CHAPTER 3

METHODOLOGY

3.1 Overview

The Water Monitoring System is consists of circuit design which essentially composed of the sensor section, data conversion and pumping system integrated with assembly language and also using software tools to develop windows as user interface.

3.2 Circuit Design

3.2.1 Level Detector Sensor

A GP2D150A is a digital infrared sensor. It has an infrared transmitter that sends out an invisible beam of light into the environment and an infrared receiver that absorbs the beam of light that is reflected back. The angle of the reflected beam indicates the proximity of the infrared receiver to the object that is reflecting the light.

This sensor can use as object detector because it can sense the object within distance range from 3cm up to 15cm, but it cannot use as distance measurement. Figure 3.1 shows the characteristic of GP2D150A which it only detect any object in the range.

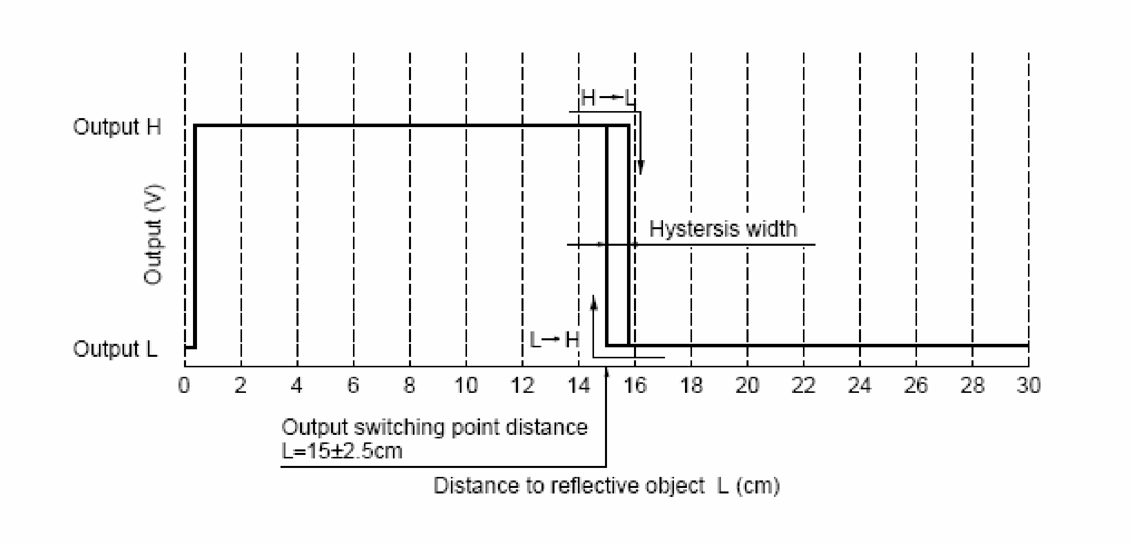


Figure 3.1: Distance versus Voltage for GP2D150A.

Water Monitoring System will state the level of water inside the tank. To set up the differences range of distance with difference value of voltage need a special sensor. Based on that, a GP2D120 distance measuring sensor is found suitable. Figure 3.2 shows the physical of GP2D120 analog sensor.

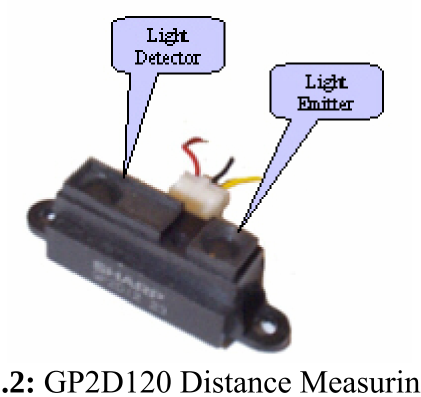


Figure 3.2: GP2D120 Distance Measuring Sensor

A GP2D120 can detect the objects in the range of 4cm to 30cm and verify the distance in term of voltage values as an output. An analog to digital converter need to convert the output signal from the sensor to microcontroller.

3.2.2 Analog To Digital Conversion Circuit

Analog data is generally acquired and converted into digital form for the purpose of processing, transmission, display and storage. Processing may consist of a large variety of operations, ranging from simple comparison to complicated mathematical manipulations. It can be for such purpose as collecting information, converting the data into useful form such as calculation of efficiency of motor speed, using data for controlling a process, generating information for display, and various other purposes. Data may be transmitted over long distances or short distances. The data may be displayed on a digital panel or stored temporarily or permanently.

Figure 3.3 shows an IC ADC0804 which is use to convert analog to digital. The input analog signal is supplied to pin 6 and the digital output signal at pin 11 through pin 18. To make it always enable, pin 1 and pin 7 is always set to ground. An external capacitor 150 pF is connected to pin 4 and resistor 10kΩ to pin 19 of ADC0804 to make its operate.

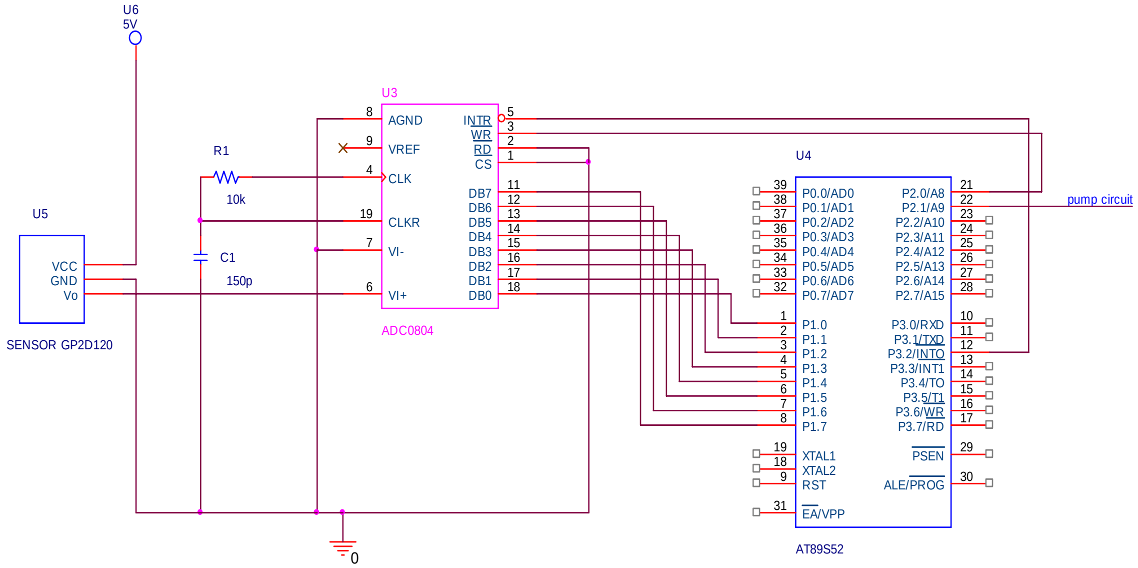


Figure 3.3: The Conversion Circuit Schematic

Port P2.0 and P3.2 of AT89S52 microcontroller are use to control the signal conversion process by using the write $(\overline{WR})$ function and interrupt $(\overline{INTR})$ function of ADC0804, where as pin 1, read $(\overline{RD})$ and pin 2, chip select $(\overline{CS})$ are set to ground. The output data 8 bit from the ADC0804 is connected to Port 1, as an input of the microcontroller.

