Name : Abhishek Srivastava.
Reg. No. : 19BCE10071
Subject : Embedded Systems
Date : 11.03.22
Slot : E11 + E12 + E13

MIDTERM EXAMINATION

Q1]

For selecting a right microprocess controller for an Embedded System can be daunting. Nent only there are several technical features to consider, but there are also business case issues such as cost and lead-times that can cripple the product.

Before choosing a microcontroller, we should workout the high levels of the system. Block diagram and flowchart them only then there is enough information to start making rational decisions on selection.

Here are a few steps that can be followed while choosing a micro-controller.

Step-1 : Make list of required hardware interfaces.

Make a generic example of the block diagram with the I/O requirements.

Step 2 : Examine the software architecture.

The software architecture can greatly affect the system. How heavy or how light the processing requirements will determine whether you go with an 80 MHz or 8 MHz 8051. Make note of any other requirement or optimisations as well. Determine the running time, processing power required and the computing power.

Step 3 : Select the architecture.

Using the Step 1 and Step 2, we will get some idea about architecture. It can be 8-bit, or 16-bit or even 32 bit ARM. These points start to converge on a solution.

Microcontroller selection can be an iterative process. It can change if other fits better.

Step 4 : Identify the RAM required.

Making sure we don't run out of space is our primary priority. We should always start with more and then move to less constrained part within same family. We can estimate the required RAM using the software architecture and the communication peripherals.

Step 5 : Start searching for microcontrollers.

A good place to start is microcontroller suplier. Most silicon providers have a search engine that allowes you to enter your peripheral sets, I/O and power requirements and it will narrow down a list of part that match the criteria.

Step 6 : Examine Cost and power Constraints.

If the device is powered from a battery and mobile, then make sure the parts are low powered else keep weeding the list down until you select a few

Step 7 : Check availability.

Check availability of parts, if its durable for our product life cycle.

Step-8 : Select a development Kit

We should research on development kits availiable. If it's not then the selected parts is most likely not a good choice and they should go back a few step to find a better one.

Step 9 : Investigate Compiler and tool.

Most microcontroller have a choice for a number of compiler, code and ~~degugging~~ debugging tools. Make sure that all the necessary tools are available.

Step 10: Start Experimenting

Take advantage by buiding up test circuits and interfacing them to the microcontroller. Choose high risk part and get them working. In any event, early experimentation will ensure that you have made the right choice.

Q2] Language processing Systems.

- The hardware understands a language, which humans cannot understand.
- We write programs in high-level language, which is easier for us to understand and remember.
- These programs are then fed into a series of tools and os components to get the desired code that can be used by the machine.

This is known as Language Processing System.

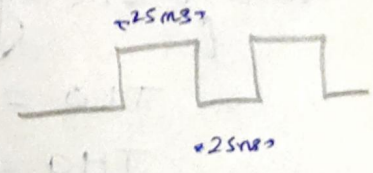
A1
Q3]

Given:

Port No : 2
Pin No. : 2
frequency = 20 Hz
Crystal Frequency = 12 MHz

{1 instruc^n cycle = 12 clock cycle}.

$$\text{Time} = \frac{1}{\text{Frequency}} = \frac{1}{20} = 50\,ms.$$



Time = 50 ms
= 25 ms (ON) + 25ms (OFF)

Delay of 25 ms

Timer 0 Mode 1 :

TMOD = Set timer / counter + mode.
TCON = TR0, TF0
TR0 = Start timer
TF0 = Timer Overflow

Delay Calculations-

XTAL = 12 MHz

Time for 1 instruction : $\frac{12}{12} = 1\,\mu s$

Delay for 25ms = ~~1us~~ 25/1 = 25000

Count = 65536 - 25000 = 40536 = 9E58 (HEX)

