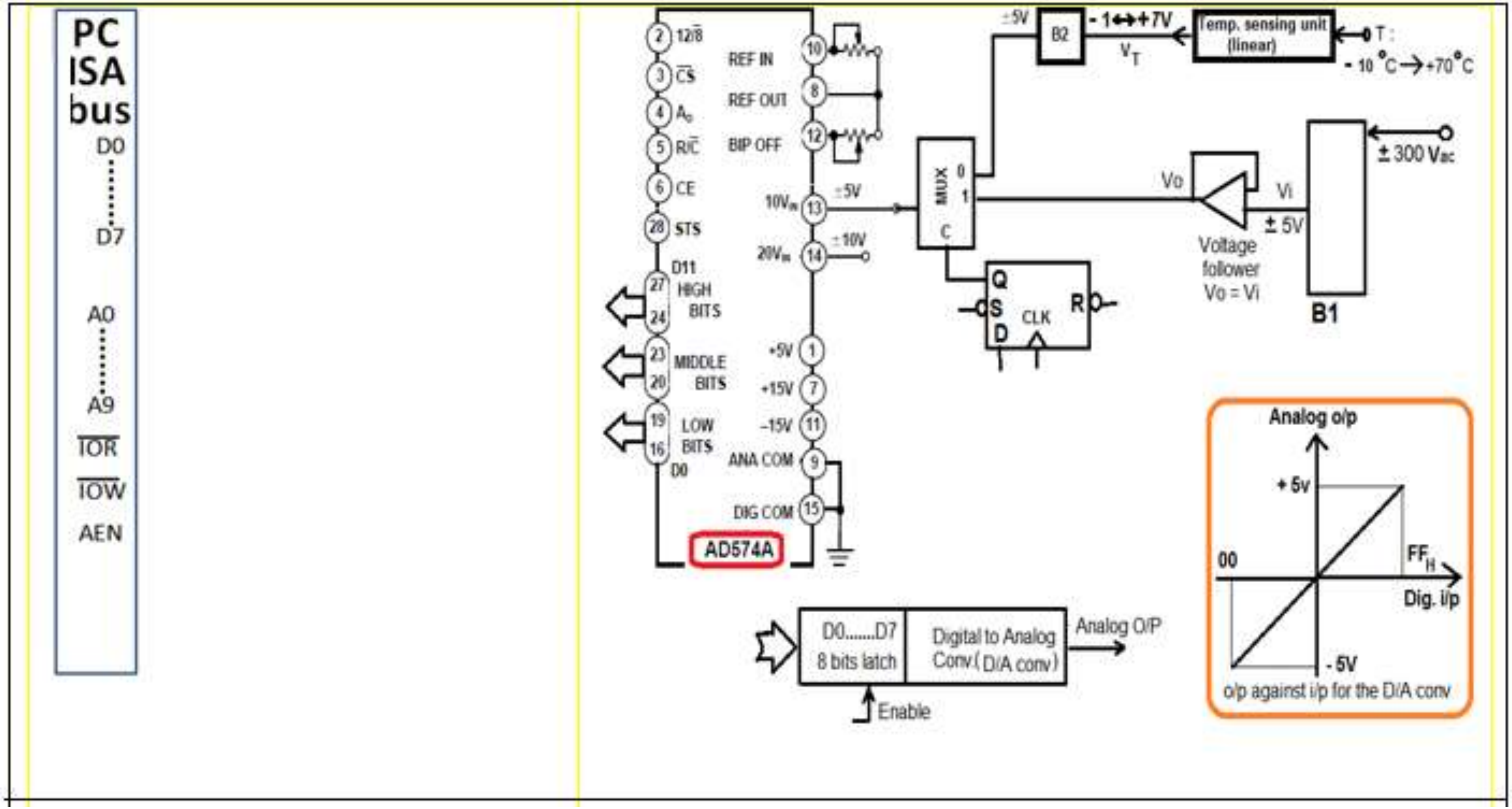
    Inputs a sample of T.

    Compares T (in °C) with T_d (the desired T) (in °C) to find the error T_e (in °C).

    Finds T_c which is equal to T_e multiplied by 2.

    Applies routine ROT1 to find $CTRL_{dig}$.

    Outputs $CTRL_{dig}$ to the D/A conv.


PC
ISA
bus
D0
D7
A0
A9
$\overline{IOR}$
$\overline{IOW}$
AEN
2 12/$\overline{8}$
3 $\overline{CS}$
4 A_0
5 R/$\overline{C}$
6 CE
28 STS
D11
27 HIGH BITS 24
23 MIDDLE BITS 20
19 LOW BITS 16
D0
REF IN 10
REF OUT 8
BIP OFF 12
$10V_{in}$ 13 ±5V
$20V_{in}$ 14 ±10V
+5V 1
+15V 7
-15V 11
ANA COM 9
DIG COM 15
AD574A
MUX
0
1
C
Q
S
D
CLK
R
±5V
B2
- 1↔+7V
V_T
Temp. sensing unit (linear)
T :
- 10 °C→+70°C
± 300 Vac
B1
Vo
Vi
± 5V
Voltage follower
Vo = Vi
Analog o/p
+ 5v
00
FF_H
Dig. i/p
- 5V
o/p against i/p for the D/A conv
D0.......D7
8 bits latch
Digital to Analog Conv.(D/A conv)
Analog O/P
Enable

REFERENCES - Group 1

1-Pc Interfacing, Pei An, 1st ed, Newnes. 1998

2-PC Architecture, Michael Karbo, 1st ed, karbosguide, 2005

3-PC Based Instrumentation and Control, Mike Tooley, 1st ed, Elsevier, 2005

Document 2: Computer Interfacing /4th / CSE-UOT / 2017-2018 (Dr L.J.S.)

****

Control and systems Eng. Dept. – University of Technology

4th year-Computer Interfacing Course

1st term course – 2017/2018

Instructor: Dr Laith J. Saud

****

This document includes:

- Notes about the PC parallel port.

- Questions related to interfacing through the parallel port.

- A case study related to interfacing through the ISA bus.

Lecture 6: THE PARALLEL PORT (The Centronic port)

Introduction

The Centronic port, also known as the printer port or the parallel port, is an industrial standard interface designed for connecting printers to a computer. A computer at least has one such a port installed. The port may come with the computer's mother-boards or with plug-in I/O cards. Adding more Centronic ports is easy and inexpensive. In total, four Centronic ports may be installed on a computer and they have logic names LPTl to LPT4.

This section describes the Centronic port from the point of view that it is used as a general purpose I/O interface. Operations specific to printers are not paid this much attention.

Port connectors

The port connectors on a computer and on a printer are different. The one on the computer is a 25 pin D-type female connector (Figure P.1.1.a), and the latter is a 36-pin female Centronic-type connector (Figure P.1.1.b) . The pin functions of the two connectors are shown in Figure P.1.1. To connect a printer to a computer, a printer cable is used (Figure P1.2). The length of the cable must not exceed 5 meters. The Centronic interface is not for long distance operation.

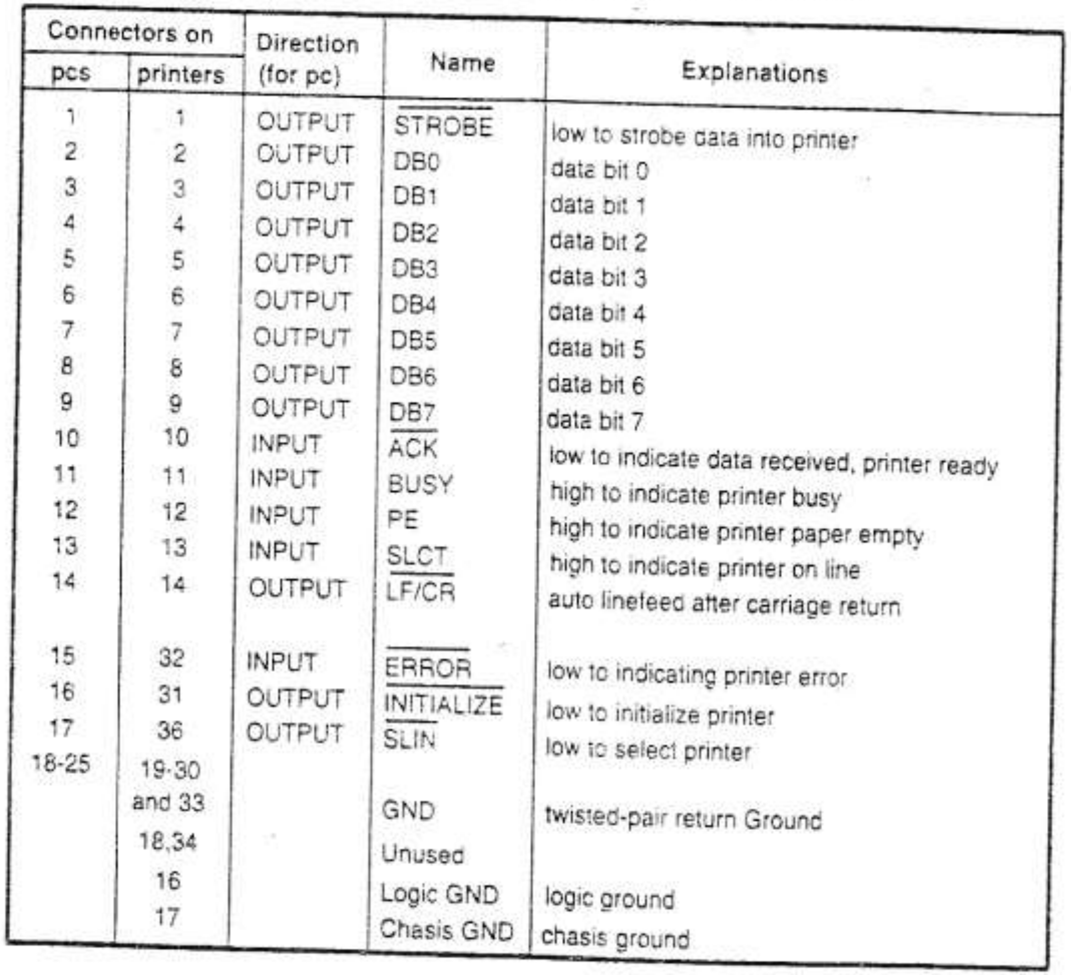
Internal hardware organization

The circuit of a generic Centronic port inside a PC is shown in Figure P1.3, Eight-bit data is latched into IC1 by writing to a port having an address: base address+0. This operation pulls down -WRITE_DATA. The output of the data forms the Data group. Data can be read into the computer from (he same address via IC2 under the control of -READ_DATA. When reading data, the Output from IC1 must be in high impedance state. This is achieved by making pin 1 (OUTPUT ENABLE) of IC1 high. A 6-bit control word is latched to IC3 by writing to base address+2 which pulls down -WRITE_CONTROL. Bit 0 to bit 3 are output to the Port connector to form the Control group. Some of the lines are inverted by open-collector inverters (IC6 and IC7). All the output lines are pulled to +5V by 4k7 resistors. These bits can be read back into the computer at the same address via IC4a under the control of -READ_CONTROL. Bit 4 of the control byte enables the interrupt and bit 5 enables or disables the output of IC1. Five lines in the Port connector (the Status group) can be read into the computer via IC4b under the control of -READ_STATUS. The address associated with this is base address+1. These inputs are pulled to +5V by 4k7 resistors and one of the lines is inverted.

In original IBM PCs, the Output enable of IC1 is tied to ground to permanently enable the Outputs.
This is the uni-directional version of the Centronic Port. From IBM PS/2, the output enable of ICl is connected to bit 5 of the control register IC3 as shown in Figure P1.3 and the port becomes a bidirectional port. It should be pointed out that many Centronic ports that come with plug-in I/O cards are uni-directional Centronic ports. A simple program can be used to detect whether your Centronic port is a uni-directional or a bi-directional one.

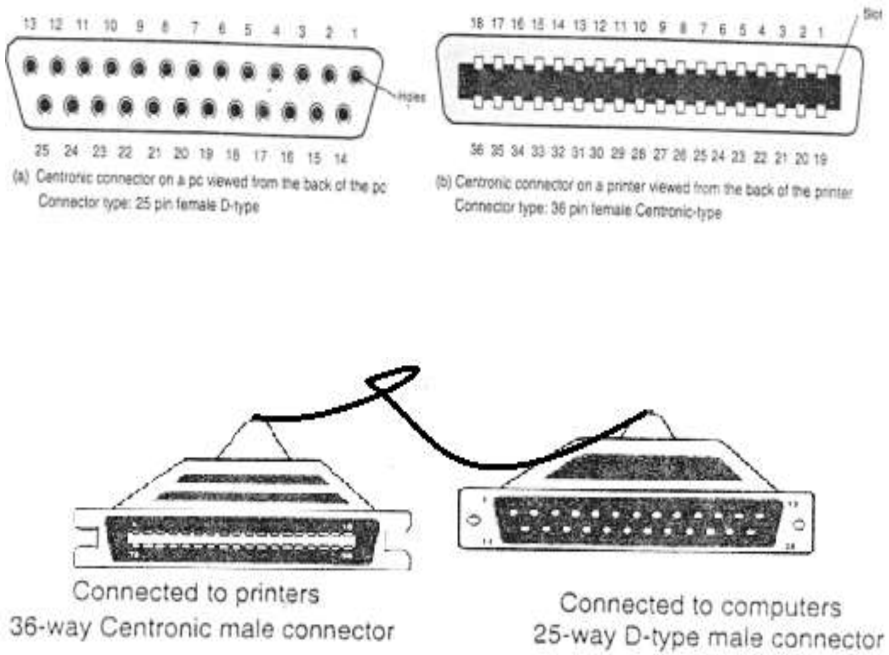
Each output line in the Data group is capable of sourcing 2.6 mA current with the voltage varying between 2.6 to 5V. Each can sink 24 mA. The lines in the Control group have a much smaller capacity to source and to sink current. They can only source 100 μA and sink 8 mA current. For both ports, short circuiting of any two outputs and connecting any lines to the ground or +5V power supply rail are strictly avoided.

Pin functions of the Centronic port connectors



Connectors on		Direction (for pc)	Name	Explanations
pcs	printers			
1	1	OUTPUT	$\overline{\text{STROBE}}$	low to strobe data into printer
2	2	OUTPUT	DB0	data bit 0
3	3	OUTPUT	DB1	data bit 1
4	4	OUTPUT	DB2	data bit 2
5	5	OUTPUT	DB3	data bit 3
6	6	OUTPUT	DB4	data bit 4
7	7	OUTPUT	DB5	data bit 5
8	8	OUTPUT	DB6	data bit 6
9	9	OUTPUT	DB7	data bit 7
10	10	INPUT	$\overline{\text{ACK}}$	low to indicate data received, printer ready
11	11	INPUT	BUSY	high to indicate printer busy
12	12	INPUT	PE	high to indicate printer paper empty
13	13	INPUT	SLCT	high to indicate printer on line
14	14	OUTPUT	$\overline{\text{LF/CR}}$	auto linefeed after carriage return
15	32	INPUT	$\overline{\text{ERROR}}$	low to indicating printer error
16	31	OUTPUT	$\overline{\text{INITIALIZE}}$	low to initialize printer
17	36	OUTPUT	$\overline{\text{SLIN}}$	low to select printer
18-25	19-30 and 33		GND	twisted-pair return Ground
	18,34		Unused	
	16		Logic GND	logic ground
	17		Chasis GND	chasis ground

Figure P1.1



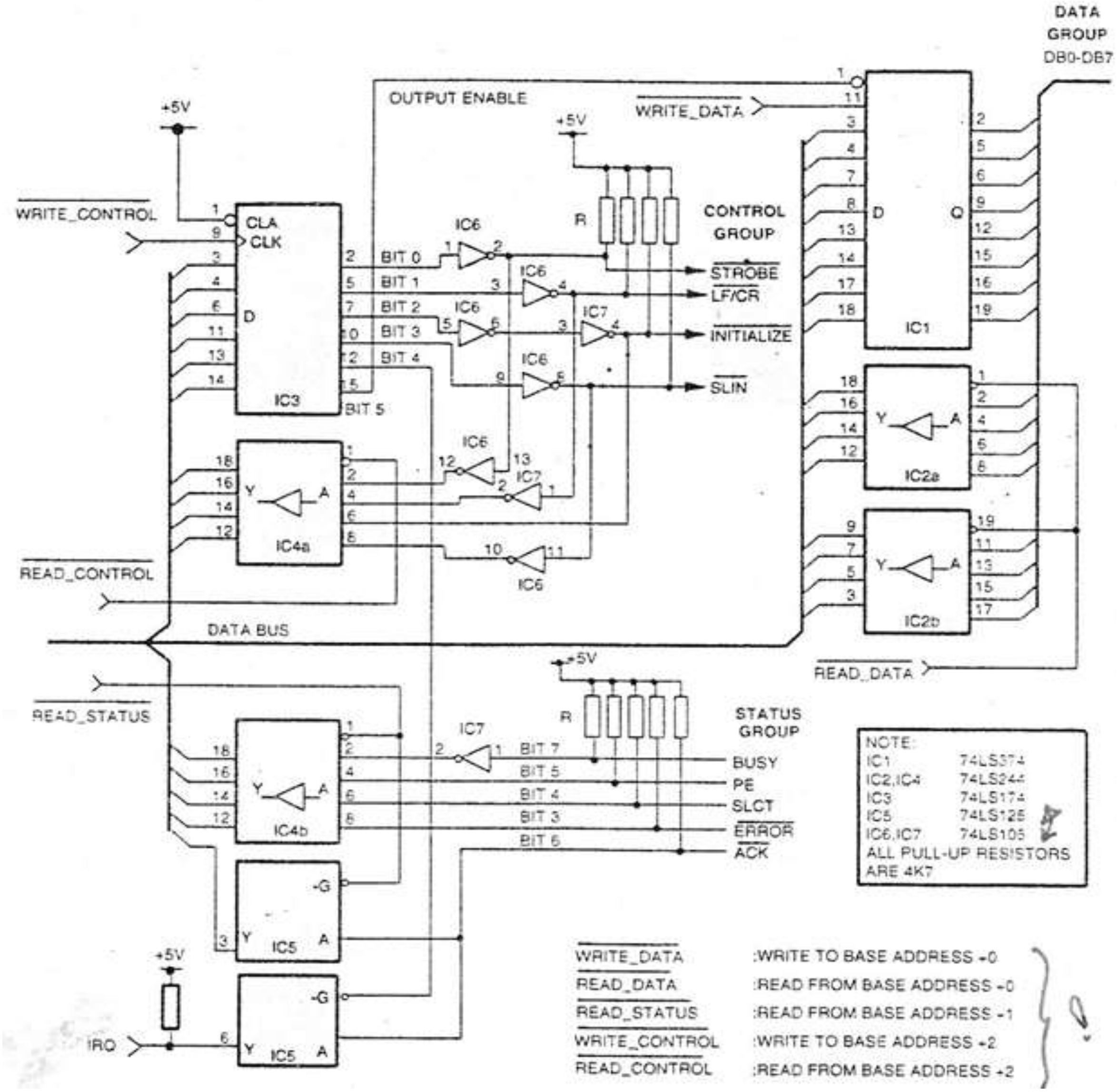
(a) Centronic connector on a pc viewed from the back of the pc
Connector type: 25 pin female D-type