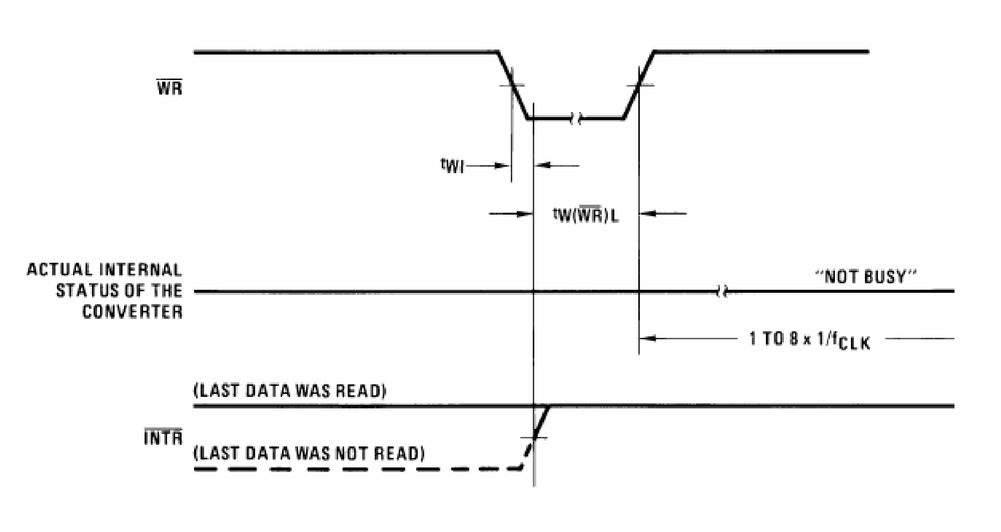
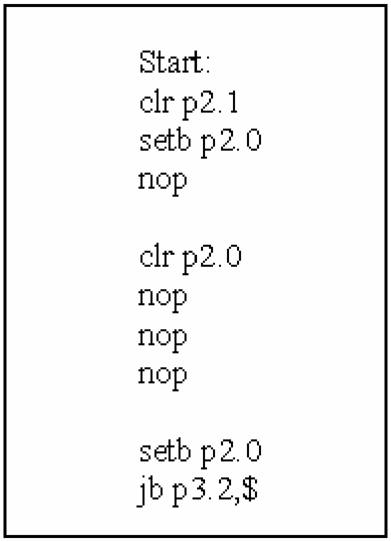


Figure 3.4: Timing Diagram for Conversion Process

Figure 3.4 is the timing diagram of a conversion process triggering which pin 3, $\overline{WR}$ has set to LOW condition for a while and then return it to the HIGH state. The conversion process starts when $\overline{WR}$ goes HIGH (rising edge triggered). When the conversion process is complete, pin 5, $\overline{INTR}$ will go to LOW status and pin 2, $\overline{RD}$ also set to LOW for loading the new value into the ADC0804's outputs D0 – D7. This outputs value will be read by P1.0 of microcontroller. Figure 3.5 shows the source code of assembly language for the conversion process control.



```
Start:
clr p2.1
setb p2.0
nop

clr p2.0
nop
nop
nop

setb p2.0
jb p3.2,$
```

Figure 3.5: Source Code for Conversion Process

3.2.3 Pump Circuit with Transistor Driven Relay

The operation of motor pump will be controlled by microcontroller. The transistor driven relay shown in Figure 3.6 is use to configure the voltage within AC and DC. This relay operates as an electrical switch which opens and closes under control of another electrical circuit. It allows one circuit to switch another circuit.

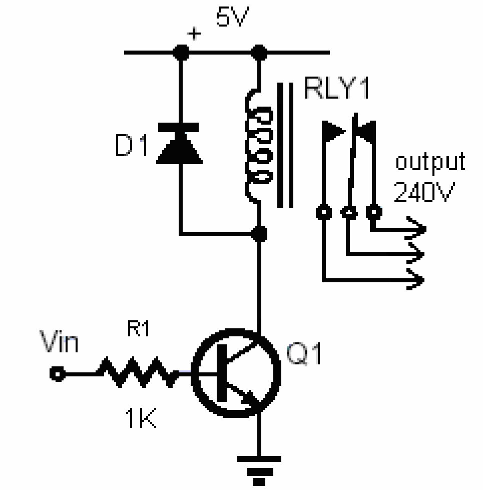


Figure 3.6: Transistor Driven Relay

Figure 3.7 show that a transistor driven relay circuit is used to connect between microcontroller and motor pump. Generally this relay will energize when supply voltage, V_{in} is 5 volt (HIGH) and will OFF when V_{in} is 0 volt (LOW).

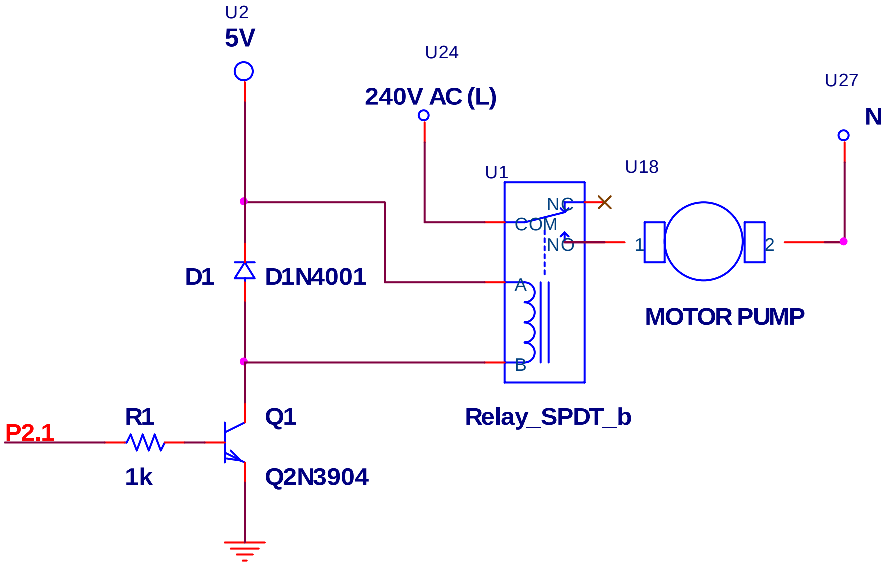


Figure 3.7: Pump Circuit with Transistor Driven Relay

A relay will operate when a suitable 'pull-in' and 'holding' dc current is pass through its energizing coil. An NPN transistor is using to control the coil current. A low-power circuitry is providing enough base current to turn the transistor on and off. An NPN transistor will control the relay with a 5V coil, operating from a +5V supply. Series base resistor R1 is used to set the base current for Q1, so that the transistor is driven into saturation (fully turned on) when the relay is to be energized. A very little power will dissipate when the transistor has a minimal voltage drop. A power diode D1 (1N4001) is connected across the relay coil, to protect the transistor from damage due to the back-EMF pulsed generated in the relay coil's inductance when Q1 turns off.