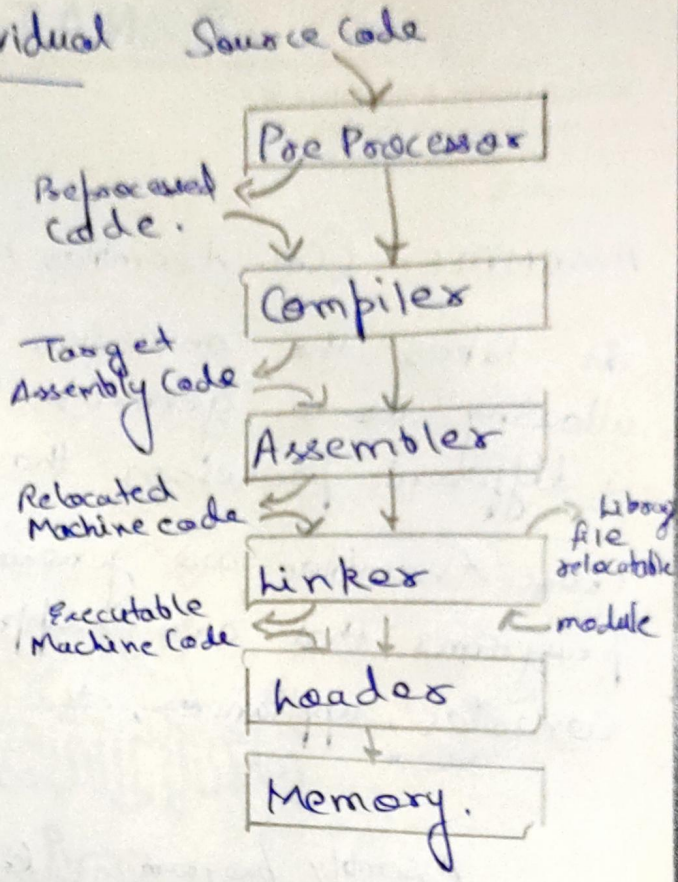
TH0 = 9E

TL0 = 58

Code

```
# include <reg51.h>
sbit mybit = P2^2;
void MSDelay ()
  {
      TL0 = 58;
      TH0 = 9E;
      TR0 = 1;
      while (TF0 == 0);
      TR0 = 0;
      TF0 = 0;
  }
void main () {
    TMOD = 0x01;
    while (1) {
      mybit = ~mybit;
      MSDelay ();
    }
}
```

Disscussing About the Individual Blocks.



COMPILERS

It is a program that translates the code written in one language to another language. Main purpose is to convert a high level language to low level language.

Eg: GCC, Keil, Clang, etc.

LINKER.

It is a program in the System that helps to link an object modules of program into a single object file. It takes object module from assember as input and form an executable output file for loader.

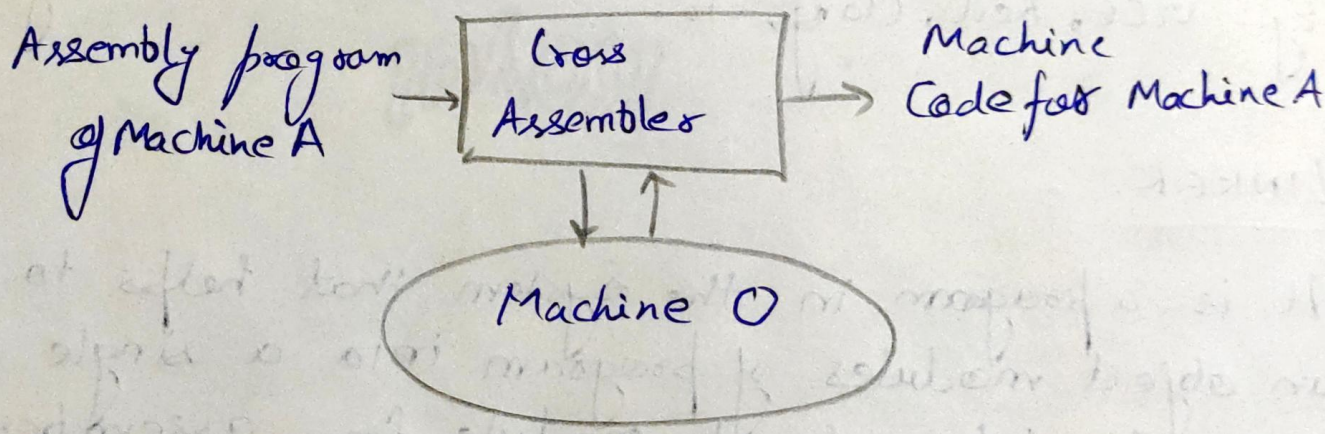
PRE PROCESSOR.

It is the part of compiler that performs preliminary operations to the code before compiler sees it. It changes text of the source code and result is a new source code. All preprocessor derivatives begin from a #Chash Symbol.