(b) Centronic connector on a printer viewed from the back of the printer
Connector type: 36 pin female Centronic-type

Connected to printers
36-way Centronic male connector

Connected to computers
25-way D-type male connector

Figure P1.2



NOTE:

IC1	74LS374
IC2,IC4	74LS244
IC3	74LS174
IC5	74LS125
IC6,IC7	74LS105

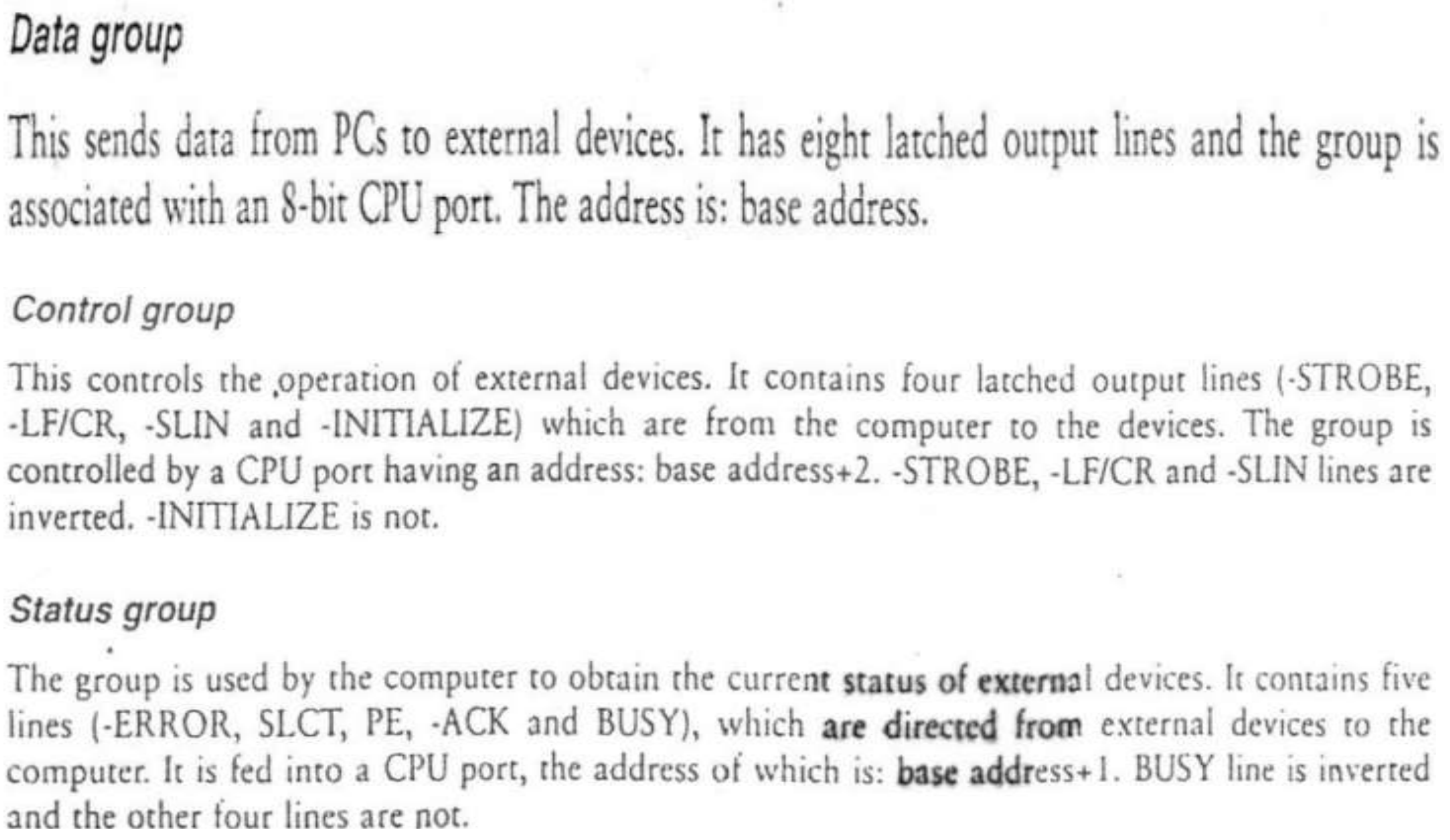
ALL PULL-UP RESISTORS ARE 4K7

WRITE_DATA	:WRITE TO BASE ADDRESS +0
READ_DATA	:READ FROM BASE ADDRESS +0
READ_STATUS	:READ FROM BASE ADDRESS +1
WRITE_CONTROL	:WRITE TO BASE ADDRESS +2
READ_CONTROL	:READ FROM BASE ADDRESS +2

Figure P1.3

As the lines in the data Port could supply a small current, they can supply power to a circuit which is connected to the Centronic port. The rate of data transfer through the Centronic port is greater than 1Mbyte/second.

In this section, the uni-directional Centronic Port is discussed in detail. The I/O lines in the Port are organized into three groups, namely, the Dara group, the Control and the Status group. Figure P1.4 gives the logic structure of the Centronic port.



Data group

This sends data from PCs to external devices. It has eight latched output lines and the group is associated with an 8-bit CPU port. The address is: base address.

Control group

This controls the ,operation of external devices. It contains four latched output lines (-STROBE, -LF/CR, -SLIN and -INITIALIZE) which are from the computer to the devices. The group is controlled by a CPU port having an address: base address+2. -STROBE, -LF/CR and -SLIN lines are inverted. -INITIALIZE is not.

Status group

The group is used by the computer to obtain the current status of external devices. It contains five lines (-ERROR, SLCT, PE, -ACK and BUSY), which are directed from external devices to the computer. It is fed into a CPU port, the address of which is: base address+1. BUSY line is inverted and the other four lines are not.

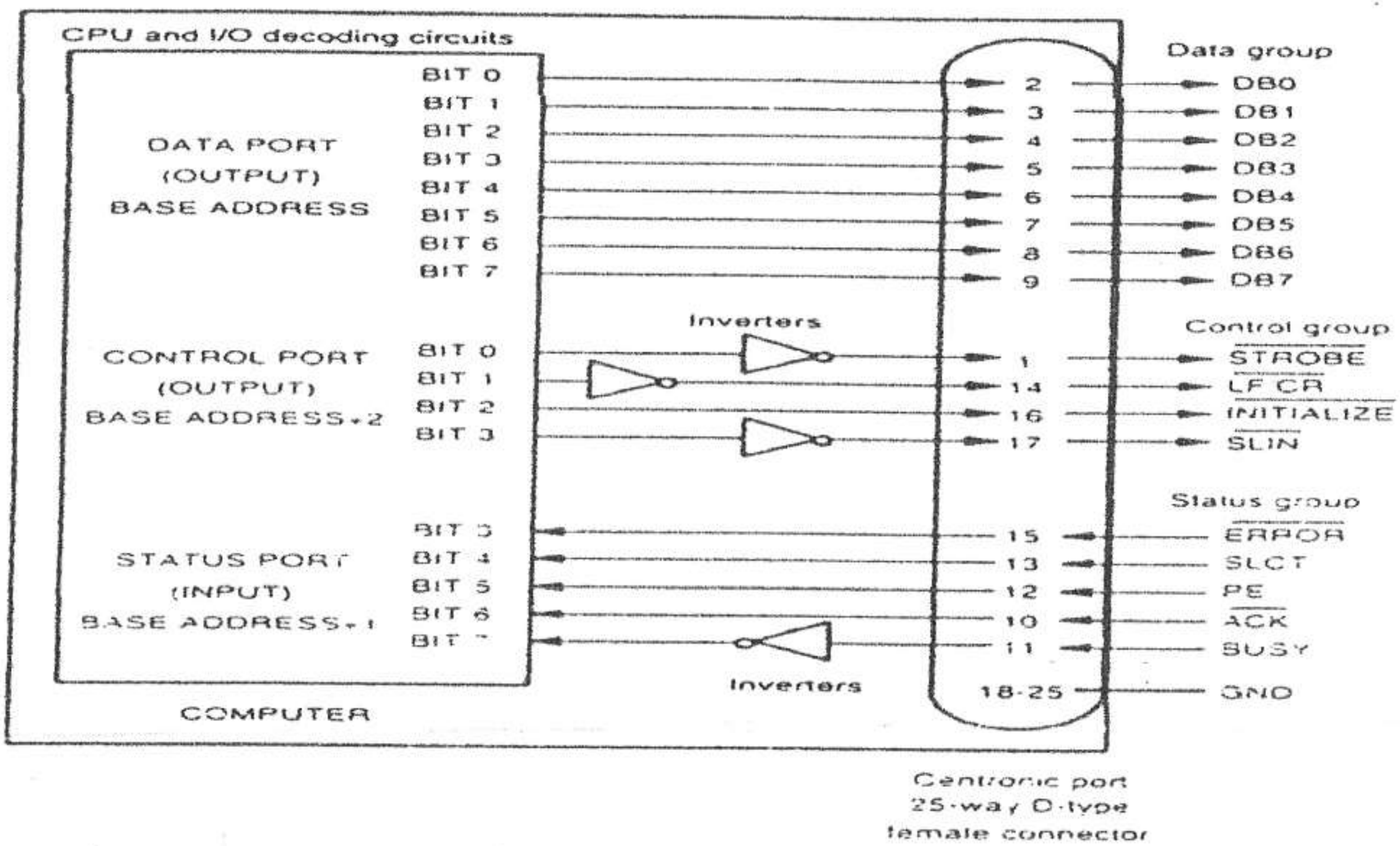

CPU and I/O decoding circuits
DATA PORT
(OUTPUT)
BASE ADDRESS
BIT 0
BIT 1
BIT 2
BIT 3
BIT 4
BIT 5
BIT 6
BIT 7
CONTROL PORT
(OUTPUT)
BASE ADDRESS+2
BIT 0
BIT 1
BIT 2
BIT 3
STATUS PORT
(INPUT)
BASE ADDRESS+1
BIT 3
BIT 4
BIT 5
BIT 6
BIT 7
Inverters
Inverters
COMPUTER
2
3
4
5
6
7
8
9
1
14
16
17
15
13
12
10
11
18-25
Data group
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
Control group
STROBE
LF CR
INITIALIZE
SLIN
Status group
ERROR
SLCT
PE
ACK
BUSY
GND
Centronic port
25-way D-type
female connector

Figure P1.4

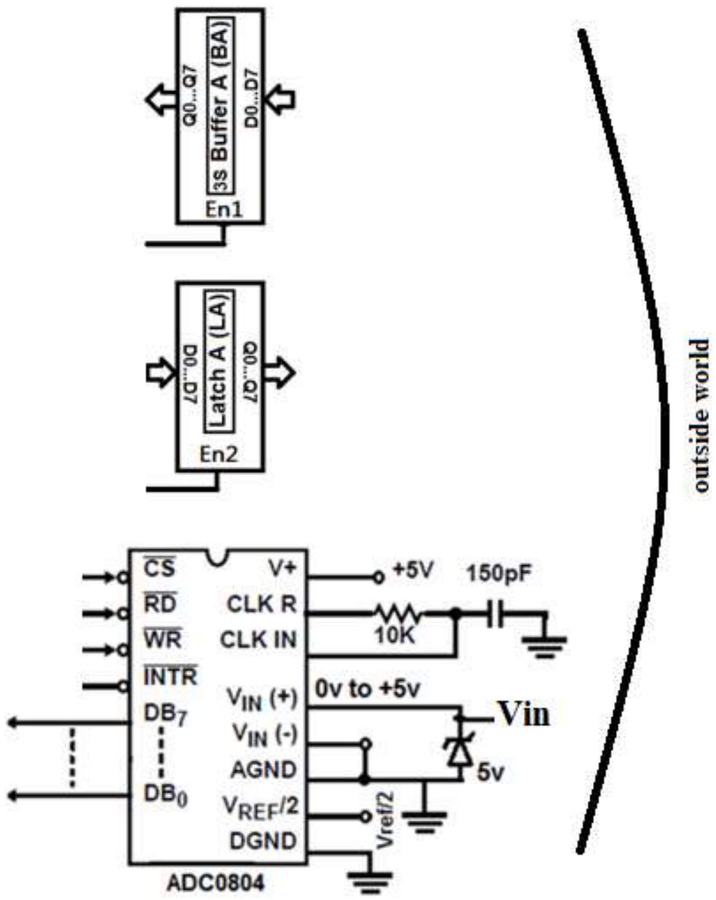
Lecture 7: QUESTIONS

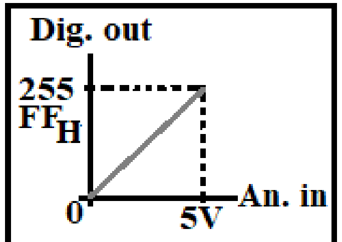
Note: Depend on the data and information given to you in the questions. You may make any necessary assumptions in case you think they are inevitable to solve the question.

Q1- Interface the ICs below to the parallel port. Then, Write a program which:

a. Inputs 8 bits of data through buffer BA and sends it to latch LA. The base address for the parallel port is 378H.

b. Reads a sample of Vin and displays it in its original analog value.

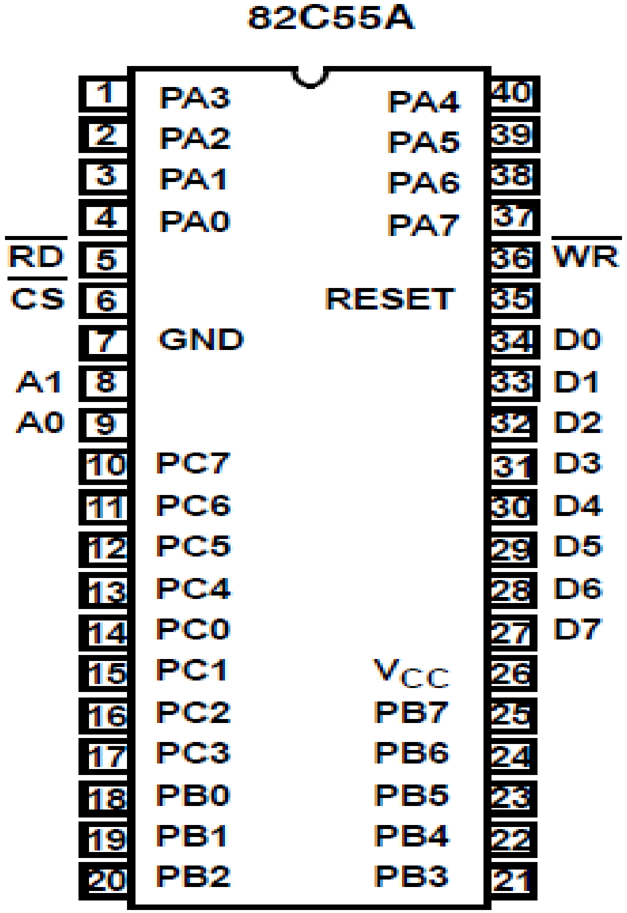




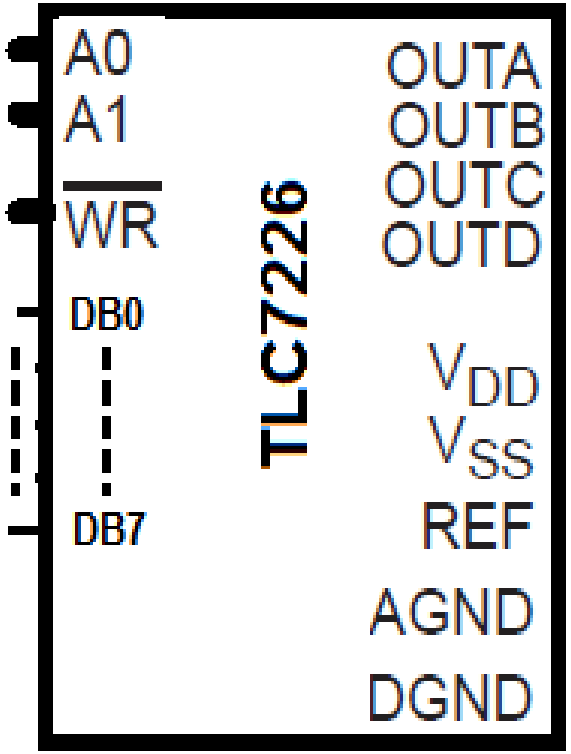
Out-In relationship for the ADC0804

......	Instruction…	Comments............	C7	C6	C5	C4	C3	C2	C1	C0
							▷○		▷○	▷○

Q2- Interface the 8255 PPI to the PC through the parallel bus. Then write a program to read data through port A and send it to port B (Considering that the PPI is already programmed in mode 0, and so that: (port A: I/P) (port B: O/P), and (port C: o/p). Consider the parallel port base address is 378H.