3.2.4 Water Monitoring System Circuit

Figure 3.9 shows a water monitoring System Circuits. The GP2D120 sensor is connected with a conversion circuit to give signal input to ADC, and then from the conversation circuit it will send the data output signal from ADC to microcontroller. The microcontroller will send that signal to control motor pump through the transistor driven relay circuit. The full circuit connection was design referring to the basic block diagram for this project at Figure 3.8.

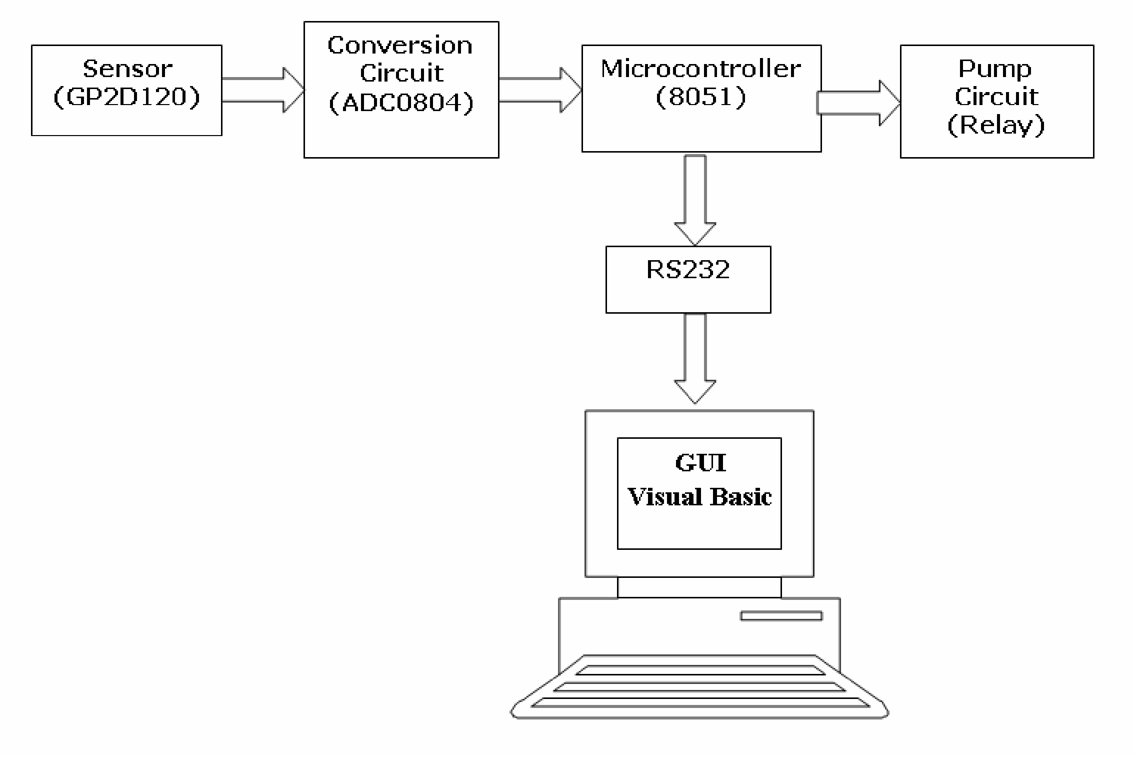


Figure 3.8: Block Diagram for Water Monitoring System

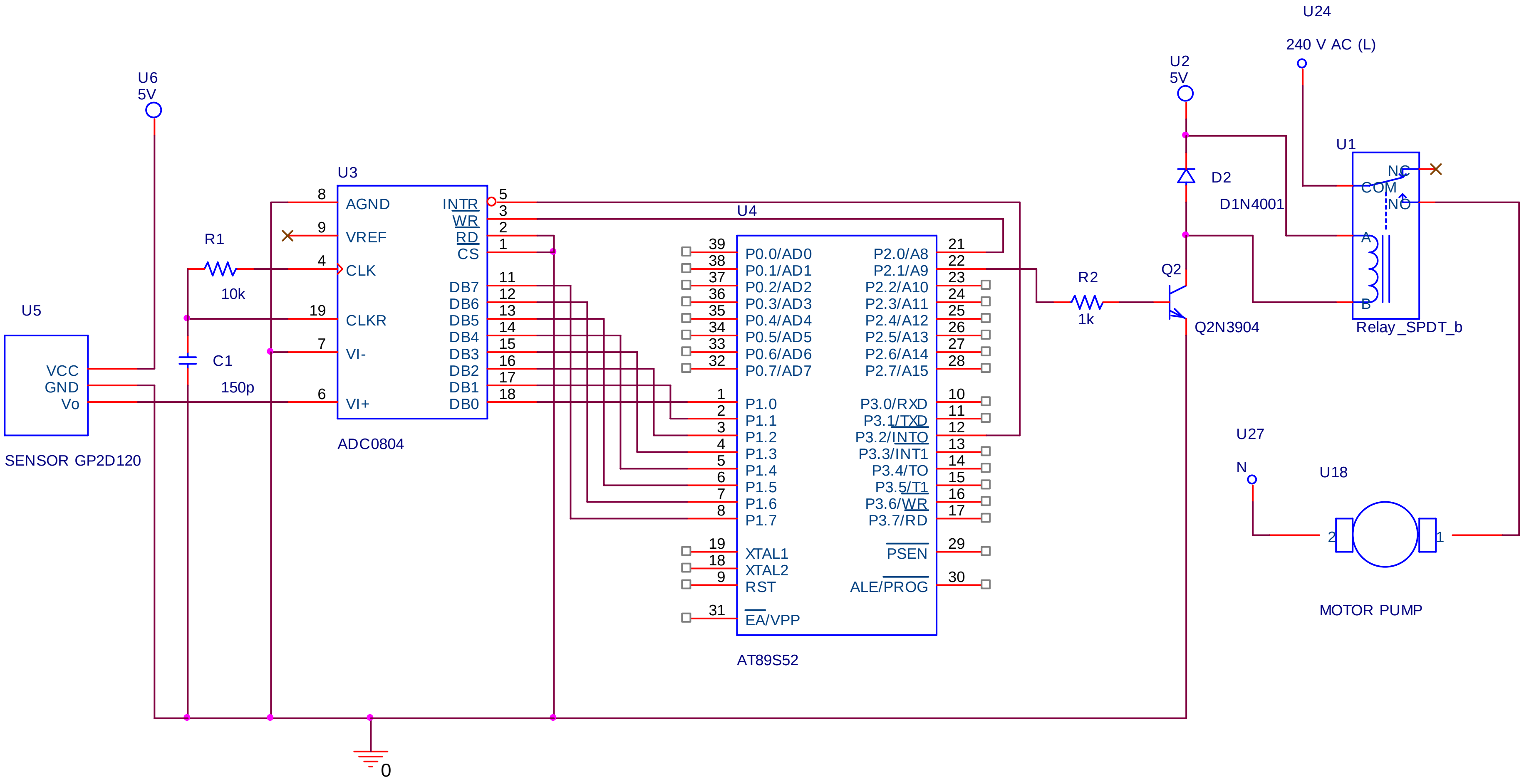


Figure 3.9: Water Monitoring System Circuit

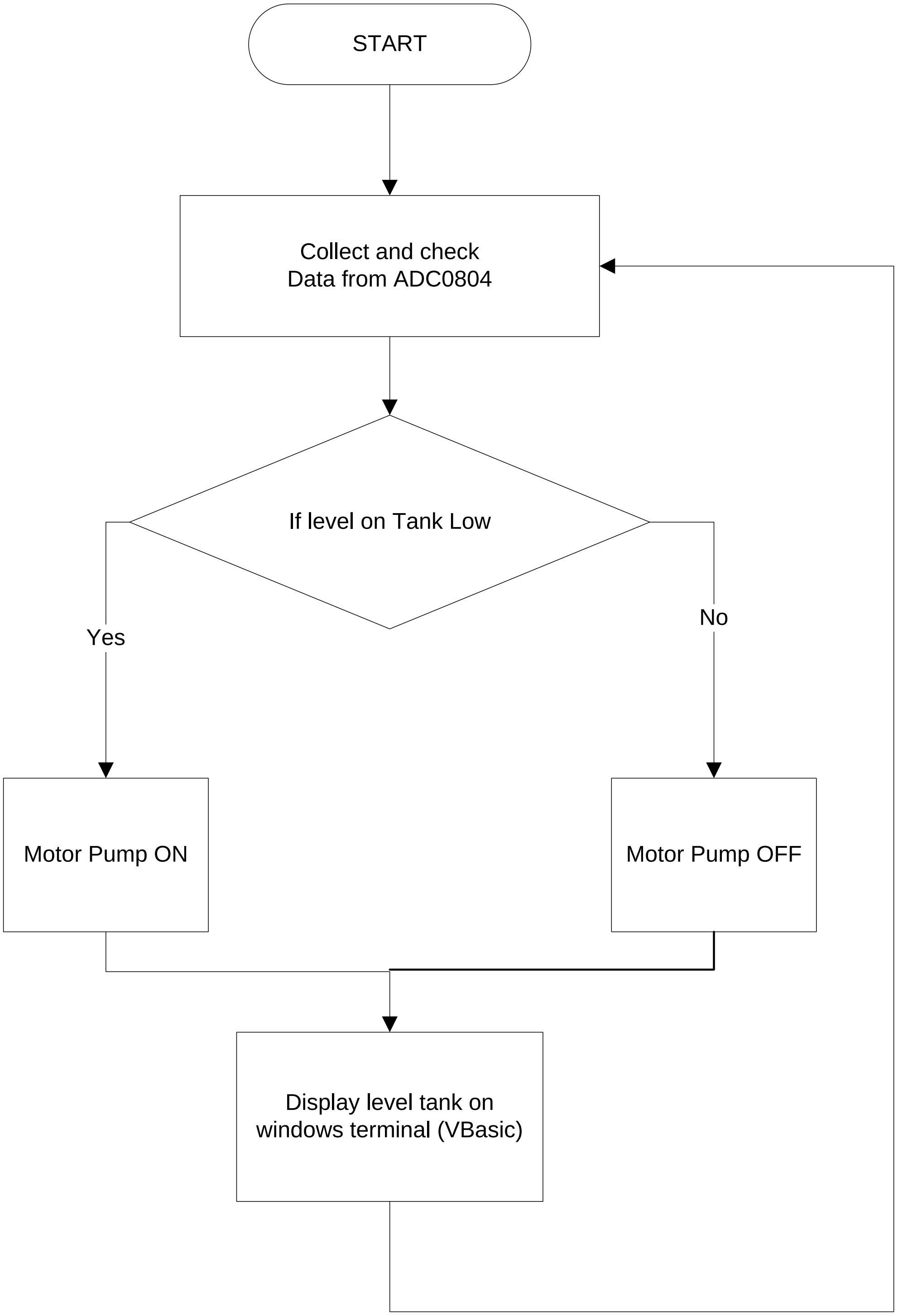
3.3 Programming.

Programming is the main key to controlling the all process in system design. In order to processing data from ADC, display the result obtained by the hardware and controls the signal flow of the system. Two