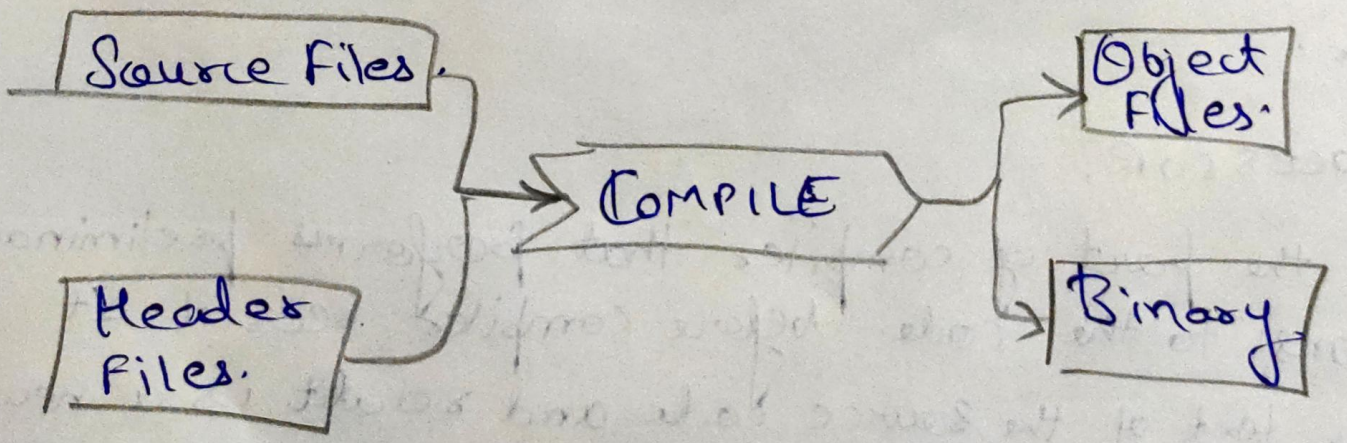
ASSEMBLER (Cross Assembler)

It takes the conversion process a step further by allowing us to generate assembly code for a different processor than the one it compiles on.

Cross Assembler are generally used to develop the programs that are supposed to run on game consoles, appliances, etc.



COMPILATION PROCESS

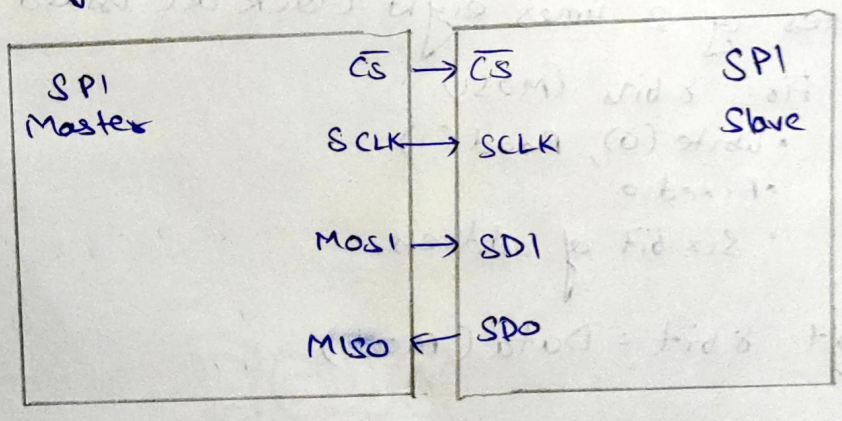


Q5] Serial peripheral Interfacing [SPI]