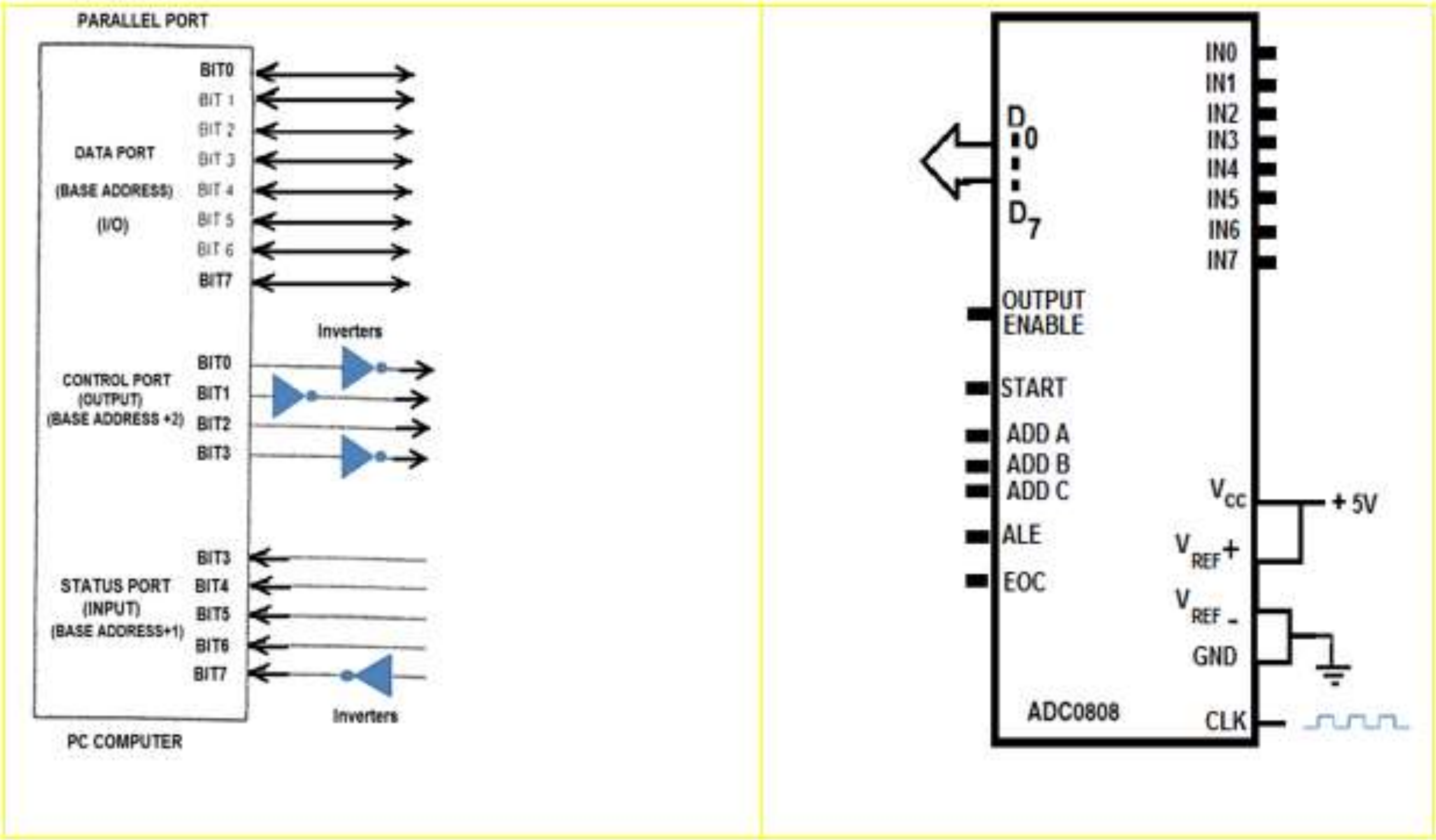
Q3 - Interface the TLC7226 to the PC through the parallel bus. Then write a program to send the value DE_H to (D/A – B). Consider the parallel port base address is 378H.



Lecture 8

Q4- A- Interface the A/D (ADC0808) in figure below to the PC parallel port.

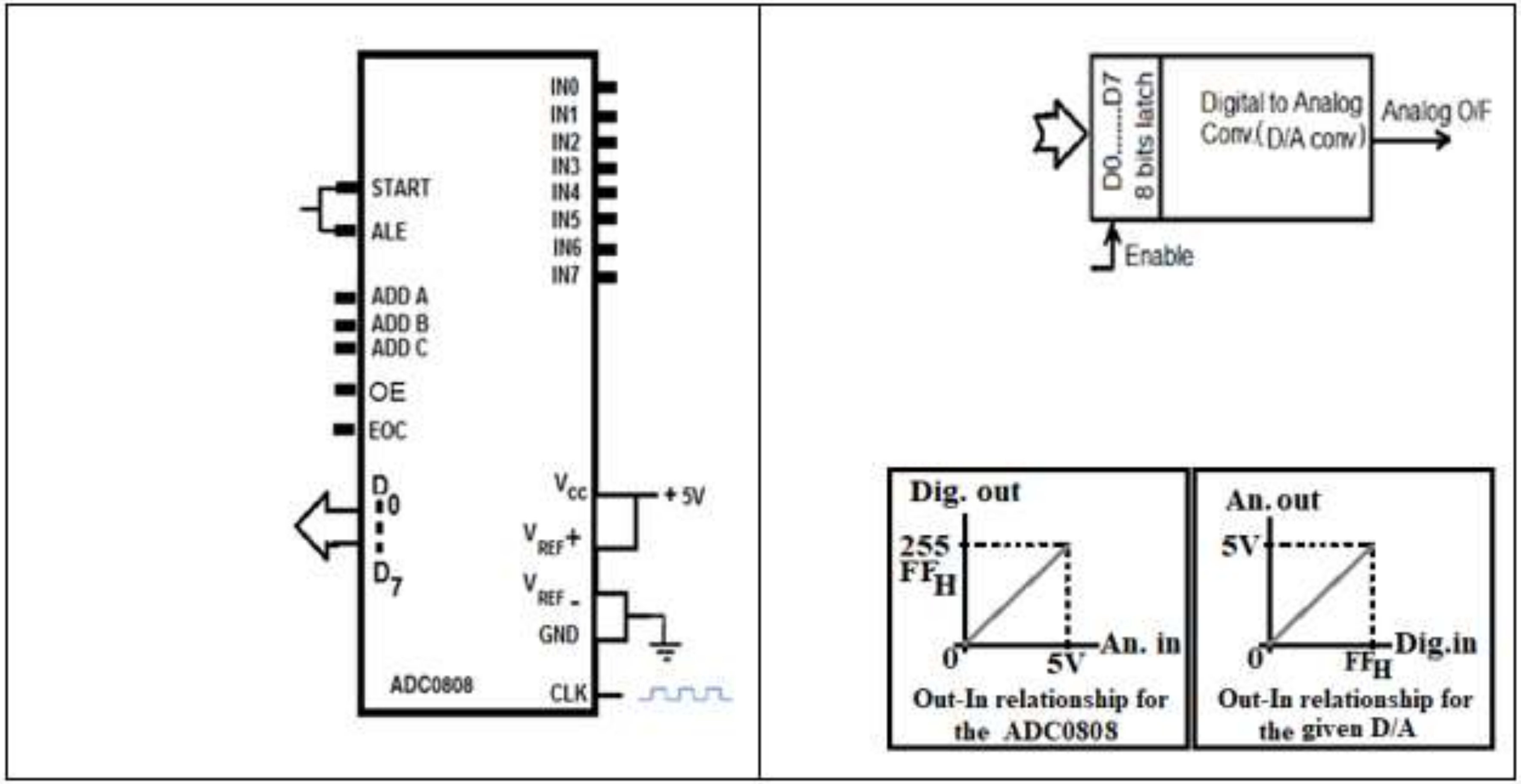
B- Write a program in pseudo language to order the A/D to start conversion after choosing input channel IN5. Consider the parallel port base address is 278_H.



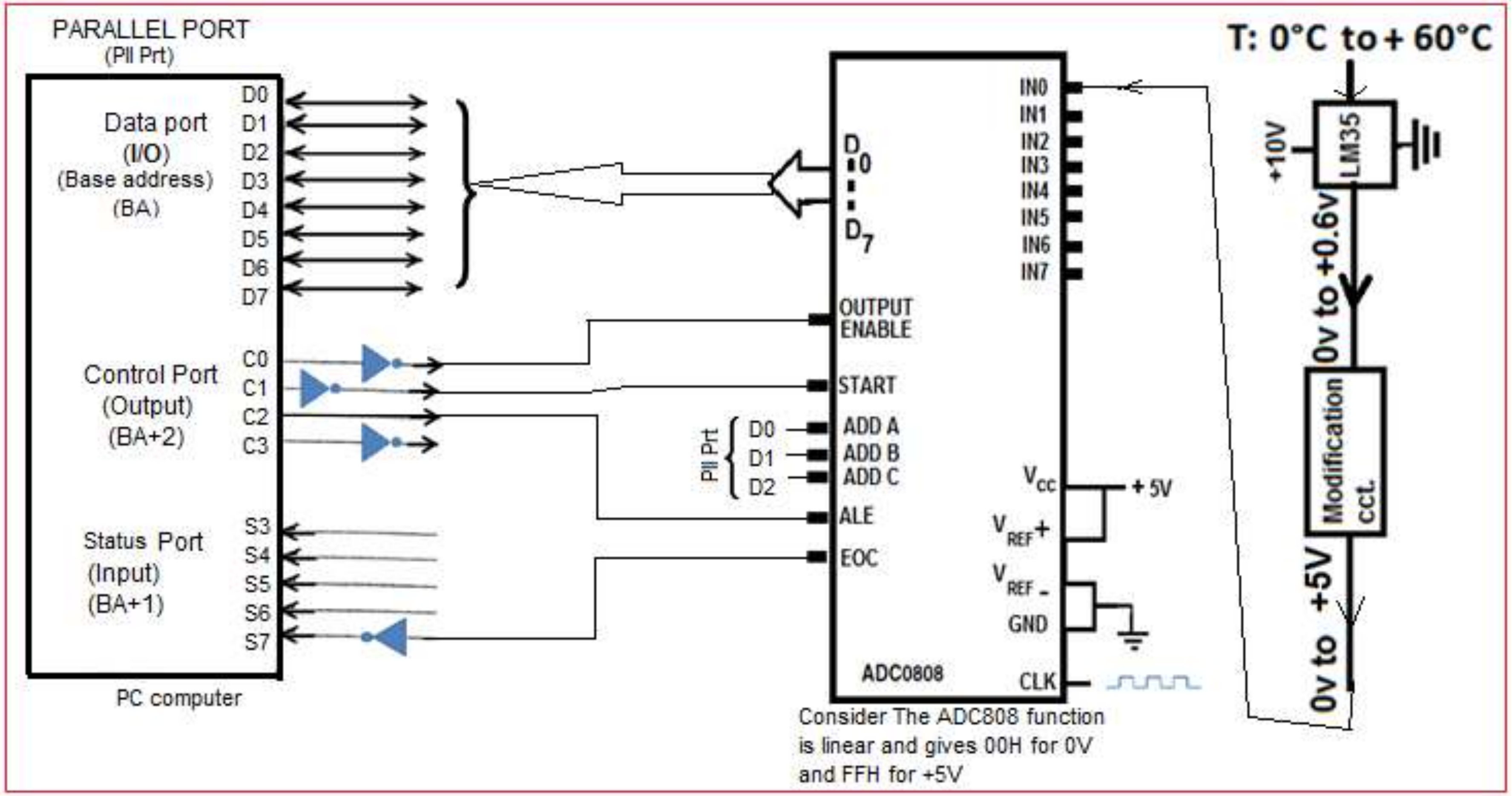
Q5- A- Interface simultaneously the ICs in figure below to the parallel port.

B- Write a program in pseudo language to: read a sample through input channel IN6, multiply the input value by 2, and then output the result of multiplication to the D/A.

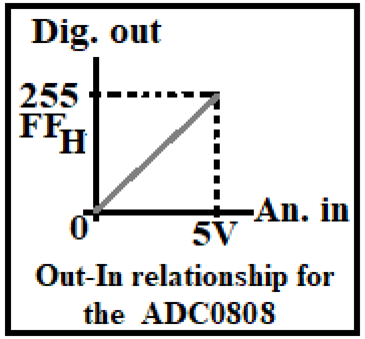
Consider the parallel port base address is 278_H.



Q6- Consider the interface circuit in figure below, write a program in pseudo language to input a sample of T and display it on screen in ⁰C . Consider the parallel port base address is 378_H.



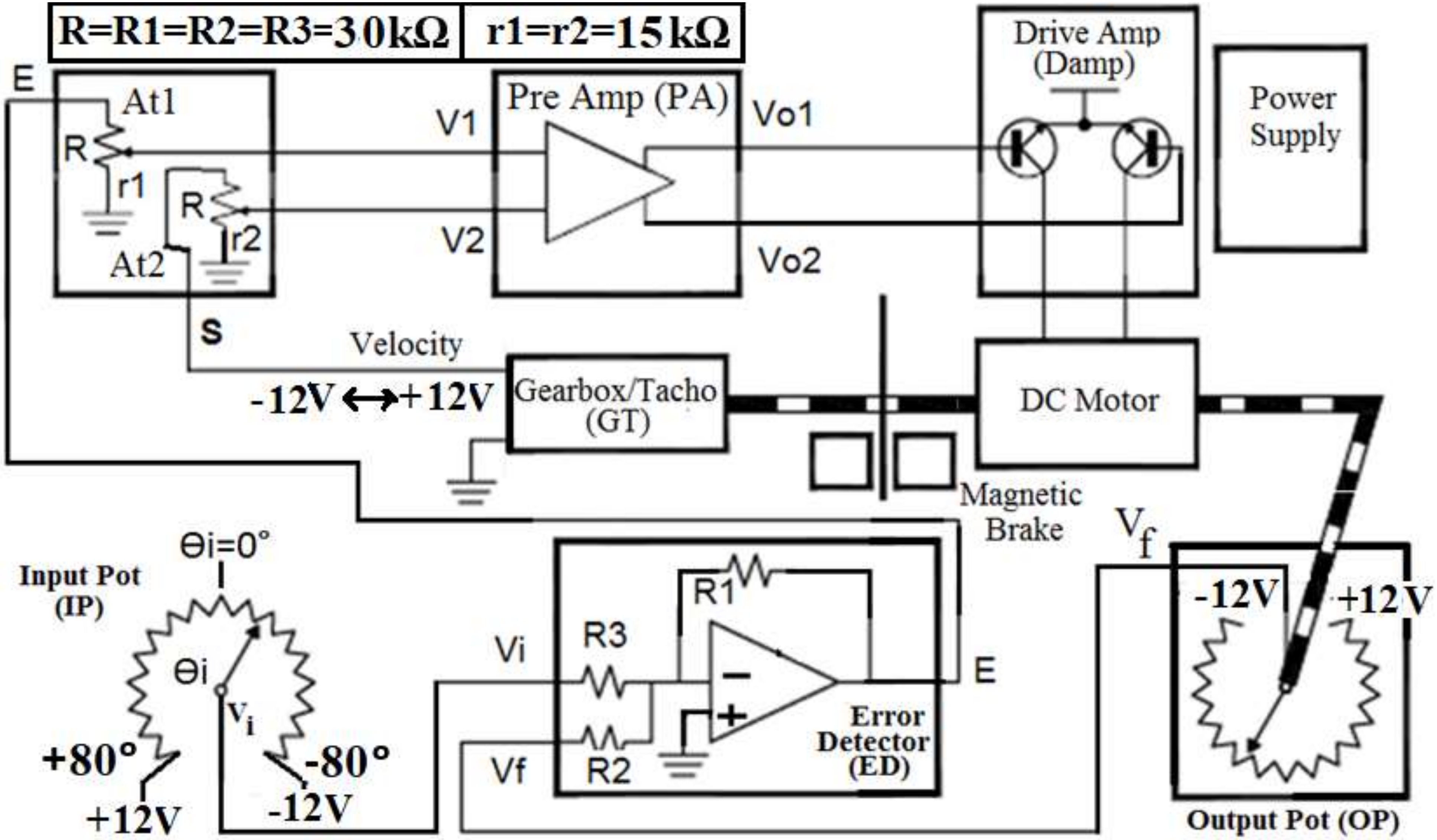
Consider The ADC808 function is linear and gives 00H for 0V and FFH for +5V



Out-In relationship for the ADC0808

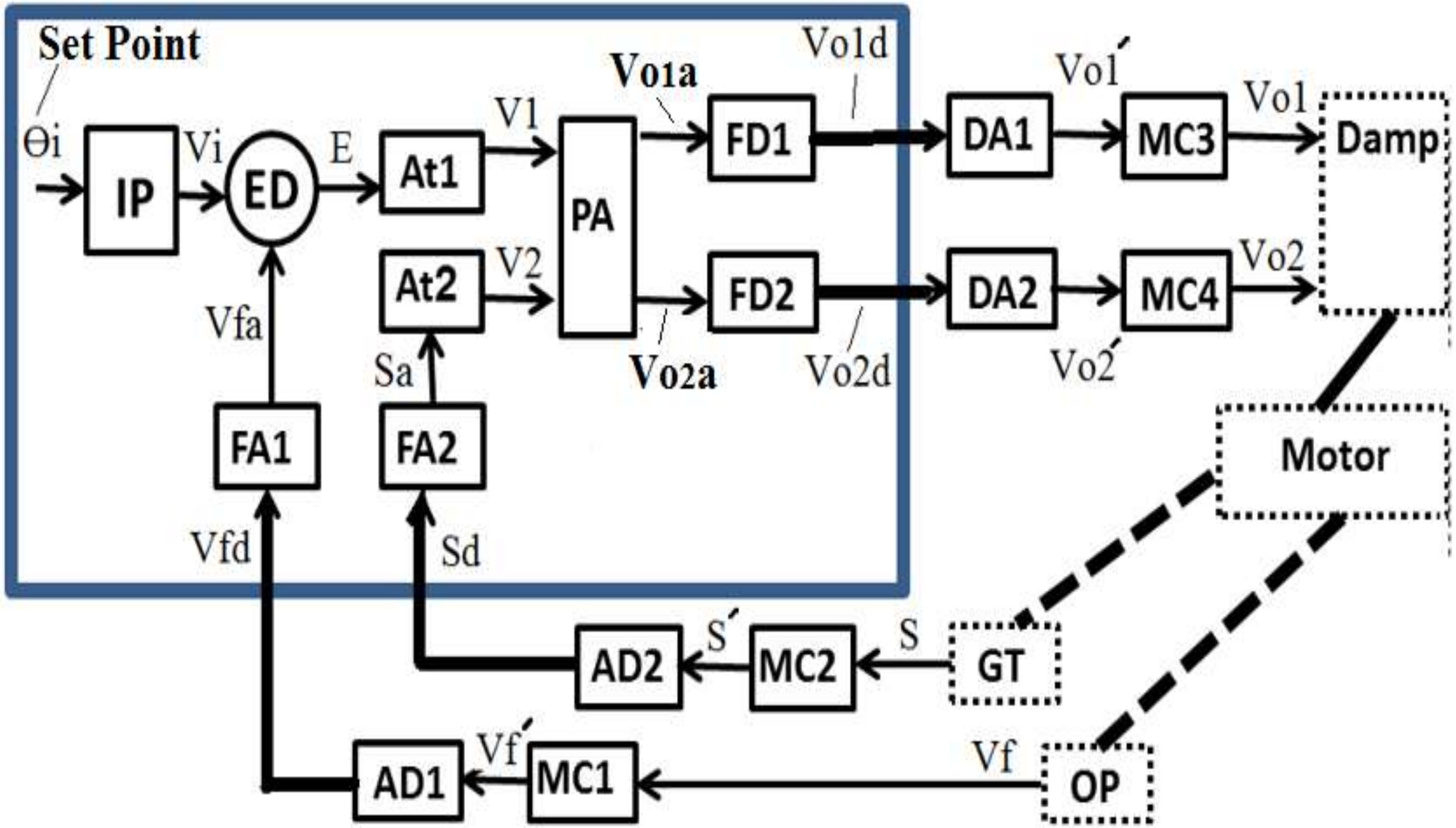
Lecture 9: CASE STUDY – 1

Figure (CS1a) represents a block diagram for a position control system. It is desired to use a digital computer to replace the components which its function could be implemented as software by the computer. The new block diagram for the new system when the computer is used is given in Figure (CS1b) .