different programs were used, which is assembly language as machine programming and Visual Basic programming as interface software. Each program operates separately, but they interact with each other through the RS232 communication port.

3.3.1 Assembly Language Programming

The 8052 assembler language is a set of mnemonics specific to the 8051 family of microcontrollers. To be able to program in this language using the mnemonics requires a thorough knowledge of the architecture of the 8051 series microcontroller being programmed. The mnemonics basically provide the tools to allow moving the data about the microcontroller and activating its ports.

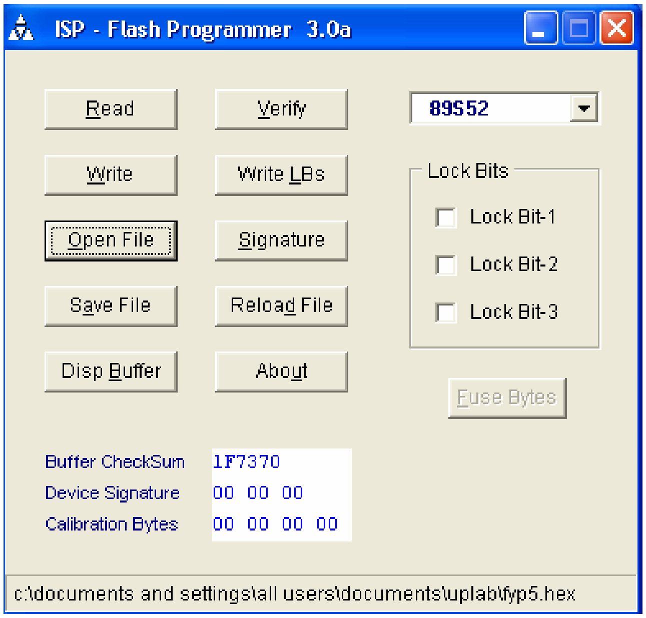
Figure 3.10 shows the flow chart of microcontroller 8051 programming which is closer to the hardware, and it is responsible for acquiring signals that have been converted into digital signal from ADC and control the motor pump using the data from sensor.