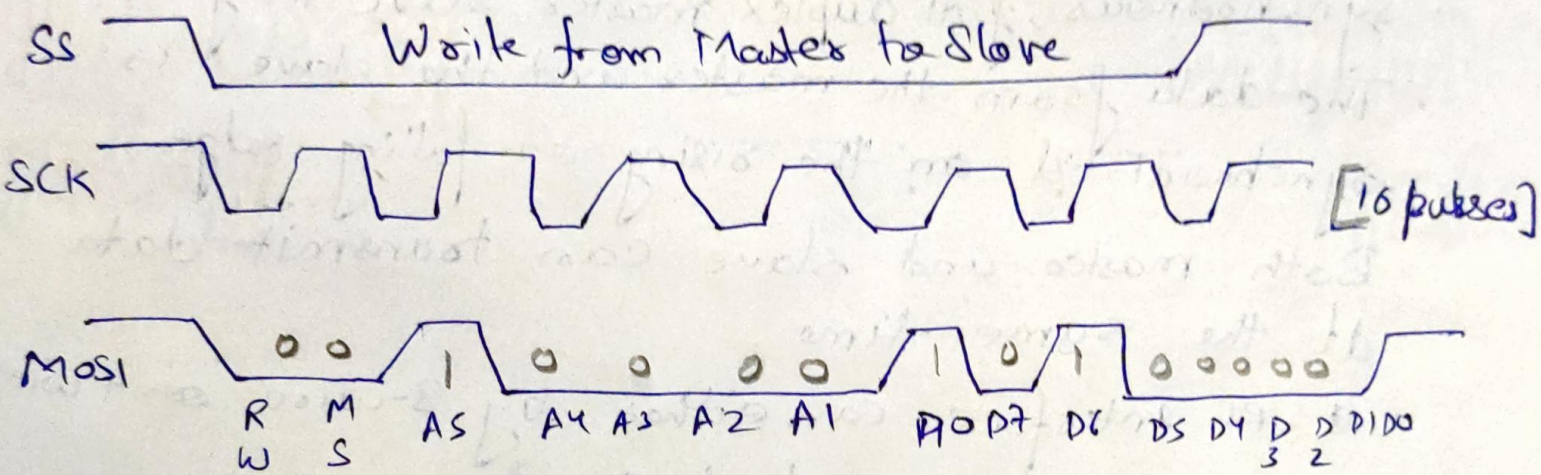
Key features

- One of the most widely used interfaces between microcontroller and peripheral ICs.
- Synchronous, full duplex master slave inteface.
- The data from the master and the slave is synchronized on the rising or falling edge.
- Both master and slave can transmit data at the same time.
- SPI interface can either by 3-wire or 4-wire.
- Master generates clock signal.

Diagram of 4-wire SIP devices

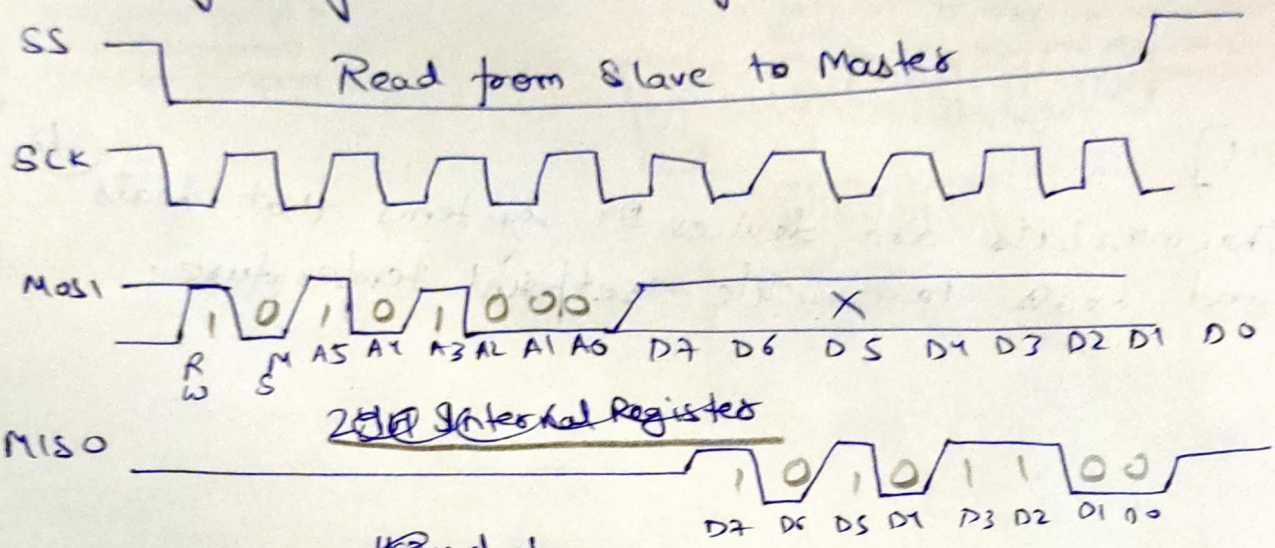


Timing Diagram (Writing Master side).