R=R1=R2=R3=30kΩ
r1=r2=15kΩ
E
At1
R
r1
R
r2
At2
V1
V2
Pre Amp (PA)
Vo1
Vo2
Drive Amp
(Damp)
Power
Supply
S
Velocity
-12V ↔ +12V
Gearbox/Tacho
(GT)
DC Motor
Magnetic
Brake
V_f
θi=0°
Input Pot
(IP)
θi
V_i
+80°
-80°
+12V
-12V
Vi
Vf
R3
R2
R1
−
+
E
Error
Detector
(ED)
-12V
+12V
Output Pot (OP)

Figure CS1a



(MC: Modification Cct) (DA: Dig. to An. Converter which gives 0V for 0H and 10V for FFH)(AD: An. to Dig. Converter which gives 0H for 0V and FFH for 5v) (FA and FD are conversion functions)

Figure CS1b-1

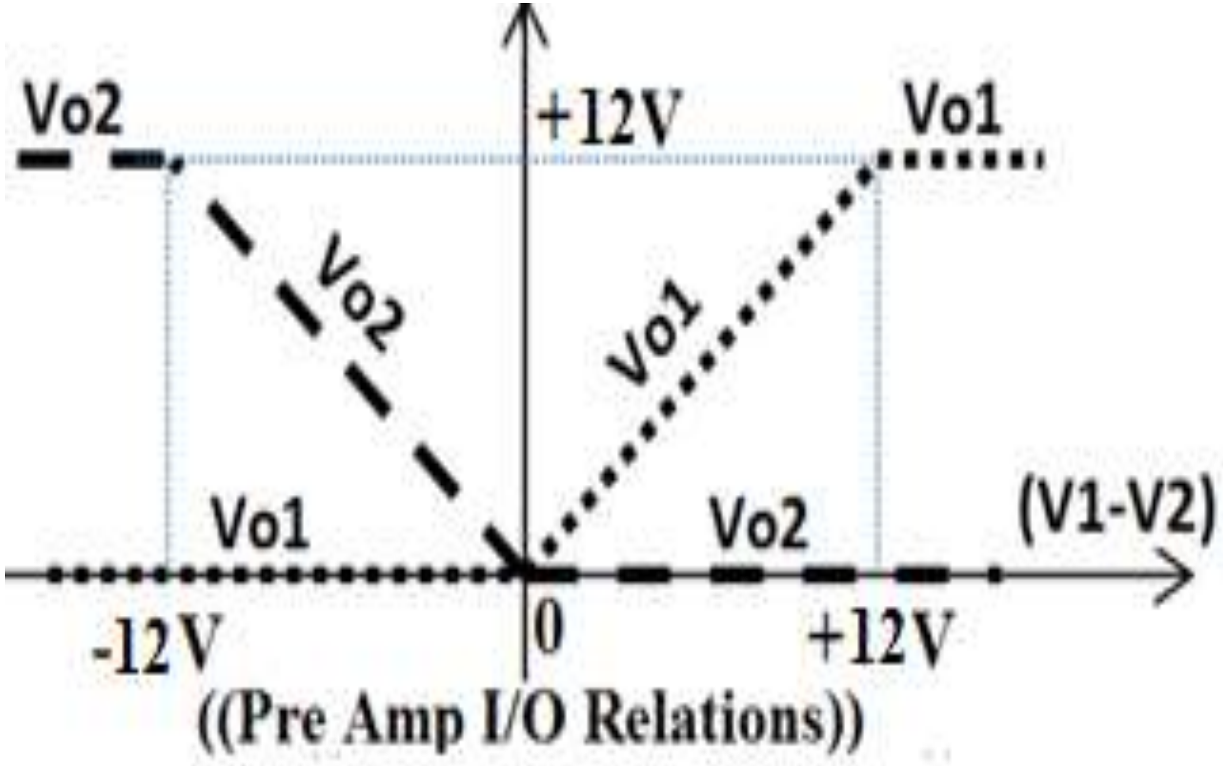
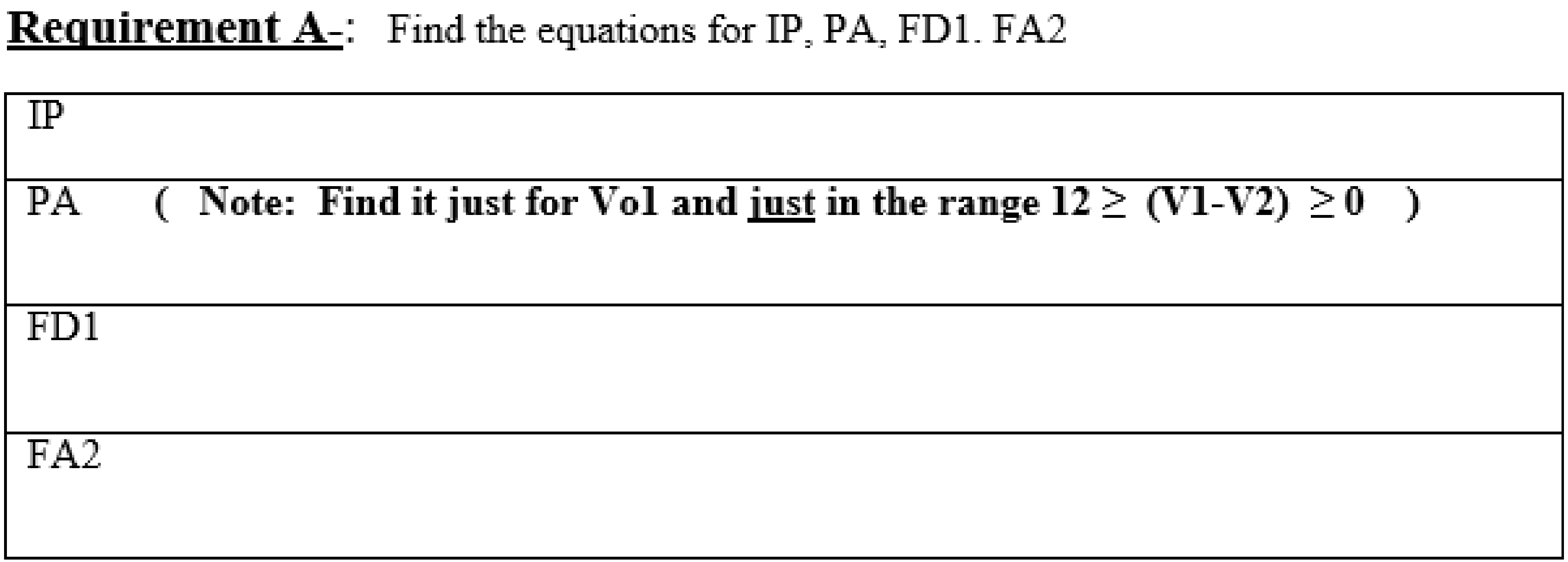


Figure CS1b-2



<u>Requirement A-</u>: Find the equations for IP, PA, FD1. FA2

IP
PA (**Note: Find it just for Vo1 and <u>just</u> in the range 12 ≥ (V1-V2) ≥ 0**)
FD1
FA2

Lecture 11

Requirement-B- Depending on Figure (CS1b) interface the system to the computer using the **ISA** bus and the circuits given in figure (CS1C1 and CS1C2) below.

Note: Consider that: (ADC0808 gives 0H for 0V and FFH for 5v), (AD7226 gives 0V for 0H and 12V for FFH).

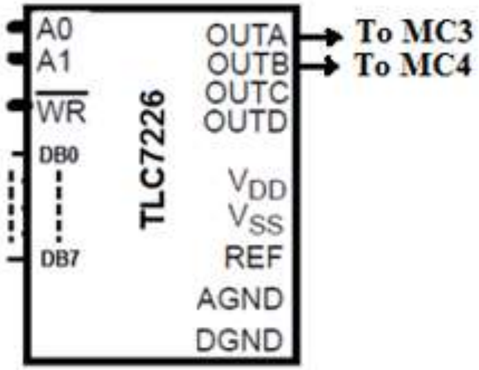
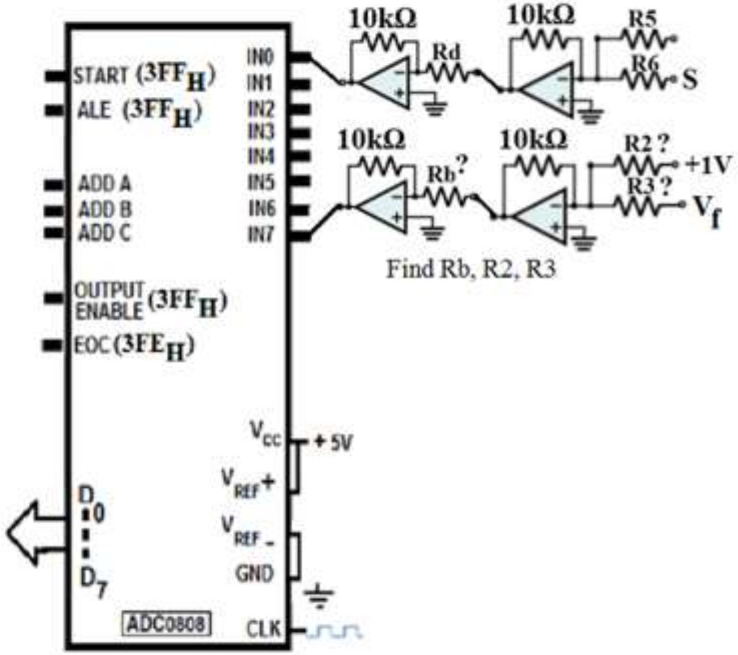


Figure CS1**C1** (Use addresses 304H to 307H)



Find Rb, R2, R3

Figure CS1**C2**

Requirement-C- Depending on your hardware design, write a program which does the followings steps:
- Read a sample of **S** (i.e. get its digital value **Sd**). - Output **Vo1d** to the D/A.

Control and systems Eng. Dept. – University of Technology

4th year-Computer Interfacing Course

1st term course – 2017/2018

Instructor: Dr Laith J. Saud

****

This document includes:

- Notes about:

- **Noise, opto-isolators, sample and hold units.**
- **PCI bus.**
- **Serial port.**

- Questions related to subjects above and some others.

- Class done evaluation tests and practice questions.

- Questions related to ISA bus and parallel port.

- Two more case studies.

Lecture 11

NOISE

Noise is a very important and wide issue. There are a lot of things to be discussed about noise, among which are:

- Definition
- Effect
- Model
- Classification (main: external and internal)
- Source
- Type
- Measuring
- Reducing or removing

Definition

- A noise to a certain system represents any unwanted and bad effects caused by any phenomena which lead to deviate system signals (in value and shape) and system action from the designed or wanted ones.

(((In electrical terms, noise is defined as the unwanted form of energy which tends to interface with the proper reception and the reproduction of transmitted signals.)))

- Noise is a must issue to care for in control and data acquisition systems to have them work correctly.
- In electrical systems, noise could spread through wired or wireless ways.
- Most laboratories and industrial environments contain abundant electrical-noise sources, including:

  AC power lines, heavy machinery, radio and TV stations, and a variety of electronic equipment. Radio stations generate high-frequency noise, while computers and other electronic equipment generate noise in all frequency ranges.
- Building a completely noise-free environment just for running tests and measurements is seldom a practical solution. Fortunately, simple devices and techniques such as using proper grounding methods, shielded and twisted wires, signal averaging methods, filters, and differential input voltage amplifiers can control the noise in most measurements. Some techniques prevent noise from entering the system, while others remove extraneous noise from the signal.

- There are a number of ways to classify noise. It can be subdivided according to

type, source, effect, or relation to the receiver, depending on circumstances.

In order to make the classification more compact, noise sources can be divided

into two main groups: • Noise whose sources are external to the receiver

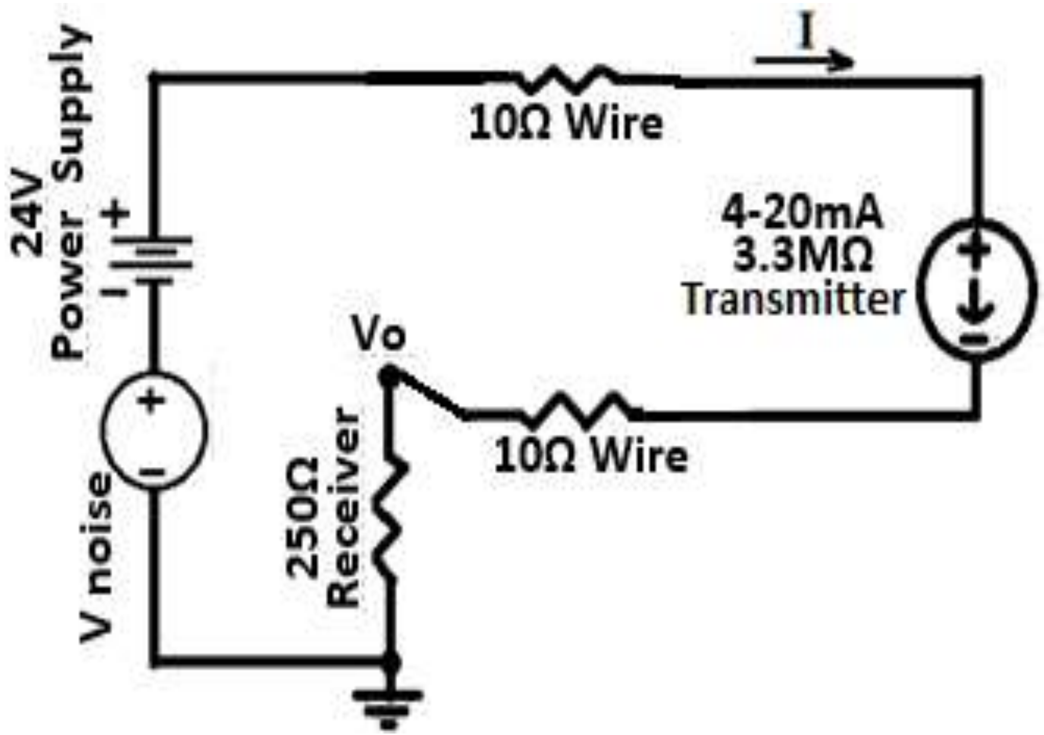
(External Noise) and • Noise source created within the receiver itself

(Internal Noise).

It is important to know that there are many different types of noise!!

EXAMPLE:

If the noise source in Figure below has an amplitude of 20 Volts, then what will be the noise voltage seen across the receiver at Vo?



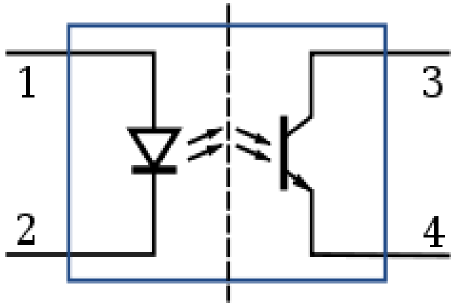
OPTO-ISOLATOR

In electronics, an **opto-isolator**, also called an **optocoupler**, **photocoupler**, or **optical isolator**, is "an electronic device designed to transfer electrical signals by utilizing light waves to provide coupling with electrical isolation between its input and output

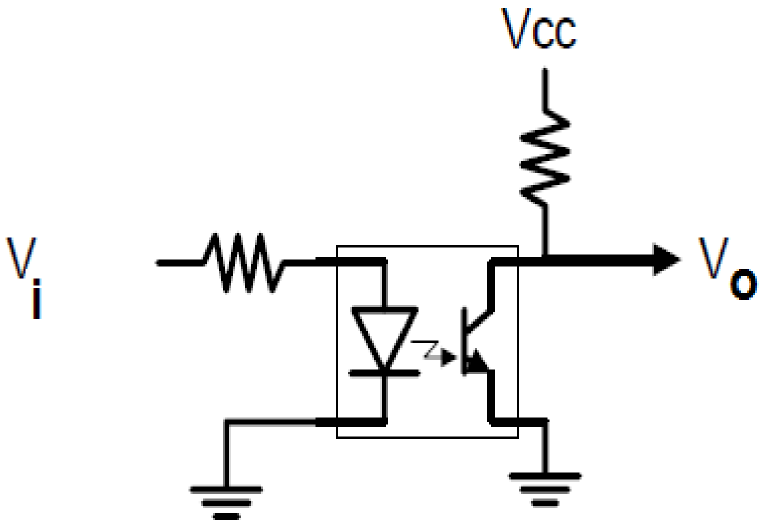
They are made up of a light emitting device, and a light sensitive device, all wrapped up in one package, but with no electrical connection between the two, just a beam of light. The light emitter is nearly always an LED. The light sensitive device may be a photodiode, phototransistor, or more esoteric devices such as thyristors, triacs etc. The cheapest kind have phototransistors.

The main purpose of an opto-isolator is "to prevent high voltages or rapidly changing voltages on one side of the circuit from damaging components or distorting transmissions on the other side". Commercially available opto-isolators withstand input-to-output voltages up to 10 kV and voltage transients with speeds up to 10 kV/μs.

Schematic diagram of an opto-isolator showing source of light (LED) on the left, dielectric barrier in the center, and sensor (phototransistor) on the right

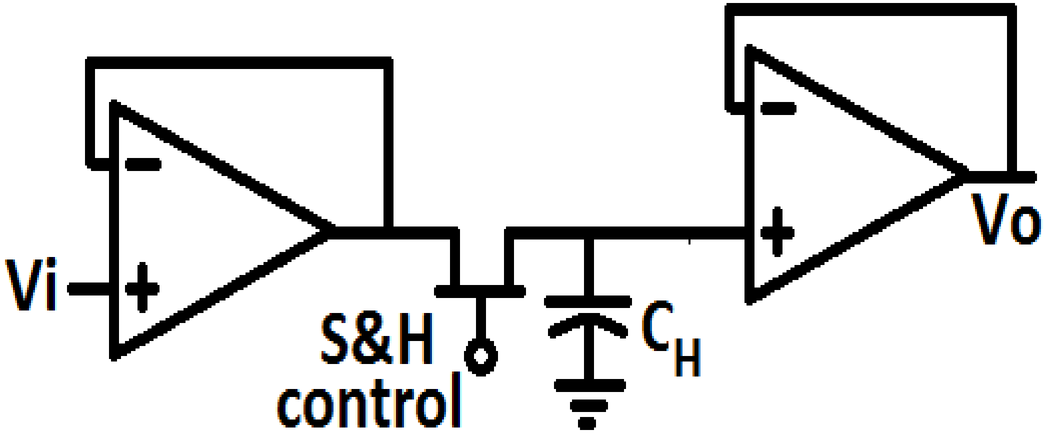


Below is a basic circuit diagram using one of these types for opto coupling :



SAMPLE AND HOLDS (S/H) –

Definition: Sample and hold circuit is a circuit used to sample an analog signal and to store its value for some length of time (for digital code conversion).