Fi

gure 3.10: Flow Chart for Microcontroller

ISP flash programmer in Figure 3.11 is user friendly interface software. As a device programmer, it really easy to use to write a program in the microcontroller chip.



89S52

Lock Bits

□ Lock Bit-1

□ Lock Bit-2

□ Lock Bit-3

Buffer CheckSum 1F7370

Device Signature 00 00 00

Calibration Bytes 00 00 00 00

c:\documents and settings\all users\documents\uplab\fyp5.hex

Figure 3.11: ISP Flash Programmer Software Interface

3.3.2 Windows Interface Setting Using Visual Basic Programming

Visual Basic (VB) Programming is used as software to develop the windows of water monitoring system project. With the display on personal computer, end user has ability to monitor the level of water in the tank. The display window interface can display output value to end user by fetching the processed data from the RS232 which is sent by the microcontroller.

Flow chart of VB programming for setup the graphic user interface (GUI) is shown in Figure 3.12.

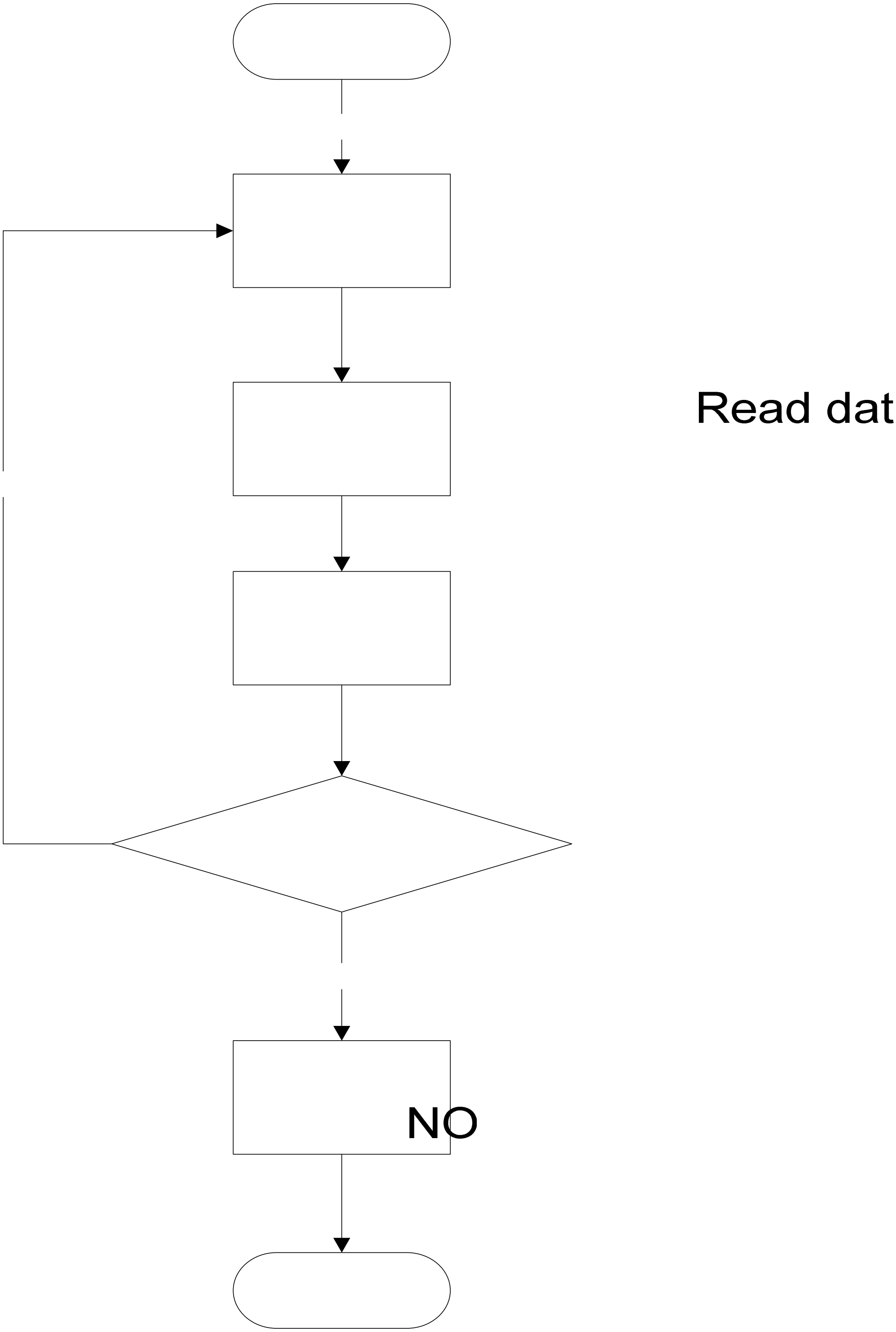


Figure 3.12: Flow Chart of Visual Basic Programming

The Visual Basic program will start by reading data from serial port; refer Figure 3.13 for the source code to setting the serial port. Data will transfer from microcontroller through the serial port and display on windows. Window will display the current level on tank and the water level increase; refer Figure 3.14 for source code.

```
Dim Data As String

Private Sub Form_Load()

Form1.Height = 1065
MSComm1.CommPort = 1
MSComm1.Settings = "9600,N,8,1"
MSComm1.InputLen = 0
MSComm1.PortOpen = True

End Sub
```

Figure 3.13: Source Code for Setting Serial Port RS232

```
If Data = "LO" Then
   Label8.Caption = "Low Level"
   Shape2.Height = 120
   Shape2.Top = 7200
   Timer1.Enabled = False

ElseIf Data = "25" Then
   Label8.Caption = "level 1"
       Shape2.Height = 1100
   Shape2.Top = 6160
   Timer1.Enabled = False
```

Figure 3.14: Source Code for Control Water Level on GUI

CHAPTER 4

RESULT AND DISCUSSION

4.1 Sensor

Distance measurement sensor is a heart of this project. It is important to find and choose the right sensor. In order to make sure that sensor suitable and can operate as required, sensor testing are done.

Based on sensor’s datasheet, the detection distances range is from 4 cm to 30 cm are linear. This characteristic has to considered for determine the distance range for the water level of this project.

Several laboratory tests using this sensor were done to get the exact data as an input signal for microcontroller. The sensor have been placed in the position faced on a white sheet of paper straight and measured the output voltage at several incremental distances scales from the sheet of paper measured. Table 4.1 shows the reading of the four conducted test as well as an average of the result.