- Slave select signal must first be forced low by the master.
- Series of 2 times eight clock are issued by master
- The Firs 8 bits (MOSI)
  - Write (0), Read (1)
  - Fixed 0
  - Six bit of address
- Next 8 bit = Data (MOSI)

Timing Diagram (Reading Master Side).

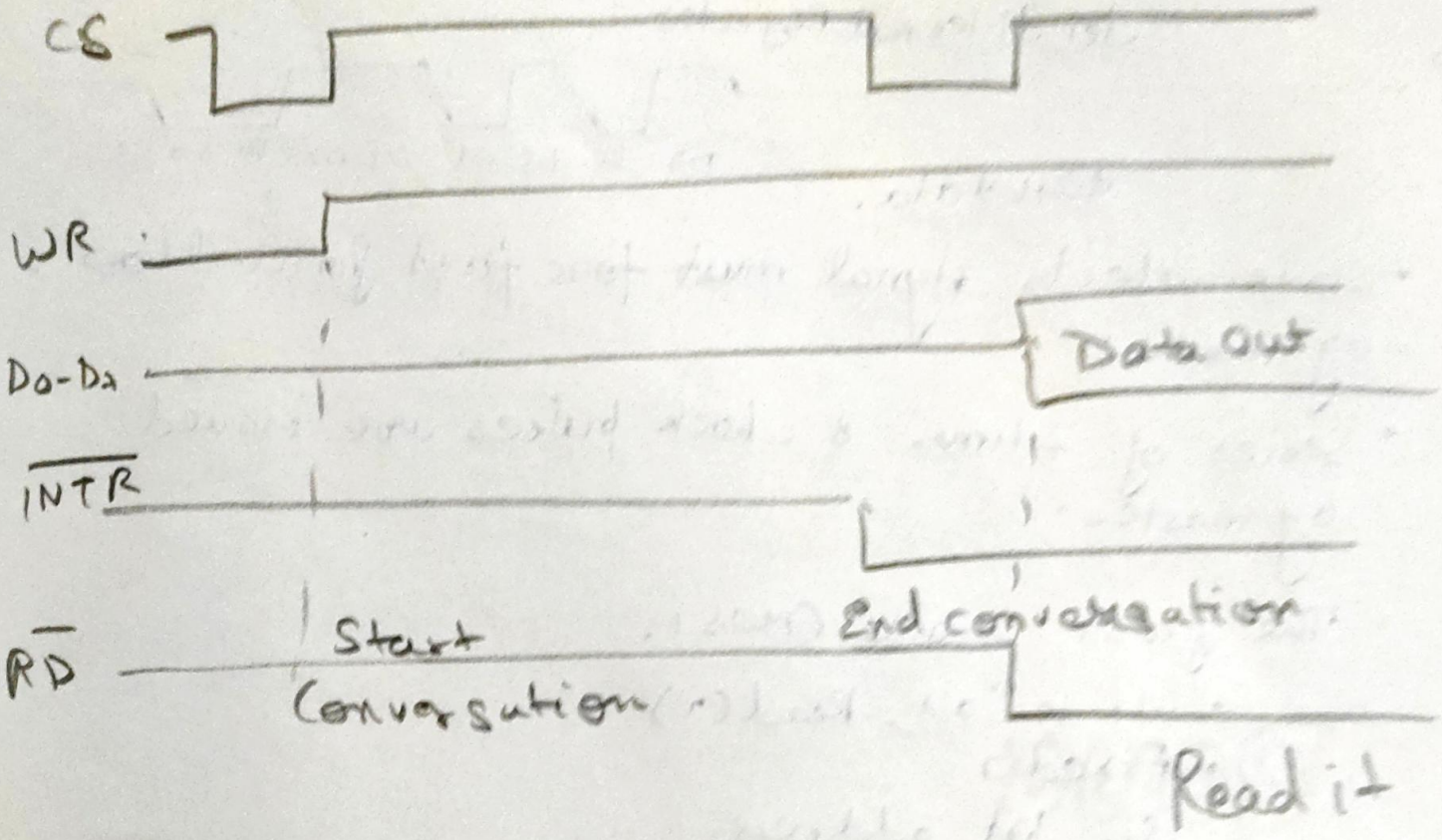


- Slave selects signal must ~~forc~~ first forced low by master.
- Series of 2 times 8 clock pulses are issued by master.
- The first 8 bits (MOSI.
  - Write (0), Read (1)
  - Fixed 0
  - Six bit address.