A typical circuit for S&H

Sample and Holds – Some notes:

- Mainly used in Analog-to-Digital Converters (ADC)
- Samples analog input signal and holds value between clock cycles
- Stable input value is required in many ADC-topologies
- Reduces ADC-error caused by internal ADC delay variations
- Sometimes referred to as Track and Hold (T/H)
- Important parameters for S/H’s
- Hold step: Voltage error during S/H-transition
- Signal isolation in hold mode
- Input signal tracking speed in sample mode
- Droop rate in hold mode: Small change in output voltage
- Aperture jitter: Sampling time uncertainty

Note: there is a lot to be discussed concerning S&H.

Lecture 12

PCI BUS PROTOCOL -- PCI = PERIPHERAL COMPONENT INTERCONNECT

PCI is a synchronous bus architecture with all data transfers being performed relative to a system clock (CLK).

PCI implements a 32-bit multiplexed Address and Data bus (AD[31:0]).

At 33 MHz, a 32-bit slot supports a maximum data transfer rate of 132 MBytes/sec.

PCI bus cycles are initiated by driving an address onto the AD[31:0] signals during the first clock edge called the *address phase*. The address phase is signaled by the activation of the FRAME# signal. The next clock edge begins the first of one or more *data phases* in which data is transferred over the AD[31:0] signals.

In PCI terminology, data is transferred between an *initiator* which is the bus master, and a *target* which is the bus slave. The initiator drives the C/BE[3:0]# signals during the address phase to signal the type of transfer (memory read, memory write, I/O read, I/O write, etc.). During data phases the C/BE[3:0]# signals serve as byte enable to indicate which data bytes are valid. Both the initiator and target may insert wait states into the data transfer by deasserting the IRDY# and TRDY# signals. Valid data transfers occur on each clock edge in which both IRDY# and TRDY# are asserted.

A PCI bus transfer consists of one address phase and any number of data phases.

Both the initiator and target may terminate a bus transfer sequence at any time. The initiator signals completion of the bus transfer by deasserting the FRAME# signal during the last data phase. A target may terminate a bus transfer by asserting the STOP# signal. When the initiator detects an active STOP# signal, it must terminate the current bus transfer and re-arbitrate for the bus before continuing. Initiators arbitrate for ownership of the bus by asserting a REQ# signal to a central arbiter. The arbiter grants ownership of the bus by asserting the GNT# signal. REQ# and GNT# are unique on a per slot basis allowing the arbiter to implement a bus fairness algorithm. Arbitration in PCI is "hidden" in the sense that it does not consume clock cycles. The current initiator's bus transfers are overlapped with the arbitration process that determines the next owner of the bus.

PCI defines support for both 5 Volt and 3.3 Volt signaling levels.

4.0 PCI Signal Descriptions

4.1 System Pins

CLK

Clock provides the timing reference for all transfers on the PCI bus. All PCI signals except reset and interrupts are sampled on the rising edge of the CLK signal. All bus timing specifications are defined relative to the rising edge.

RST#

Reset is driven active low to cause a hardware reset of a PCI device.

4.2 Address and Data Pins

AD[31:0]

Address and Data are multiplexed onto these pins. AD[31:0] transfers a 32-bit physical address during "address phases", and transfers 32-bits of data information during "data phases". An address phase occurs during the clock following a high to low transition on the FRAME# signal. A data phase occurs when both IRDY# and TRDY# are asserted low. During write transactions the initiator drives valid data on AD[31:0] during each cycle it drives IRDY# low. The target drives TRDY# low when it is able to accept the write data. When both IRDY# and TRDY# are low, the target captures the write data and the transaction is completed. For read transactions the opposite occurs. The target drives TRDY# low when valid data is driven on AD[31:0], and the initiator drives IRDY# low when it is able to accept the data. When both IRDY# and TRDY# are low, the initiator captures the data and the transaction is completed. Bit 31 is the most significant AD bit. Bit 0 is the least significant AD bit.

C/BE[3:0]#

Bus Command and Byte Enables are multiplexed onto these pins. During the address phase of a transaction these signals carry the bus command that defines the type of transfer to be performed. See the table below for a list of valid bus command codes. During the data phase of a transaction these signals carry byte enable information. C/BE[3]# is the byte enable for the most significant byte (AD[31:24]) and C/BE[0]# is the byte enable for the lease significant byte (AD[7:0]). The C/BE[3:0]# signals are driven only by the initiator and are actively driven through the all address and data phases of a transaction**.** **))**

4.3 Interface Control Pins

FRAME#

Cycle Frame is driven low by the initiator to signal the start of a new bus transaction. The address phase occurs during the first clock cycle after a high to low transition on the FRAME# signal. If the initiator intends to perform a transaction with only a single data phase, then it will return FRAME# back high after only one cycle. If multiple data phases are to be performed, the initiator will hold FRAME# low in all but the last data phase. The initiator signals its intent to perform a *master initiated termination* by driving FRAME# high during the last data phase of a transaction. During a *target initiated termination* the initiator will continue to drive FRAME# low through the end of the transaction.

IRDY#

Initiator Ready is driven low by the initiator as an indication it is ready to complete the current data phase of the transaction. During writes it indicates the initiator has placed valid data on AD[31:0]. During reads it indicates the initiator is ready to accept data on AD[31:0]. Once asserted, the initiator holds IRDY# low until TRDY# is driven low to complete the transfer, or the target uses the STOP# signal to terminate without performing the data transfer. IRDY# permits the initiator to insert wait states as needed to slow the data transfer.

TRDY#

Target Ready is driven low by the target as an indication it is read to complete the current data phase of the transaction. During writes it indicates the target is ready to accept data on AD[31:0]. During reads it indicates the target has placed valid data on the AD[31:0] signals. Once asserted, the target holds TRDY# low until IRDY# is driven low to complete the transfer. TRDY# permits the target to insert wait states as needed to slow the data transfer.

STOP#

Stop is driven low by the target to request the initiator terminate the current transaction. In the event that a target requires a long period of time to respond to a transaction, it may use the STOP# signal to suspend the transaction so the bus can be used to perform other transfers in the interim. When the target terminates a transaction without performing any data phases it is called a *retry*. If one or more data phases are completed before the target terminates the transaction, it is called a *disconnect*. A retry or disconnect signals the initiator that it must return at a later time to attempt performing the transaction again. In the event of a fatal error such as a hardware problem the target may use STOP# and DEVSEL# to signal an abnormal termination of the bus transfer called a *target abort*. The initiator can use the target abort to signal system software that a fatal error has been detected.

LOCK#

Lock may be asserted by an initiator to request exclusive access for performing multiple transactions with a target. It prevents other initiators from modifying the locked addresses until the agent initiating the lock can complete its transaction.

DEVSEL#

Device Select is driven active low by a PCI target when it detects its address on the PCI bus. DEVSEL# may be driven one, two, or three clocks following the address phase. DEVSEL# must be asserted with or prior to the clock edge in which the TRDY# signal is asserted. Once DEVSEL# has been asserted, it cannot be deasserted until the last data phase has completed, or the target issues a target abort. If the initiator never receives an active DEVSEL# it terminates the transaction in what is termed a *master abort*.

4.4 Arbitration Pins (Initiator Only)

REQ#

Request is used by a PCI device to request use of the bus.

GNT#

Grant indicates that a PCI device's request to use the bus has been granted.

4.5 Error Reporting Pins

4.6 Interrupt Pins

4.7 Cache Support Pins (Optional)

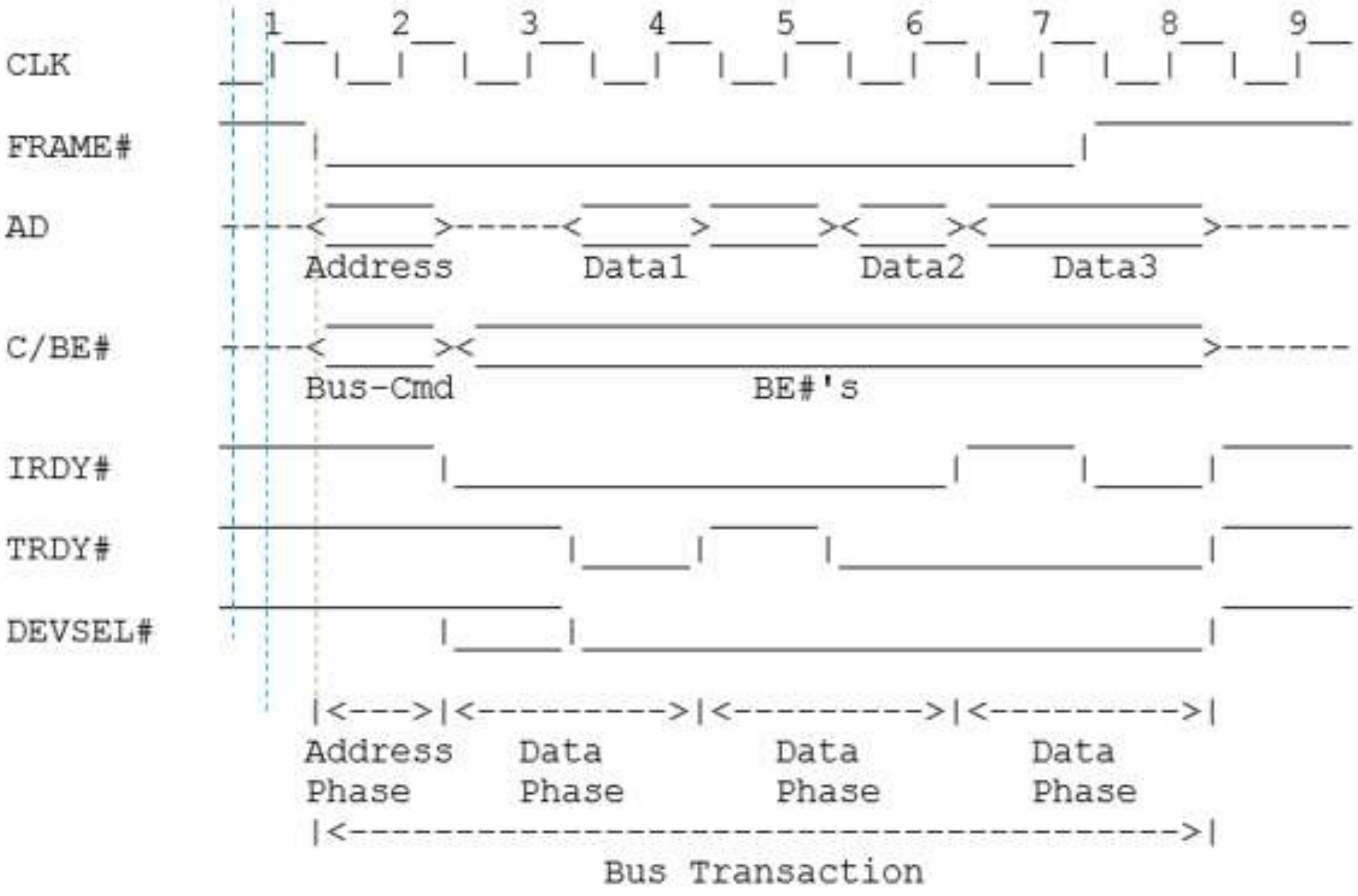
4.8 Additional Pins

5.0 PCI Bus Timing Diagrams

5.1 Read Transaction

Read Transaction

The following timing diagram illustrates a read transaction on the PCI bus:



The following is a cycle by cycle description of the read transaction:

- Cycle 1 - The bus is idle.
- Cycle 2 - The initiator asserts a valid address and places a read command on the C/BE# signals. This is the address phase.
- Cycle 3 - The initiator tri-states the address in preparation for the target driving read data. The initiator now drives valid byte enable information on the C/BE# signals. The initiator asserts IRDY# low indicating it is ready to capture read data. The target asserts DEVSEL# low (in this cycle or the next) as an acknowledgment it has positively decoded the address. The target drives TRDY# high indicating it is not yet providing valid read data.
- Cycle 4 - The target provides valid data and asserts TRDY# low indicating to the initiator that data is valid. IRDY# and TRDY# are both low during this cycle causing a data transfer to take place. The initiator captures the data. This is the first data phase.

- Cycle 5 - The target deasserts TRDY# high indicating it needs more time to prepare the next data transfer.
- Cycle 6 - The second data phase occurs as both IRDY# and TRDY# are low. The initiator captures the data provided by the target.
- Cycle 7 - The target provides valid data for the third data phase, but the initiator indicates it is not ready by deasserting IRDY# high.
- Cycle 8 - The initiator re-asserts IRDY# low to complete the third data phase. The initiator captures the data provided by the target. The initiator drives FRAME# high indicating this is the final data phase (master termination).
- Cycle 9 - FRAME#, AD, and C/BE# are tri-stated, as IRDY#, TRDY#, and DEVSEL# are driven inactive high for one cycle prior to being tri-stated.

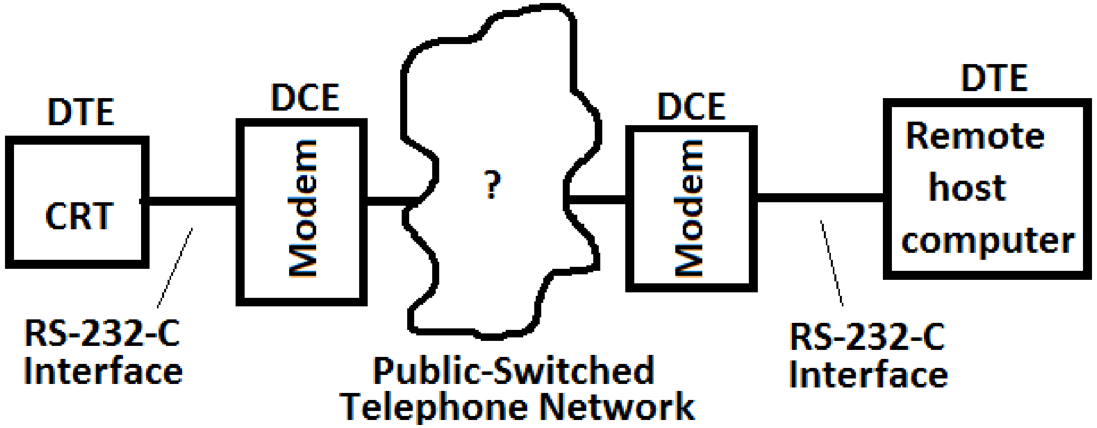
Lecture 13

SERIAL PORT (Namely: The RS-232)

Definition:

An Asynchronous port on the computer used to connect a serial device to the computer and capable of transmitting one bit at a time.

Serial connection was intended to connect devices separated by long distances using the phone network.
The Rs-232 standard for the serial communication ports was developed by EIA American company.
The RS-232-C was intended to address local interfaces associated with long-distance data communications that that involve the telephone network.
Below is an example:



The fact that RS232_C was intended to address local interfaces associated with long-distance data communications that that involve the telephone network has not kept it from being applied to a wide variety of short distance communications interfaces, such as computer to terminal , computer to printer , and even computer to disk.   RS232-c was really never intended to become the short distance interface standard as it happened later. The null modem is no more than an elegant kludge to make such devices as computers and terminals (normally DTEs) look like modems (DCEs) so that Rs 2323-C will apply.

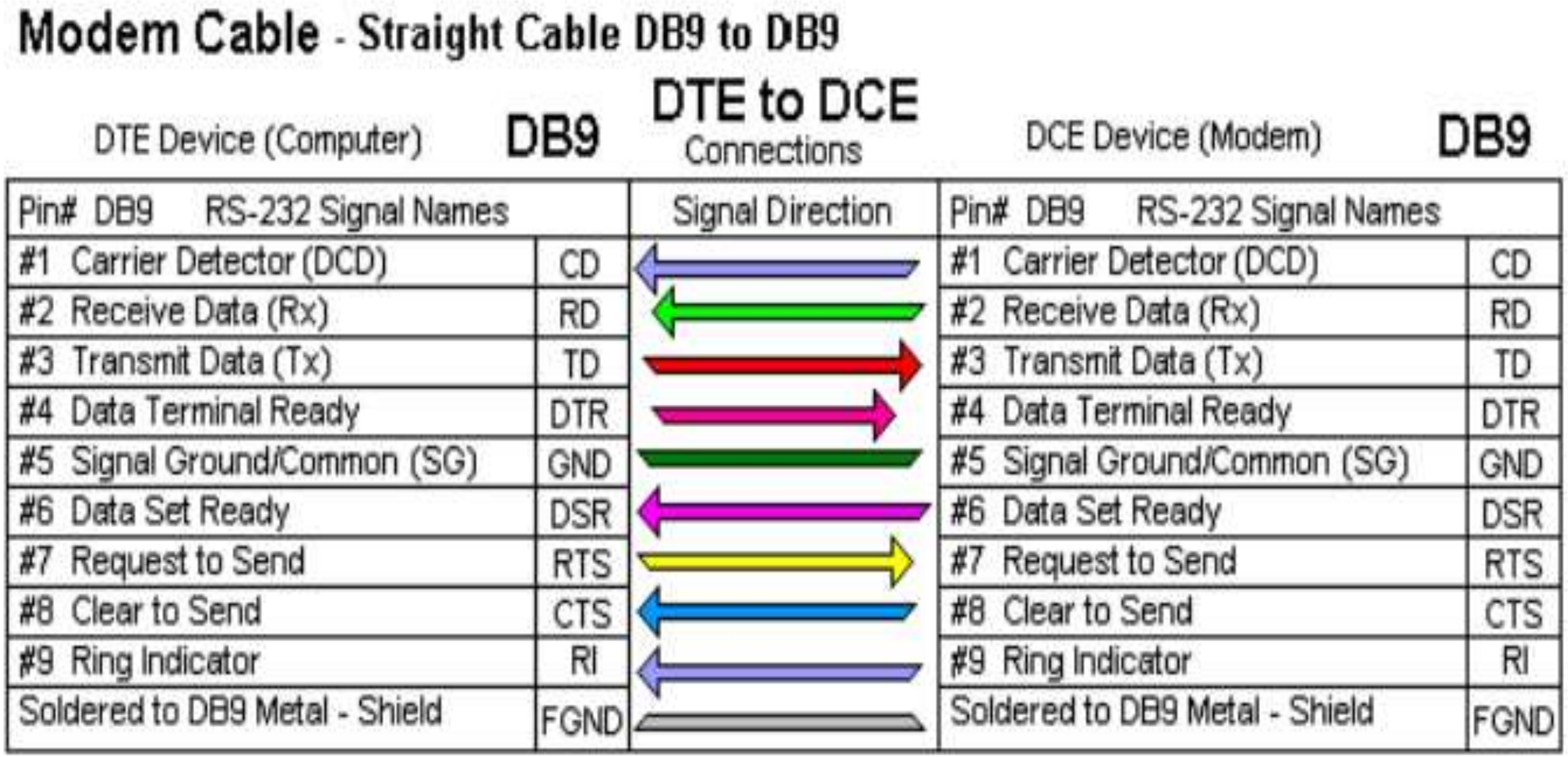
DTEs and DCEs

There are two types of RS-232 ports, DTE and DCE type.
DTE (**D**ata **T**erminal **E**quipment): Generally Computer or Terminal
DCE (**D**ata **C**ommunications **E**quipment): Modem
The signal names and pin numbers are the same for both, but signal flow is opposite! The pin labeled Tx can be input, and Rx the output.