Table 4.1: Sensor Testing Results

Distance(cm)	Reading Voltage$_{\text{Trial1}}$(V)	Reading Voltage$_{\text{Trial2}}$(V)	Reading Voltage$_{\text{Trial3}}$(V)	Reading Voltage$_{\text{Trial4}}$(V)	Voltage $_{\text{Average Of Trials}}$(V)
1	1.59	1.57	1.57	1.59	1.58
2	1.92	1.94	1.96	1.97	1.95
3	2.94	2.92	2.92	2.94	2.93
4	2.69	2.69	2.72	2.71	2.70
5	2.25	2.28	2.28	2.28	2.27
6	1.96	1.95	1.97	1.98	1.97
7	1.72	1.72	1.7	1.72	1.72
8	1.52	1.53	1.54	1.54	1.53
9	1.37	1.39	1.37	1.39	1.38
10	1.24	1.24	1.24	1.25	1.24
11	1.15	1.13	1.14	1.14	1.14
12	1.05	1.05	1.05	1.05	1.05
13	0.96	0.98	0.98	0.98	0.98
14	0.92	0.92	0.92	0.92	0.92
15	0.86	0.86	0.86	0.86	0.86
16	0.8	0.8	0.8	0.8	0.80
17	0.76	0.76	0.76	0.76	0.76
18	0.71	0.72	0.72	0.72	0.72
19	0.67	0.69	0.69	0.69	0.69
20	0.65	0.65	0.65	0.65	0.65
21	0.61	0.61	0.61	0.61	0.61
22	0.57	0.59	0.59	0.59	0.59
23	0.55	0.56	0.55	0.56	0.56
24	0.53	0.53	0.53	0.53	0.53
25	0.51	0.51	0.51	0.51	0.51

The result reading of that sensor had been plot on graph for average voltage versus distance shown in Figure 4.1

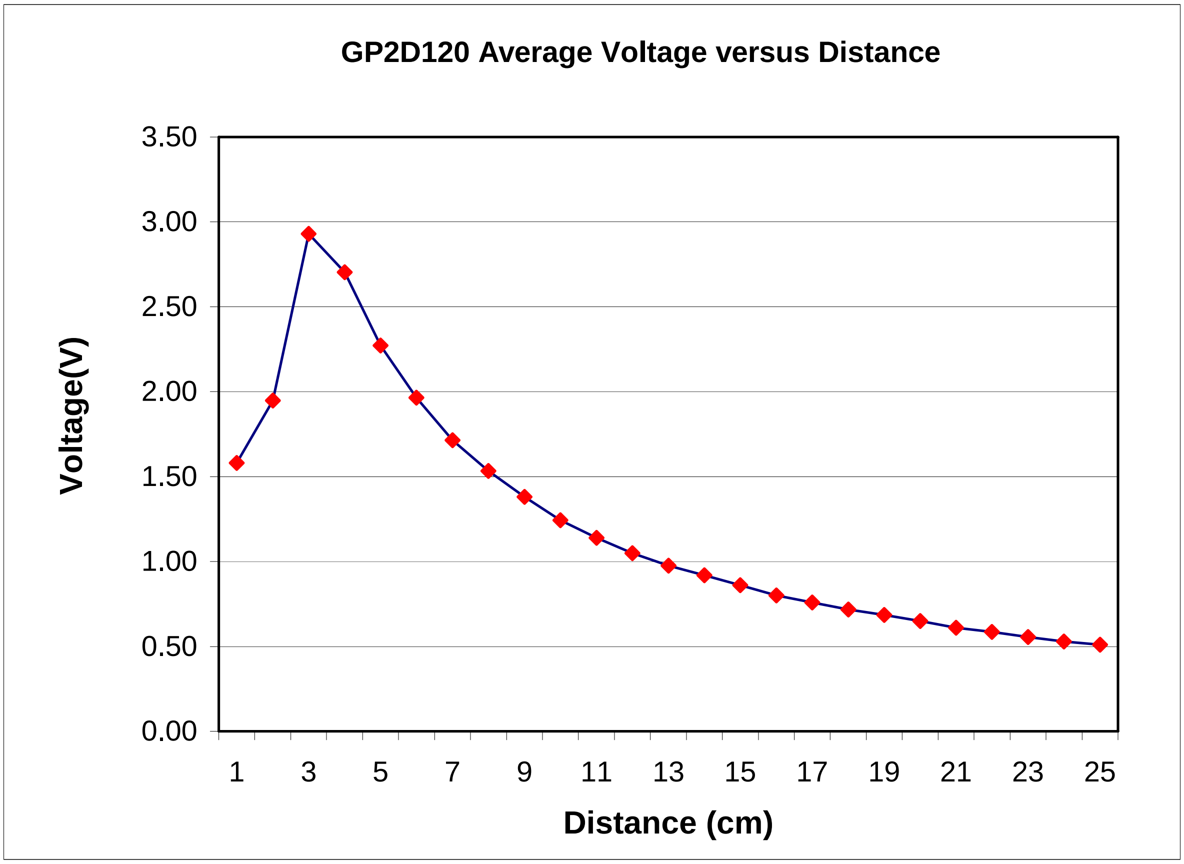


Figure 4.1: GP2D120 Average Voltage versus Distance

Because of some basic trigonometry within the triangle from the emitter to reflection spot to receiver of the sensor, the output voltage of these new detectors is non-linear with respect to the distance being measured. Moreover, as expected from the device specifications, there is a range, namely closer than 4 cm to the object, where the distance cannot be accurately measured.

4.2 Signal Conversion

A sensor has sense the object from multiple ranges of distances and provided a voltage output signals. Table 4.2 shows the output voltage reading of an ADC based on the input signals from a sensor.

Table 4.2: ADC0804 Output Voltage

Input Voltage(V) to ADC at Vin+	Output Voltage (V) D0	Output Voltage (V) D1	Output Voltage (V) D2	Output Voltage (V) D3	Output Voltage (V) D4	Output Voltage (V) D5	Output Voltage (V) D6	Output Voltage (V) D7
3.0	1.50	4.80	4.80	4.80	4.87	0.00	0.00	3.49
2.5	3.28	4.67	0.00	0.00	0.00	0.00	0.00	3.50
2.0	2.52	2.60	2.00	4.90	0.00	3.96	4.09	0.00
1.5	1.50	3.00	0.00	0.00	4.60	0.00	4.09	0.00
1.0	1.50	3.38	4.80	0.00	4.80	3.67	0.00	0.00
0.5	2.50	2.50	0.00	4.85	4.85	0.00	0.00	0.00

Assembly language are formulate as a command to transfer an input signals with the difference voltage values from sensor to an ADC. These values were converted to 8 bits binary data as an ADC output signal which is shown in Table 4.3 where logic 1 stated an output voltage larger than 3.0 V and logic 0 stated output voltage smaller than 3.0V.

Table 4.3: ADC0804 Output in Binary

Input Voltage(V) to ADC at Vin+	Output Voltage in Binary D0	Output Voltage in Binary D1	Output Voltage in Binary D2	Output Voltage in Binary D3	Output Voltage in Binary D4	Output Voltage in Binary D5	Output Voltage in Binary D6	Output Voltage in Binary D7
3.0	0	1	1	1	1	0	0	1
2.5	1	1	0	0	0	0	0	1
2.0	0	0	0	1	0	1	1	0
1.5	0	1	0	0	1	0	1	0

1.0	0	1	1	0	1	1	0	0
0.5	0	0	0	1	1	0	0	0

Microcontroller with source code in Appendix A is used to convert these ADC binary values into hexadecimal values. Table 4.4 shows the output of ADC in hexadecimal values. Figure 4.2 shows the assembly language code how to compare the ADC value with the output range.