- Next 8 bits → Data (MOSI)

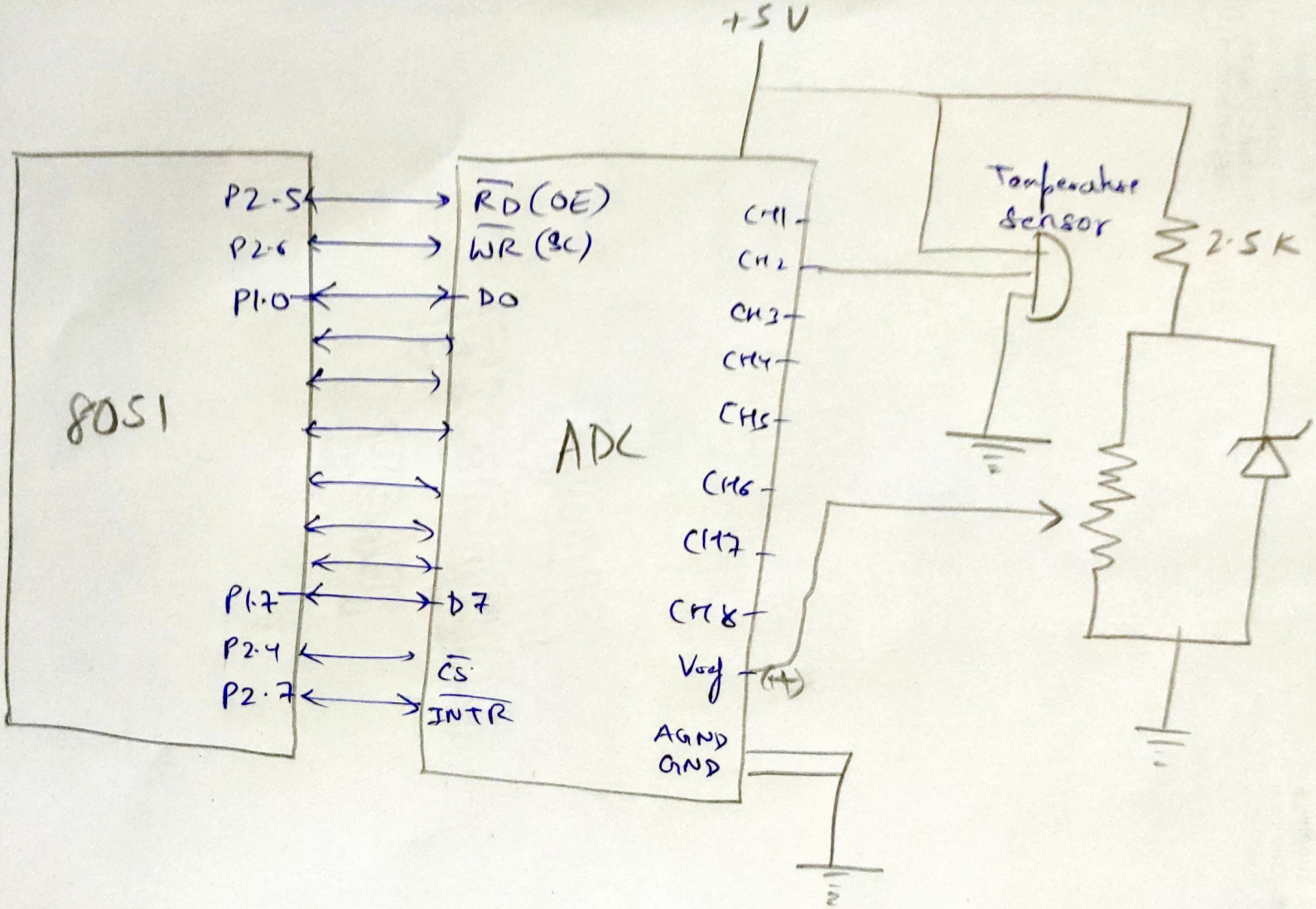
Q4]

Thermostats are devices or systems that heats and cools to a scale setpoint temperature.



Block Diagram

PTO


+5V
8051
P2.5
P2.6
P1.0
P1.7
P2.4
P2.7
RD (OE)
WR (SC)
D0
D7
CS
INTR
ADC
CH1
CH2
CH3
CH4
CH5
CH6
CH7
CH8
Vref
AGND
GND
Temperature sensor
2.5K