The two ports types are complementary, the **Output** signals on a DTE port are **Inputs** to a DCE port, and **Output** signals on a DCE port are **Inputs** to a DTE port. The signal names match each other and connect pin for pin. Signal flow is in the direction of the arrows. Below are some connection cases for serial port.

Connecting two devices using RS-232 ((*Hardware Handshaking Connections)*)

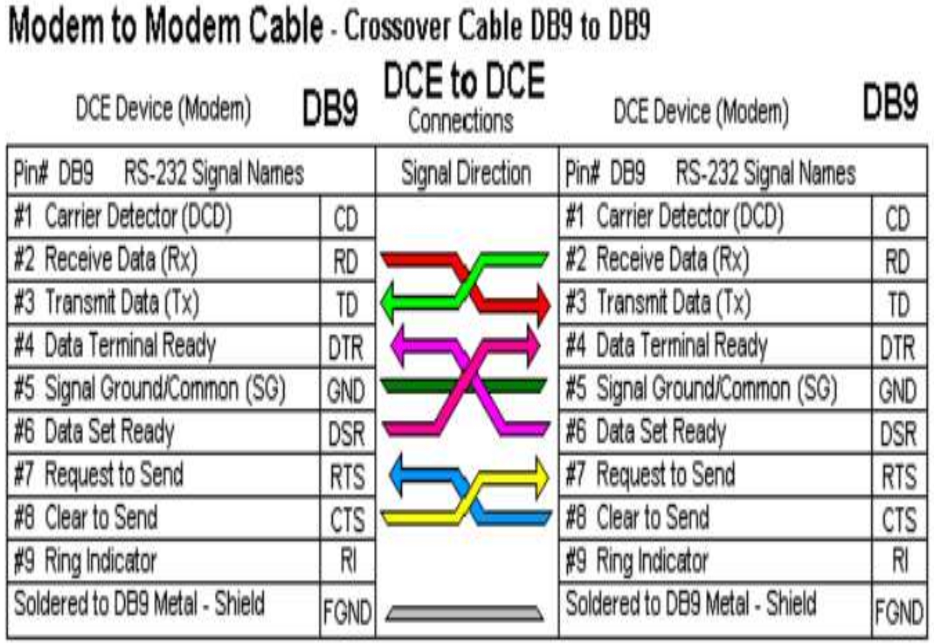
Rule of Thumb: When connecting a DTE device to a DCE device, match the signal names. When connecting two DTE or two DCE devices together, use a Crossover cable. (TD crosses to RD, RTS to CTS, DTR to DSR as shown in Modem to Modem connections. The cable for two computers (DTE) also simulates modem connections to CD/DSR, so it is commonly called a "Null Modem" cable.



Modem Cable - Straight Cable DB9 to DB9

DTE Device (Computer) **DB9** | **DTE to DCE** Connections | DCE Device (Modem) **DB9**

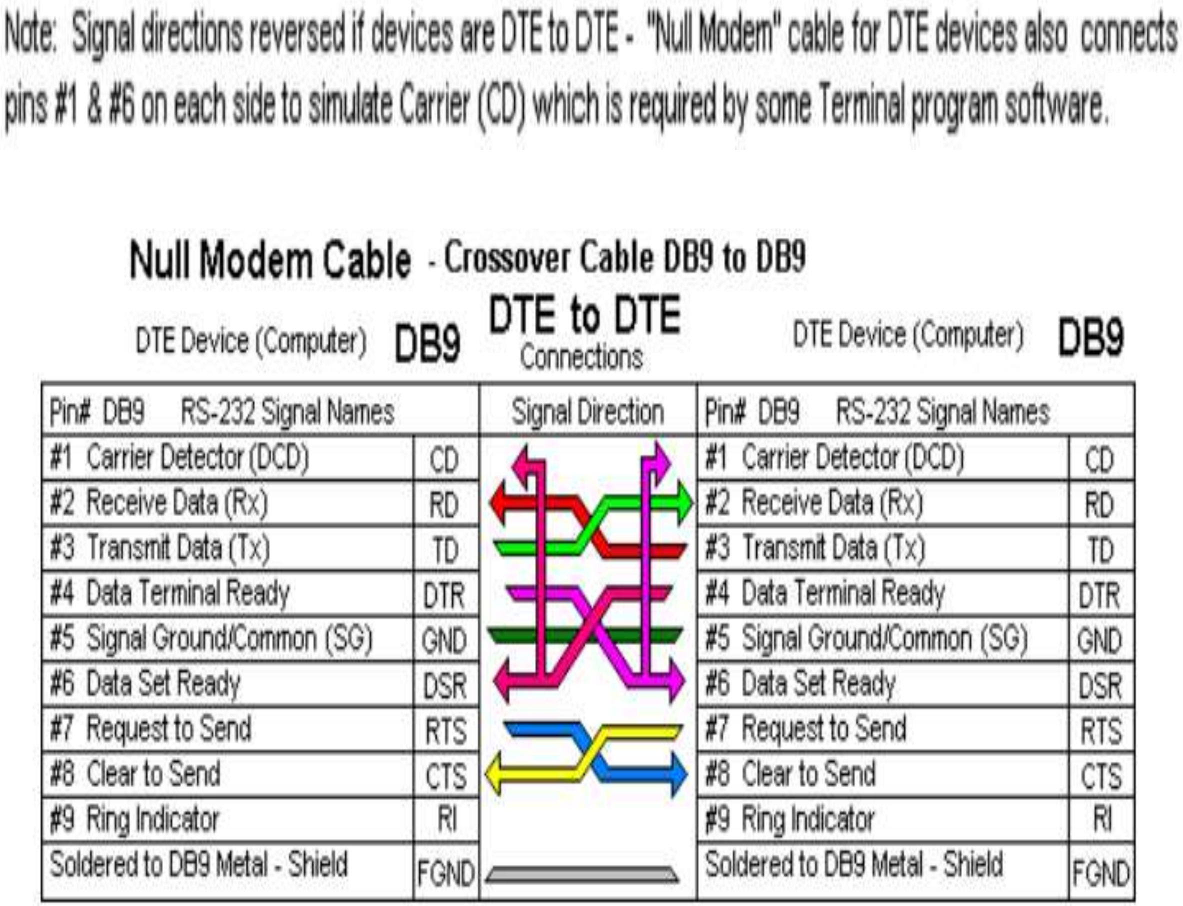
Pin# DB9 RS-232 Signal Names		Signal Direction	Pin# DB9 RS-232 Signal Names	
#1 Carrier Detector (DCD)	CD	←	#1 Carrier Detector (DCD)	CD
#2 Receive Data (Rx)	RD	←	#2 Receive Data (Rx)	RD
#3 Transmit Data (Tx)	TD	→	#3 Transmit Data (Tx)	TD
#4 Data Terminal Ready	DTR	→	#4 Data Terminal Ready	DTR
#5 Signal Ground/Common (SG)	GND	—	#5 Signal Ground/Common (SG)	GND
#6 Data Set Ready	DSR	←	#6 Data Set Ready	DSR
#7 Request to Send	RTS	→	#7 Request to Send	RTS
#8 Clear to Send	CTS	←	#8 Clear to Send	CTS
#9 Ring Indicator	RI	←	#9 Ring Indicator	RI
Soldered to DB9 Metal - Shield	FGND	—	Soldered to DB9 Metal - Shield	FGND



Modem to Modem Cable - Crossover Cable DB9 to DB9

DCE Device (Modem) **DB9** — **DCE to DCE** Connections — DCE Device (Modem) **DB9**

Pin# DB9 RS-232 Signal Names		Signal Direction	Pin# DB9 RS-232 Signal Names	
#1 Carrier Detector (DCD)	CD		#1 Carrier Detector (DCD)	CD
#2 Receive Data (Rx)	RD		#2 Receive Data (Rx)	RD
#3 Transmit Data (Tx)	TD		#3 Transmit Data (Tx)	TD
#4 Data Terminal Ready	DTR		#4 Data Terminal Ready	DTR
#5 Signal Ground/Common (SG)	GND		#5 Signal Ground/Common (SG)	GND
#6 Data Set Ready	DSR		#6 Data Set Ready	DSR
#7 Request to Send	RTS		#7 Request to Send	RTS
#8 Clear to Send	CTS		#8 Clear to Send	CTS
#9 Ring Indicator	RI		#9 Ring Indicator	RI
Soldered to DB9 Metal - Shield	FGND		Soldered to DB9 Metal - Shield	FGND



Note: Signal directions reversed if devices are DTE to DTE - "Null Modem" cable for DTE devices also connects pins #1 & #6 on each side to simulate Carrier (CD) which is required by some Terminal program software.

Null Modem Cable - Crossover Cable DB9 to DB9

DTE Device (Computer) **DB9** — **DTE to DTE** Connections — DTE Device (Computer) **DB9**

Pin# DB9 RS-232 Signal Names		Signal Direction	Pin# DB9 RS-232 Signal Names	
#1 Carrier Detector (DCD)	CD		#1 Carrier Detector (DCD)	CD
#2 Receive Data (Rx)	RD		#2 Receive Data (Rx)	RD
#3 Transmit Data (Tx)	TD		#3 Transmit Data (Tx)	TD
#4 Data Terminal Ready	DTR		#4 Data Terminal Ready	DTR
#5 Signal Ground/Common (SG)	GND		#5 Signal Ground/Common (SG)	GND
#6 Data Set Ready	DSR		#6 Data Set Ready	DSR
#7 Request to Send	RTS		#7 Request to Send	RTS
#8 Clear to Send	CTS		#8 Clear to Send	CTS
#9 Ring Indicator	RI		#9 Ring Indicator	RI
Soldered to DB9 Metal - Shield	FGND		Soldered to DB9 Metal - Shield	FGND

Identifying DTE and DCE Type Connections:

What devices have **DTE** type RS-232 ports?

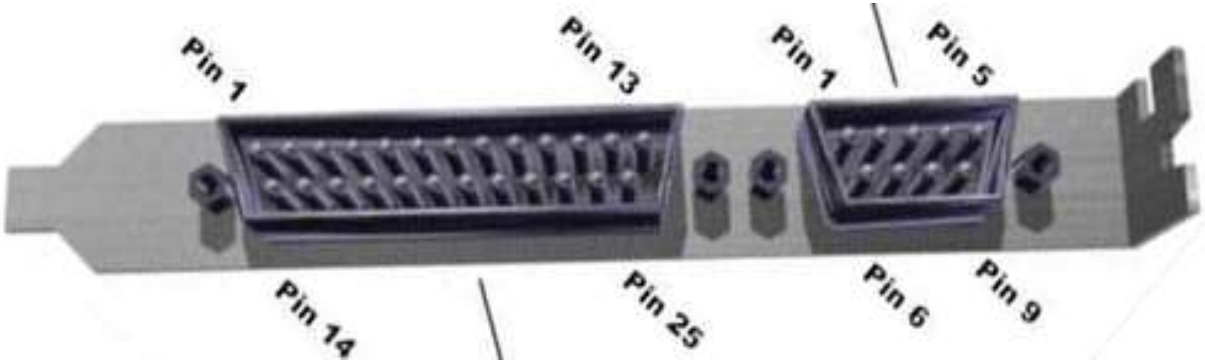
A DTE device is "Data Terminal Equipment", this includes Computers, Serial Printers, PLC's, Video Cameras, Video Recorders, Video Editors, and most devices which are not used to extend communications. Think **COMPUTER** for **DTE**.

What devices have DCE type RS-232 ports?

A DCE device is "Data Communications Equipment", this includes devices intended to plug directly into a DTE port, PDA cables, Modems and devices that extend communications like a modem, such as RS-422, RS-485, or Fiber Optic converters or Radio Modems. Think **MODEM** for **DCE**.

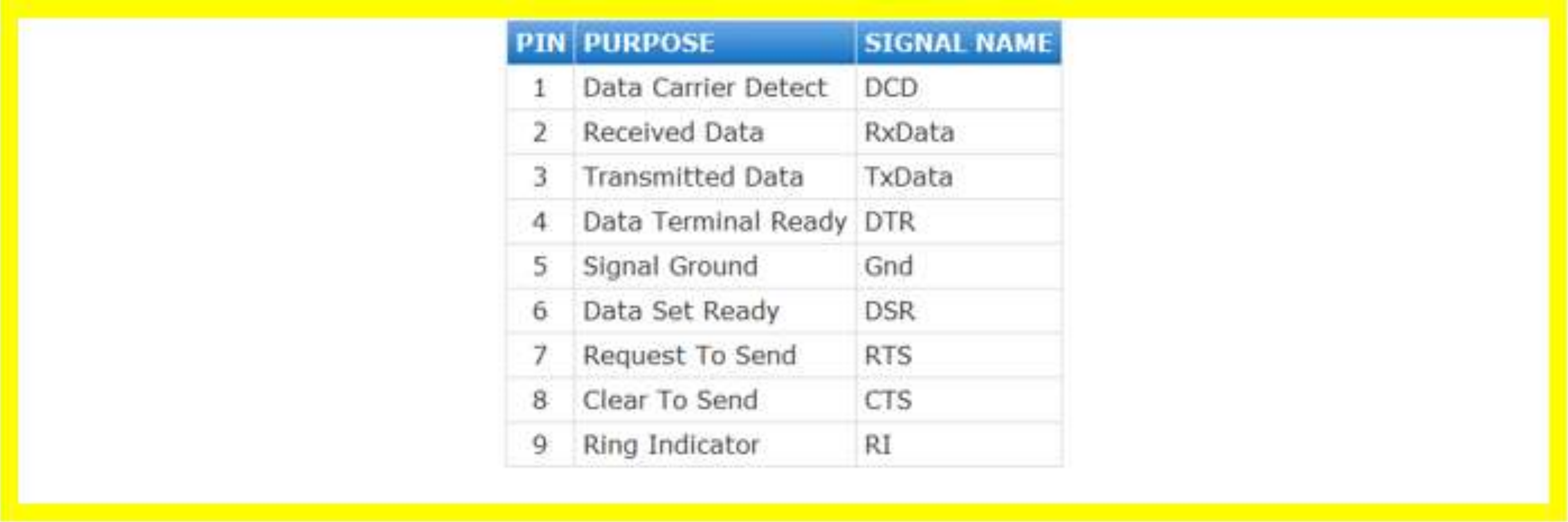
Serial ports are typically identified on IBM compatible computers as COM (communications) ports. For example, a mouse might be connected to COM1 and a modem to COM2. With the introduction of USB**,** FireWire, and other faster solutions serial ports are rarely used when compared to how often they've been used in the past. In the picture to the right is a close up of a **DB9** serial port on the back of a computer.

In the above graphic of a serial port you can notice the DB9 serial port connection is easy to identify. The connection is in the shape of the letter D, is a male connector, and has 9 pins. Serial port connectors are either of 9 pins or 25 pins as shown.



The 9 pin connection information

Below is a listing of each of the pins located on the DB9 connector and what each of these pins are for. As can be seen in the above picture pin one is in the top left and pin 9 is in the bottom right.



PIN	PURPOSE	SIGNAL NAME
1	Data Carrier Detect	DCD
2	Received Data	RxData
3	Transmitted Data	TxData
4	Data Terminal Ready	DTR
5	Signal Ground	Gnd
6	Data Set Ready	DSR
7	Request To Send	RTS
8	Clear To Send	CTS
9	Ring Indicator	RI

Examples for what the serial port is used for:

Below is a listing of various hardware components that can be purchased and used with your serial port.

Mouse - One of the most commonly used devices for serial ports, usually used with computers with no PS/2 or USB ports and specialty mice.

Modem - Another commonly used device for serial ports. Used commonly with older computers, however, is also commonly used for its ease of use.

Network - One of the original uses of the serial port, which allowed two computers to connect together and allow large files to be transferred between the two.

Printer - Today, this is not a commonly used device for serial ports. However, was frequently used with older printers and plotters.

Considered to be one of the most basic external connections to a computer, the **serial port** has been an integral part of most computers for more than 20 years. Although many of the newer systems have done away with the serial port completely in favor of USB connections, most modems still use the serial port, as do some printers, PDAs and digital cameras. Few computers have more than two serial ports.

Essentially, serial ports provide a standard connector and protocol to let you attach devices, such as modems, to your computer.

Notes

All computer operating systems in use today support serial ports, because serial ports have been around for decades. Parallel ports are a more recent invention and are much faster than serial ports. USB ports are only a few years old, and will likely replace both serial and parallel ports completely over the next several years.