Table 4.4: ADC0804 Output in Hexadecimal

Distances (cm)	Input Voltage(V) From sensor to ADC at Vin+	Output Voltage (Hex) from ADC display on TERMULTR			
		Reading 1	Reading 2	Reading 3	Reading 4
25	0.51	1B	1A	19	1A
20	0.65	21	22	23	21
15	0.86	2C	2D	2C	2D
10	1.25	40	41	40	41
5	2.28	77	76	74	75

```
            mov a,p1                ; get data  from ADC
            cjne a,#19h,xsame19     ; compare data  from ADC  with 19H
            jmp cm25
xsame19:    jnc more19
            mov dptr,#XVALID
            call tmltr
            jmp start
more19:     cjne a,#21h,xsame21     ; compare data  from ADC  with 21H
            jmp cm20
```

Figure 4.2: Source Code to Compare Output Range with ADC Value

4.3 User Interface

Based on Table 4.4, the specific values of distance range were identified. There are 5 ranges of distances. These ranges were compared to an input signal from an ADC to determine the level of the water in tank. Also it was used as a controller signal for the pumping motor systems.

Table 4.5: Outputs Range for GUI.

Distances (cm)	Output Range (Hexadecimal)	The Output Display on GUI
> 25	$X < 19$	LO (LOW)
25	$19 \le X < 21$	25
20	$21 \le X < 2C$	20
15	$2C \le X < 40$	15
10	$40 \le X < 76$	10
5	$76 \le X < 78$	5
< 5	$X \ge 78$	HI (HIGH)

The hexadecimal output ranges in Table 4.5 was used to develop a graphic user interface (GUI). VB programming was applied as software tool to produce an interface source code for GUI. Figure 4.2 is a display window for Water Monitoring System. The source code of VB programming can be referring on Appendix C.

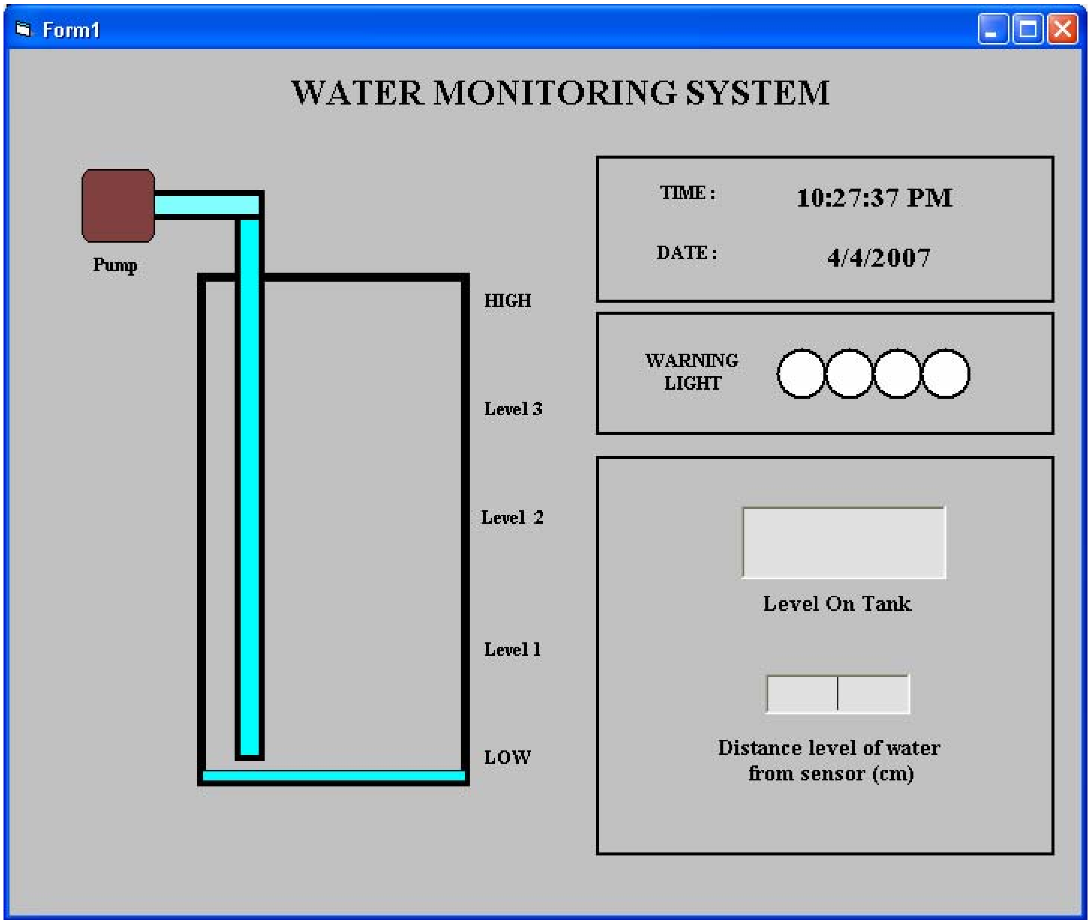


Figure 4.3: Water Monitoring System Display Window

On the display window, it shows the level of water inside the tank, the value of water distance from sensor in cm and also the warning light as an alarm which will blinking to alert user when the tank was full.

Hardware Design

Basically, the construction of hardware design of this project can be present as 3 main modules. Plate 4.1 shows an ADC circuit, Plate 4.2 shows a pump driven circuit, Plate 4.3 shows the position of sensor and Plate 4.4 shows the combination of all circuit with microcontroller inside the black box.