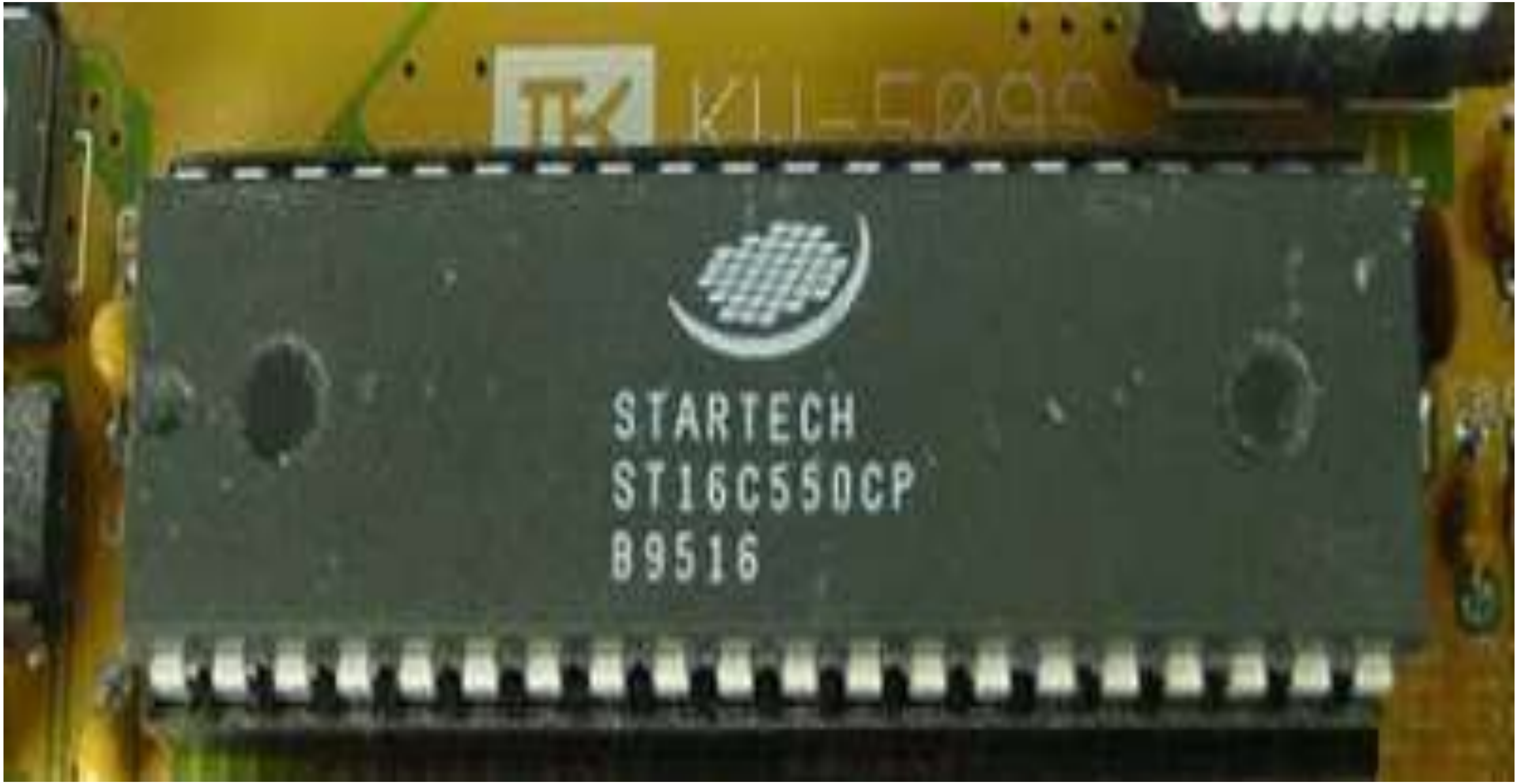
The name "serial" comes from the fact that a serial port "serializes" data. That is, it takes a byte of data and transmits the 8 bits in the byte one at a time. The advantage is that a serial port needs only one wire to transmit the 8 bits (while a parallel port needs 8). The disadvantage is that it takes 8 times longer to transmit the data than it would if there were 8 wires. Serial ports lower cable costs and make cables smaller.

Before each byte of data, a serial port sends a start bit, which is a single bit with a value of 0. After each byte of data, it sends a stop bit to signal that the byte is complete. It may also send a parity bit.

Serial ports, also called **communication (COM) ports**, are **bi-directional**. Bi-directional communication allows each device to receive data as well as transmit it. Serial devices use different pins to receive and transmit data -- using the same pins would limit communication to **half-duplex**, meaning that information could only travel in one direction at a time. Using different pins allows for **full-duplex** communication, in which information can travel in both directions at once.

Serial ports rely on a special controller chip, the **Universal Asynchronous Receiver/Transmitter (UART)**, to function properly. The UART chip takes the parallel output of the computer's system bus and transforms it into serial form for transmission through the serial port.

In order to function faster, most UART chips have a built-in buffer of anywhere from 16 to 64 kilobytes. This buffer allows the chip to cache data coming in from the system bus while it is processing data going out to the serial port. While most standard serial ports have a maximum transfer rate of 115 Kbps (kilobits per second), high speed serial ports, such as **Enhanced Serial Port (ESP)** and **Super Enhanced Serial Port (Super ESP)**, can reach data transfer rates of 460 Kbps


STARTECH
ST16C550CP
B9516

The Serial Connection Pins Purpose (Role):

The external connector for a serial port can be either 9 pins or 25 pins. Originally, the primary use of a serial port was to connect a modem to your computer. The pin assignments reflect that. Below is an explanation for the purpose or role of each pin when a modem is connected.

9-pin connector:

Carrier Detect - Determines if the modem is connected to a working phone line.

Receive Data - Computer receives information sent from the modem.

Transmit Data - Computer sends information to the modem.

Data Terminal Ready (DTR)- Computer tells the modem that it is ready to talk.

Signal Ground - Pin is grounded.

Data Set Ready (DSR) - Modem tells the computer that it is ready to talk.

Request To Send (RTS) - Computer asks the modem if it can send information.

Clear To Send (CTS) - Modem tells the computer that it can send information.

Ring Indicator - Once a call has been placed, computer acknowledges signal (sent from modem) that a ring is detected.

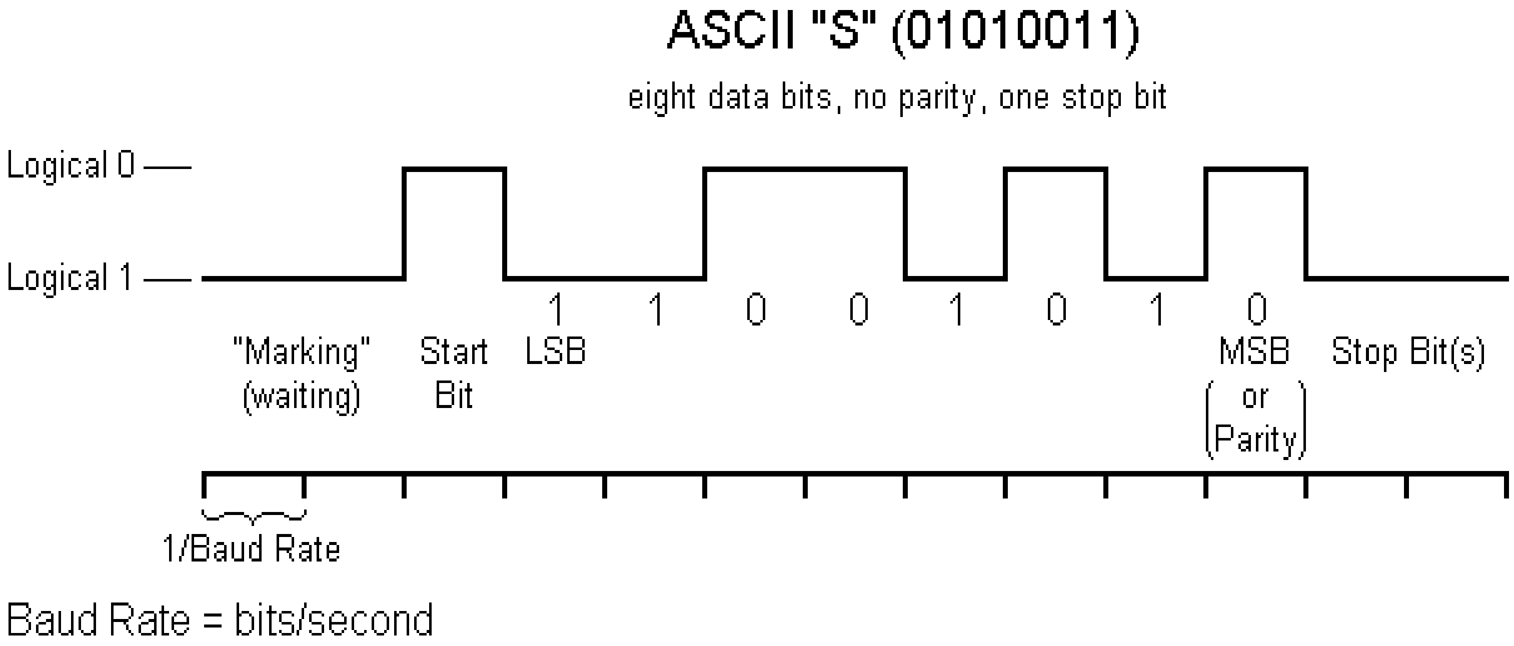
Flow Control:

An important aspect of serial communications is the concept of **flow control**. This is the ability of one device to tell another device to stop sending data for a while. The commands Request to Send (RTS), Clear To Send (CTS), Data Terminal Ready (DTR) and Data Set Ready (DSR) are used to enable flow control.

Let's look at an example of how flow control works: You have a modem that communicates at 56 Kbps. The serial connection between your computer and your modem transmits at 115 Kbps, which is over twice as fast. This means that the modem is getting more data coming from the computer than it can transmit over the phone line. Even if the modem has a 128K buffer to store data in, it will still quickly run out of buffer space and be unable to function properly with all that data streaming in.

With flow control, the modem can stop the flow of data from the computer before it overruns the modem's buffer. The computer is constantly sending a signal on the Request to Send pin, and checking for a signal on the Clear to Send pin. If there is no Clear to Send response, the computer stops sending data, waiting for the Clear to Send before it resumes. This allows the modem to keep the flow of data running smoothly.

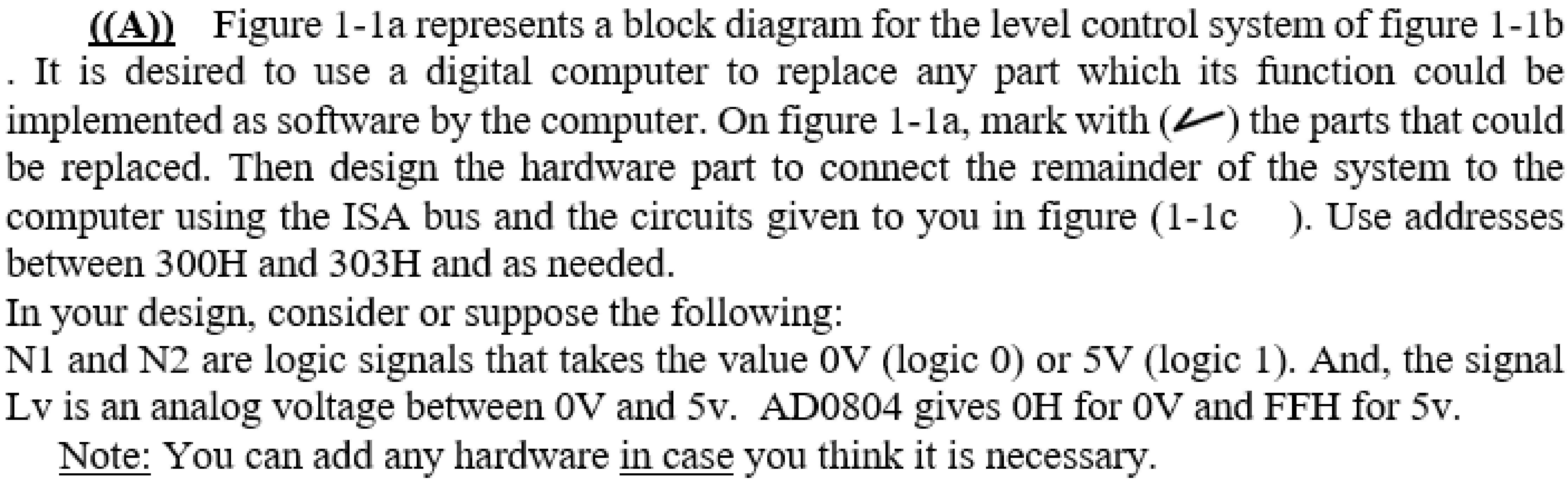
A sample Timing Diagram of Asynchronous Signal



Voltage sent over the pins can be in one of two states, **On** or **Off**. On (binary value "1") means that the pin is transmitting a signal between -3 and -25 volts, while Off (binary value "0") means that it is transmitting a signal between +3 and +25 volts...

The **baud rate** is the rate at which information is transferred in a communication channel. In the serial port context, "9600 baud" means that the serial port is capable of transferring a maximum of 9600 bits per second. There are different baud rates that can be used depending on the devices capabilities.

Lecture 14 - Case Study 2



((A)) Figure 1-1a represents a block diagram for the level control system of figure 1-1b . It is desired to use a digital computer to replace any part which its function could be implemented as software by the computer. On figure 1-1a, mark with (✓) the parts that could be replaced. Then design the hardware part to connect the remainder of the system to the computer using the ISA bus and the circuits given to you in figure (1-1c). Use addresses between 300H and 303H and as needed.

In your design, consider or suppose the following:

N1 and N2 are logic signals that takes the value 0V (logic 0) or 5V (logic 1). And, the signal Lv is an analog voltage between 0V and 5v. AD0804 gives 0H for 0V and FFH for 5v.

Note: You can add any hardware in case you think it is necessary.

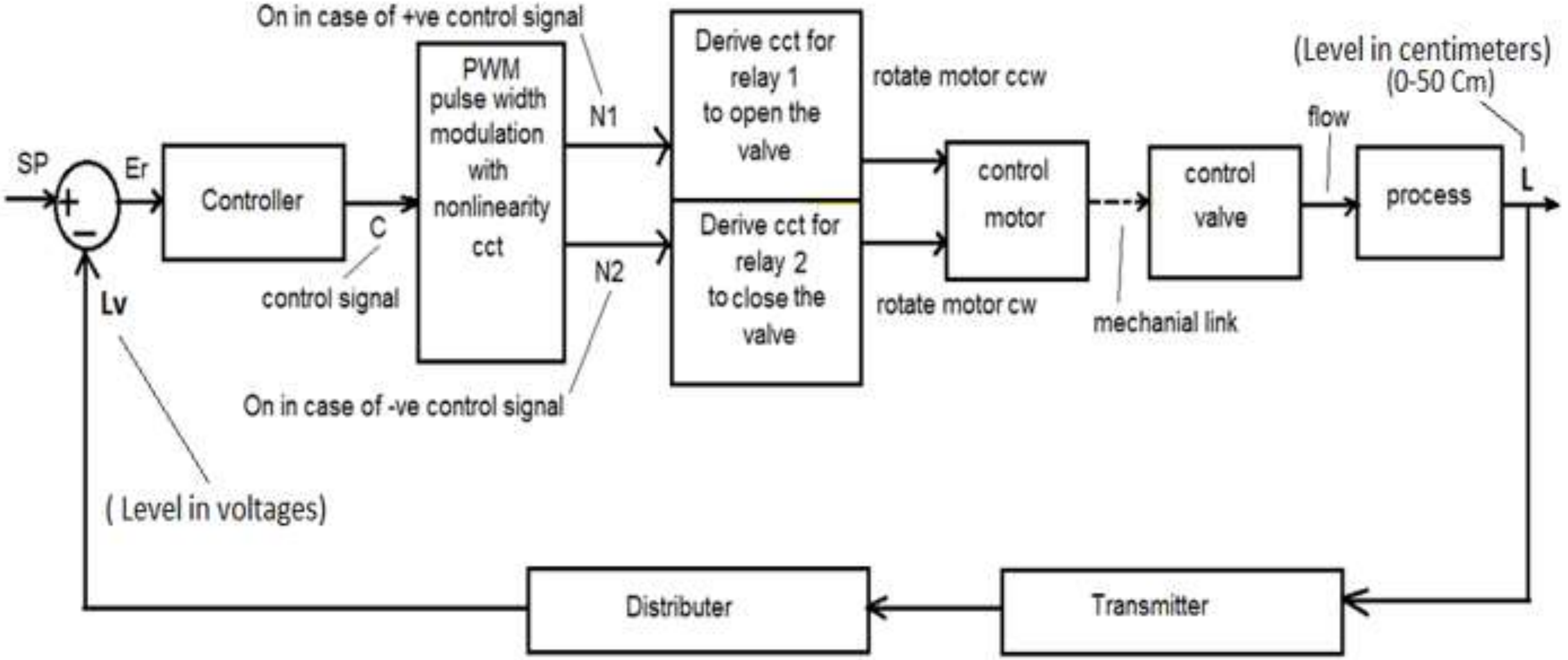

On in case of +ve control signal
PWM
pulse width
modulation
with
nonlinearity
cct
N1
Derive cct for
relay 1
to open the
valve
rotate motor ccw
(Level in centimeters)
(0-50 Cm)
flow
SP
+
–
Er
Controller
C
control signal
N2
Derive cct for
relay 2
to close the
valve
rotate motor cw
control
motor
mechanial link
control
valve
process
L
Lv
On in case of -ve control signal
(Level in voltages)
Distributer
Transmitter

Figure 1-1a

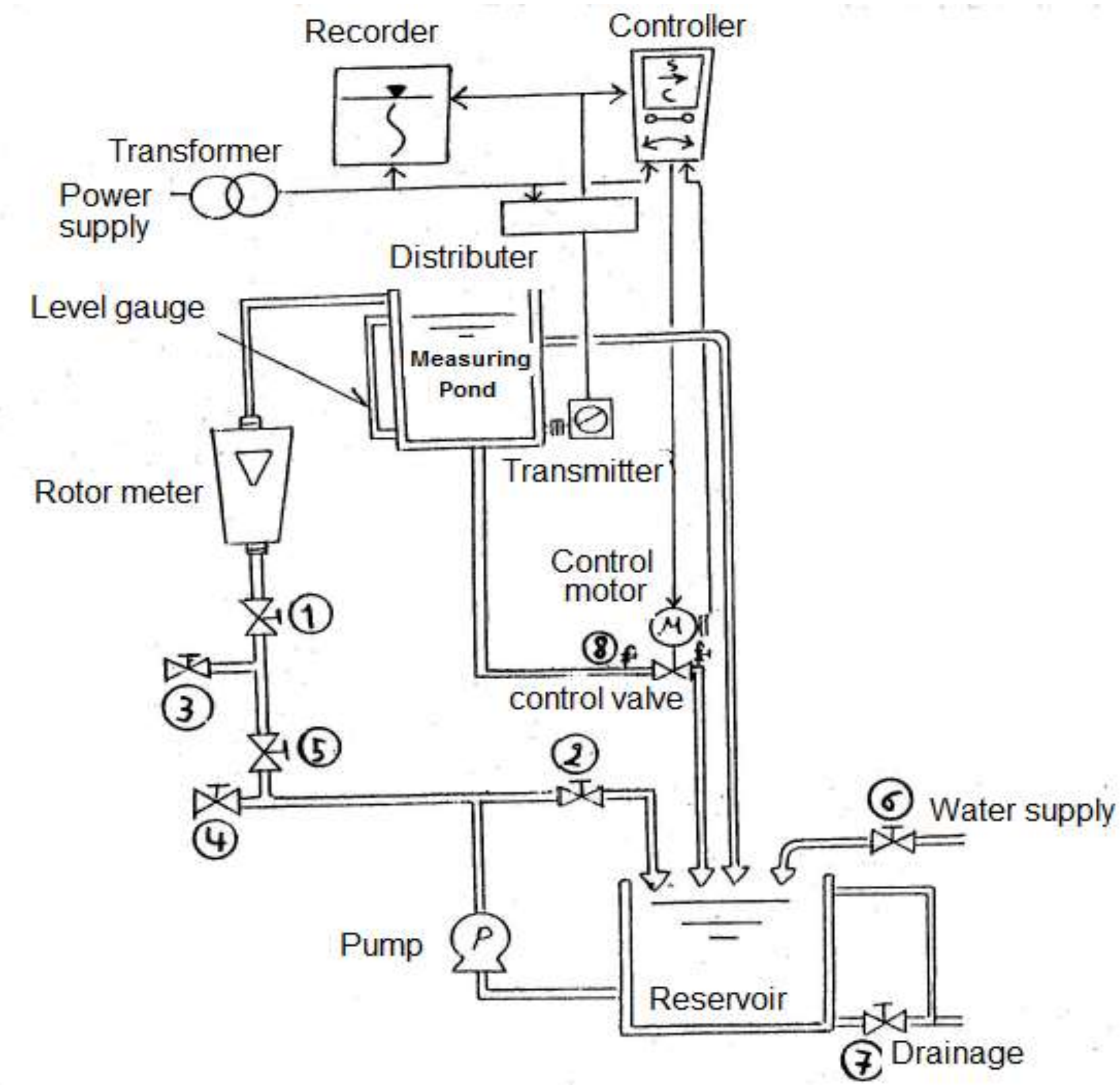

Recorder
Controller
Transformer
Power supply
Distributer
Level gauge
Measuring Pond
Transmitter
Rotor meter
Control motor
M
1
8
control valve
3
5
2
6
Water supply
4
Pump
P
Reservoir
7
Drainage

Figure 1.1.b

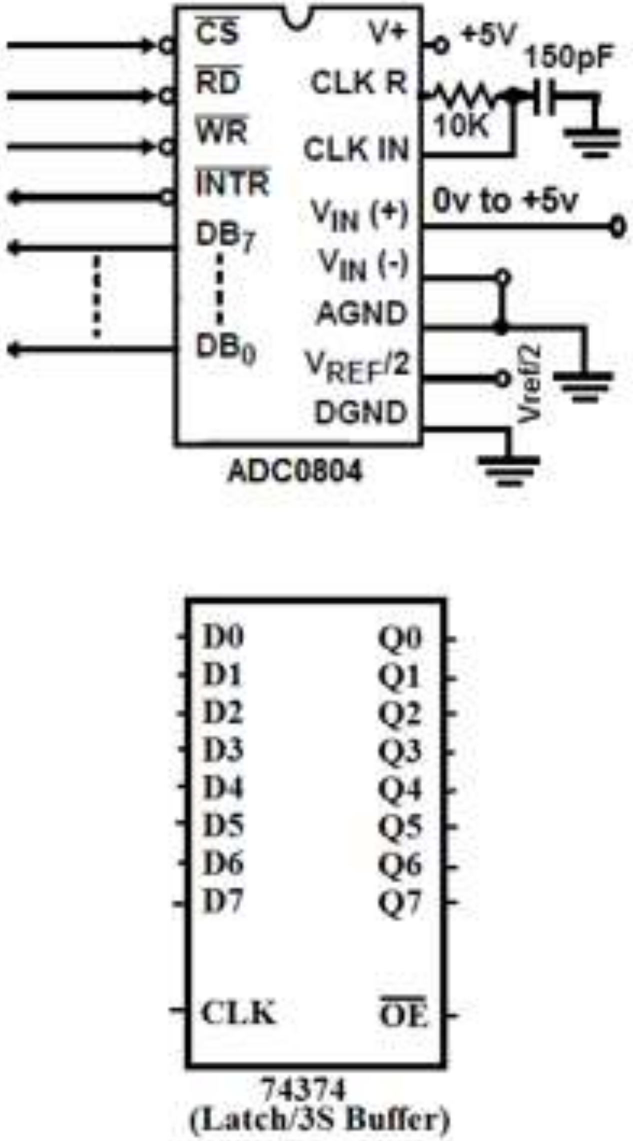
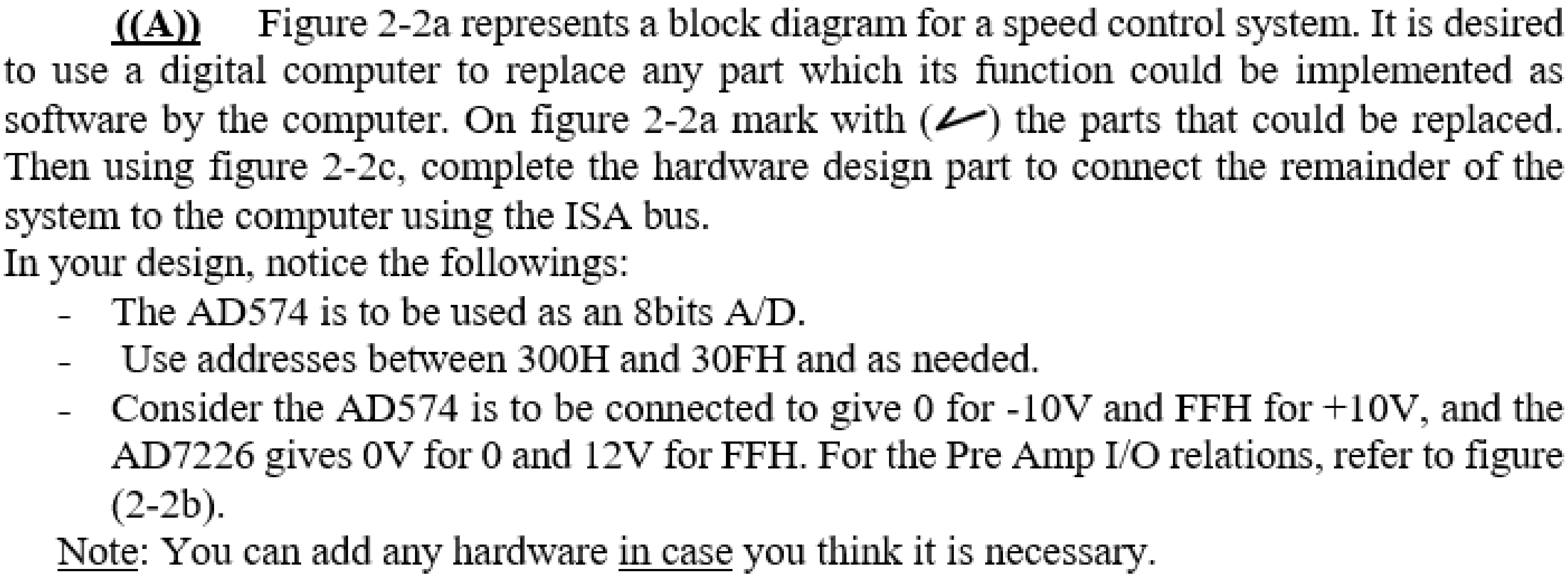

CS
RD
WR
INTR
DB7
DB0
V+
+5V
150pF
CLK R
10K
CLK IN
0v to +5v
VIN (+)
VIN (-)
AGND
VREF/2
Vref/2
DGND
ADC0804
D0
D1
D2
D3
D4
D5
D6
D7
CLK
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
OE
74374
(Latch/3S Buffer)

Figure 1-1c

((B)) Depending on your design in figure (1-1c) , write an instruction (or instructions) to make N1=0 and N2=1

((C)) Depending on your design in figure (1-1c) , write a complete program to run the system.

Lecture 15 - Case Study 3



((A)) Figure 2-2a represents a block diagram for a speed control system. It is desired to use a digital computer to replace any part which its function could be implemented as software by the computer. On figure 2-2a mark with (✓) the parts that could be replaced. Then using figure 2-2c, complete the hardware design part to connect the remainder of the system to the computer using the ISA bus.

In your design, notice the followings:

- The AD574 is to be used as an 8bits A/D.
- Use addresses between 300H and 30FH and as needed.
- Consider the AD574 is to be connected to give 0 for -10V and FFH for +10V, and the AD7226 gives 0V for 0 and 12V for FFH. For the Pre Amp I/O relations, refer to figure (2-2b).

Note: You can add any hardware in case you think it is necessary.

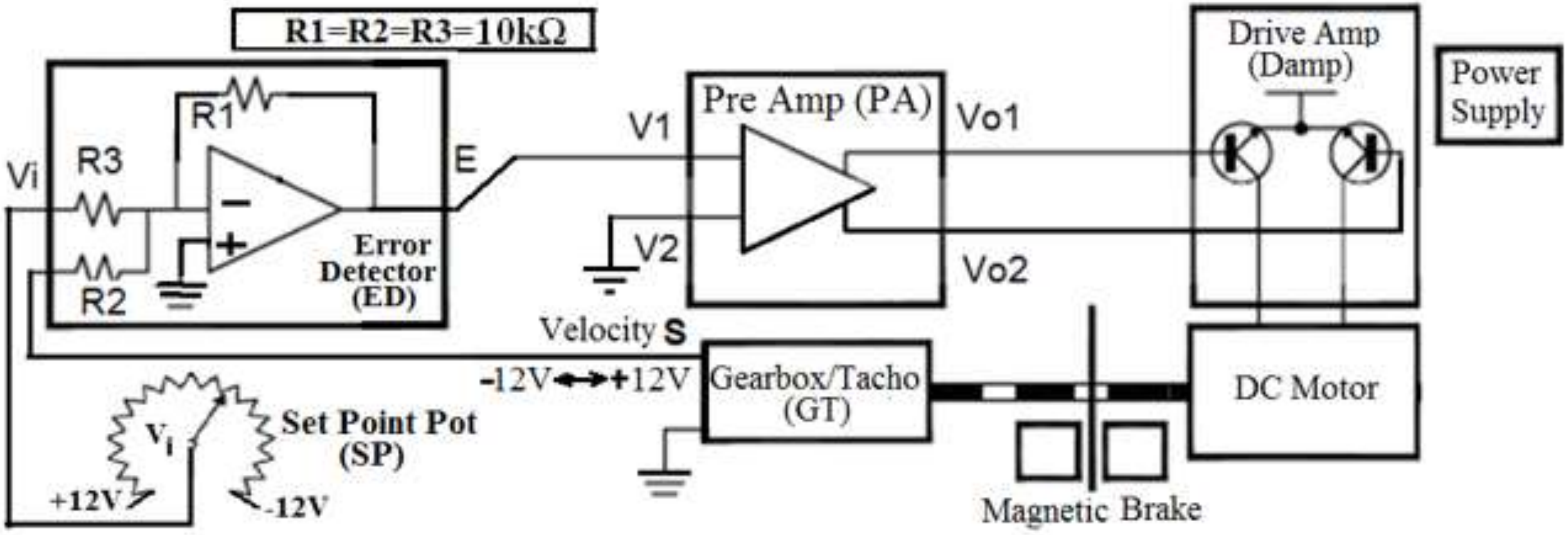
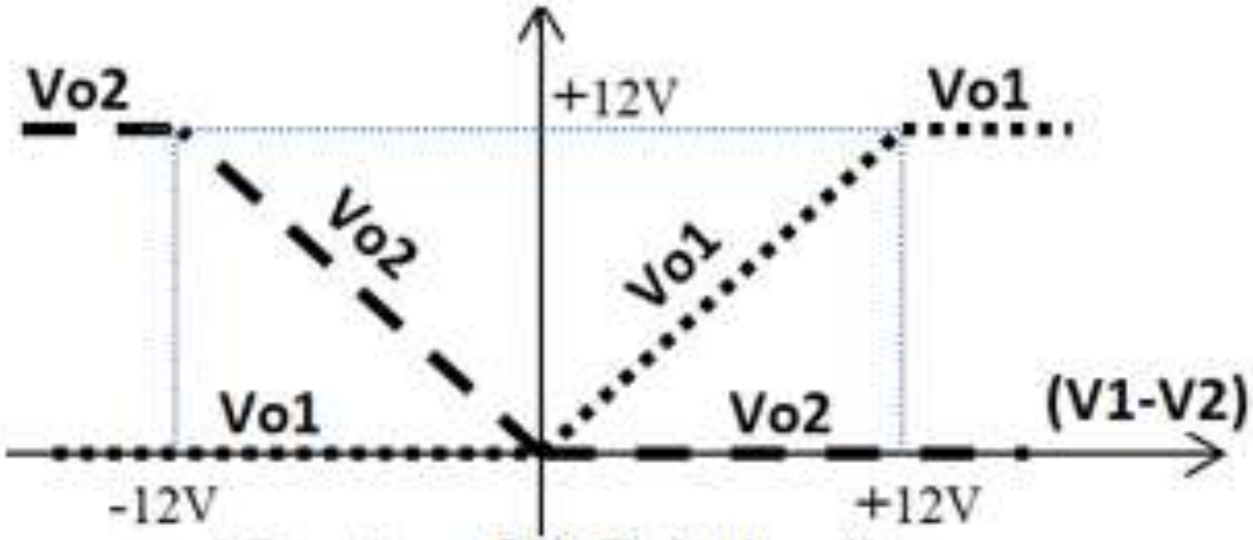

R1=R2=R3=10kΩ
R1
R3
Vi
R2
Error Detector (ED)
E
V1
V2
Pre Amp (PA)
Vo1
Vo2
Drive Amp (Damp)
Power Supply
Velocity S
-12V↔+12V
Gearbox/Tacho (GT)
DC Motor
Magnetic Brake
Set Point Pot (SP)
V_i
+12V
-12V

Figure 2-2a



((Pre Amp I/O Relations))

Figure 2-2b

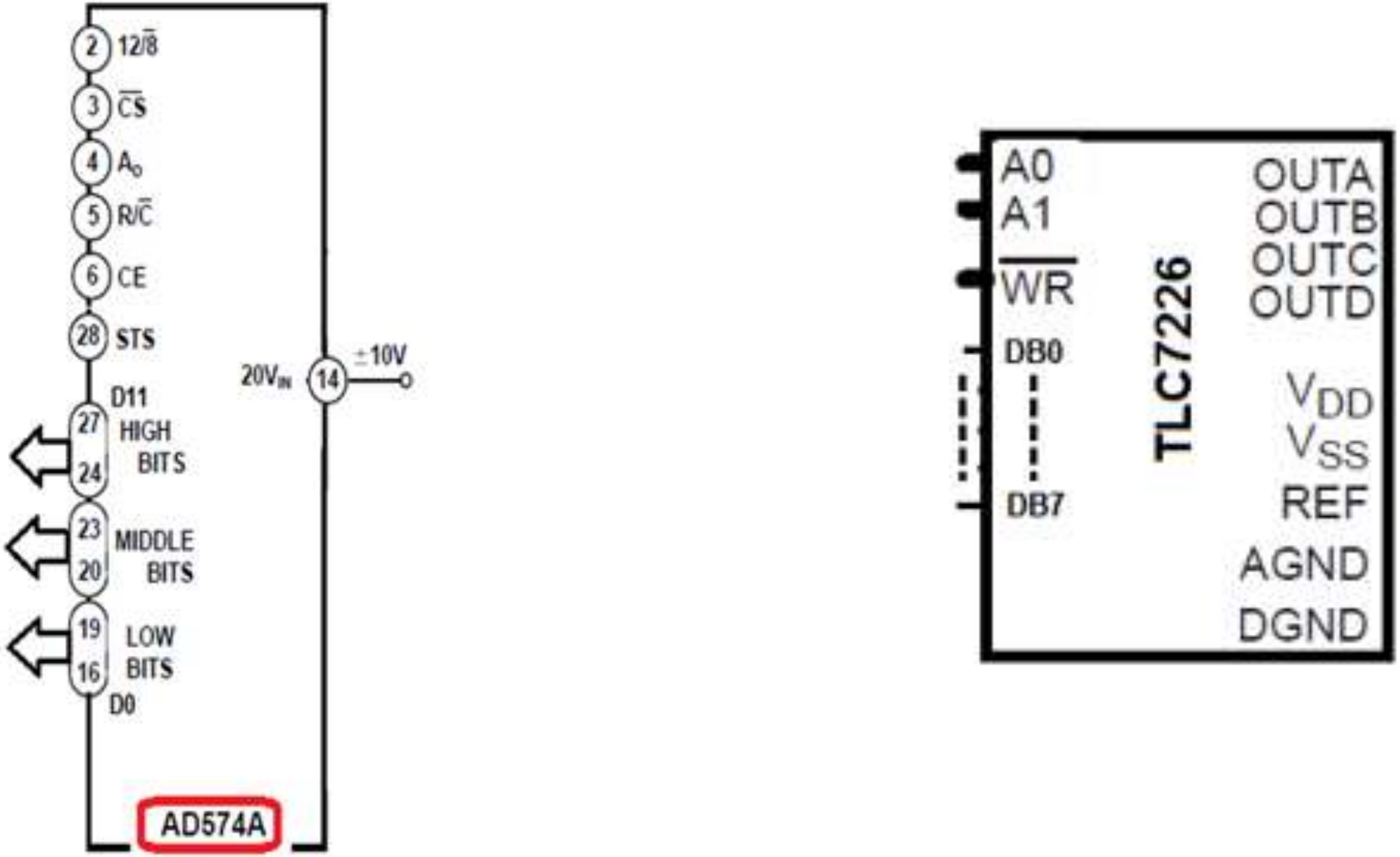

2
12/$\overline{8}$
3
$\overline{CS}$
4
A_0
5
R/$\overline{C}$
6
CE
28
STS
20V_{IN}
14
±10V
D11
27
HIGH
BITS
24
23
MIDDLE
BITS
20
19
LOW
BITS
16
D0
AD574A
A0
A1
$\overline{WR}$
DB0
DB7
TLC7226
OUTA
OUTB
OUTC
OUTD
V_{DD}
V_{SS}
REF
AGND
DGND

Figure 2-2c

((B)) Depending on your design in figure (2-2c) , write an instruction (or instructions) to send **Vo1d** to the D/A. (Note: **Vo1d** is the digital value for **Vo1).**

((C)) Depending on your design in figure (2-2c) , write a complete program to run the system.