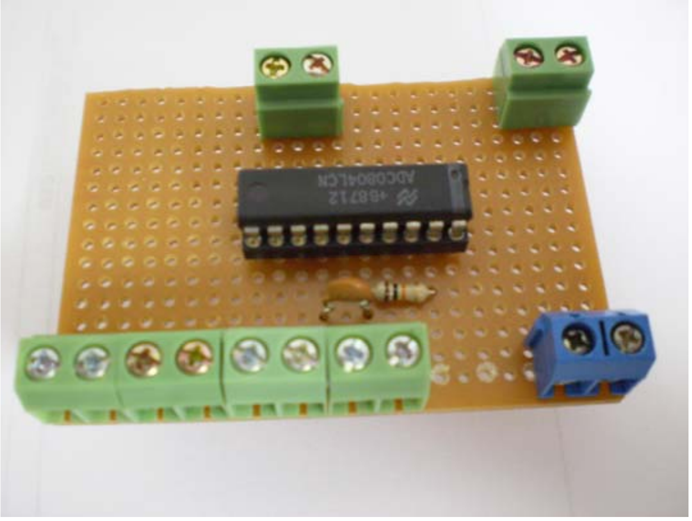


Plate 4.1: ADC Circuit

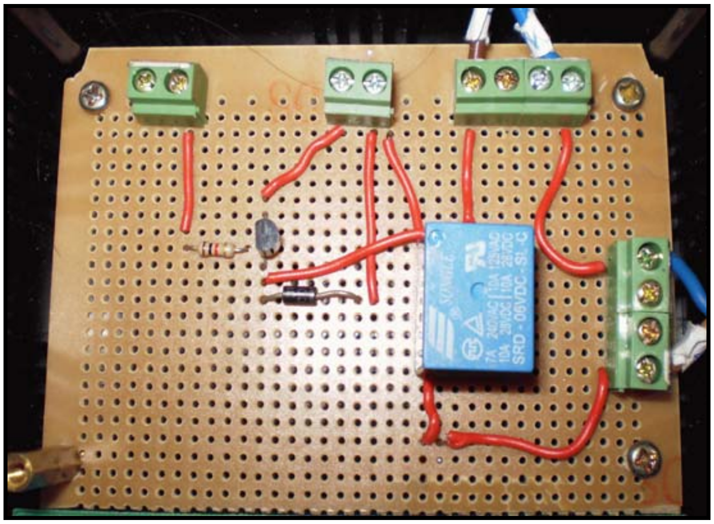


Plate 4.2: Pump Circuit with Transistor Driven Relay.

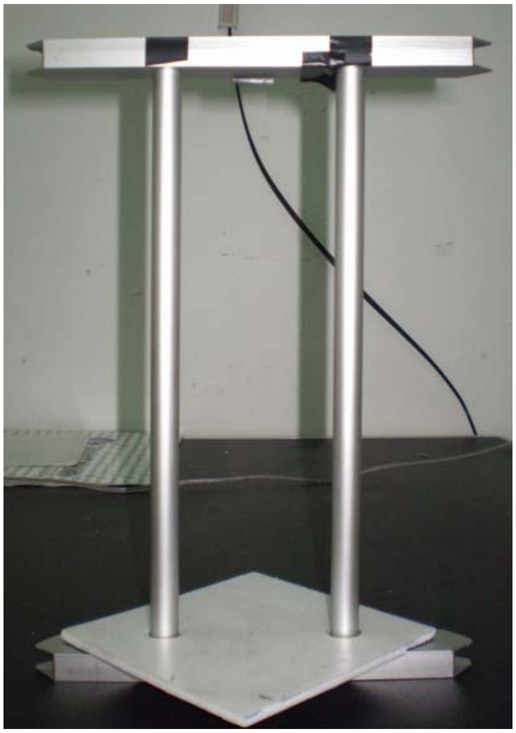


Plate 4.3: Sensor Position

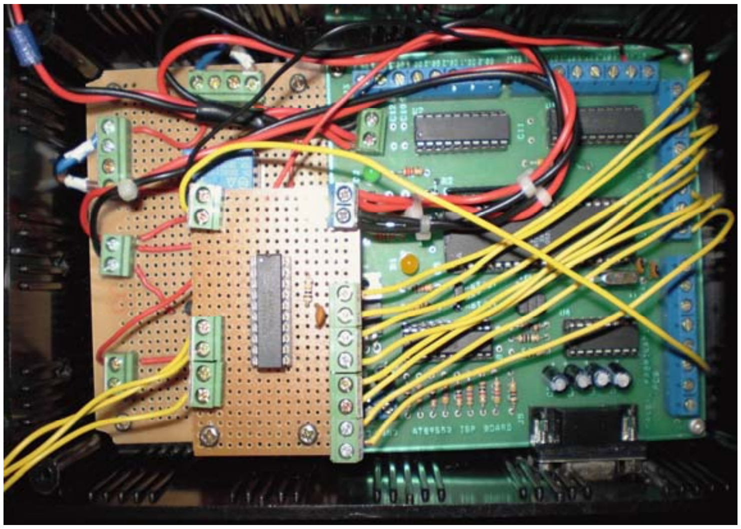


Plate 4.4: Combination all Circuit

CHAPTER 5

CONCLUSION

5.1 Summary

Water monitoring system is a project that can be applied in agriculture sector. This project is the solution to help the user to pump water from well reservoir or river into the water tank using the automatic pumping system and monitor the water level on the dump monitoring system. As agriculture is been focused nowadays, it is important to apply this project in a wide farm.

Knowledge about selecting the transducer as a detector to detect the water level is very important. The detector is a great addition to the suite of detectors available to detect distance of the object. They are quite inexpensive, use very little power, fit in small spaces, and have a unique range that is ideally suited to small robots in human spaces such as hallways, rooms, and the occasional maze.

A pump system in this project is using an ac pump. Transistor driven relay circuit is integrate together to make it can operate in dc condition because it is able to control an output circuit. If the coil is energized with DC voltage, a diode is frequently installed across the coil, to dissipate the energy from the collapsing magnetic field at deactivation, which would otherwise generate a spike of voltage and might cause damage to circuit components. If the coil is designed to be energized with AC voltage, a small copper ring can be crimped to the end of the solenoid. This "shading ring" creates a small out-of-phase current, which increases the minimum pull on the armature during the AC cycle

Upon completion of this project, understandable in the basic and fundamentals of data acquisition theories is very important. ADC is used to convert analog signal to digital signal. The simplest digital signals have only two states, and are called binary. All whole numbers can be represented in binary form as strings of ones and zeros. Digital signals propagate more efficiently than analog signals, because digital impulses, which are well-defined and are easier for electronic circuits to distinguish from noise, which is chaotic. This is the chief advantage of digital modes in communications. Computers "talk" and "think" in terms of binary digital data; while a microprocessor can analyze analog data, it must be converted into digital form for the computer to make sense of it.

This project was successful, where it achieves the goal and main objective that is to contribute to the society use. It can be concluded that the theories learned are proven true indeed. During developing this project, there are some difficulties that occur such as misconnection and unsynchronized programming with hardware. This contribute problem to the project flow and need very determine trouble shooting.

5.2 Recommendation For Future Project

Water monitoring system has a good potential to implement in future especially for agriculture sector. This system is connected in serial connection between microcontroller and PC interfacing. This project will be more effective if it can be replaced with wireless connection which is accomplished without the use of a "hard wired" connection. With wireless communication, the information or data can be transfer more efficiently between monitor system and pump system for any distances involved.

This project can be expanded by using 2 pumps instead of using 1 pump. The first pump will be used to allow the amount of water from sources that will be flow into the tank and the other pump will be used to flow the water out from tank which controlled by using the other type of sensor. More sensors will be suggested to be more application for future project such as temperature sensor or humidity sensor. The temperature sensor will be suggested to control the water flow from tank to farm by detecting the temperature's environment changes.

The other recommendation to improve this project is by adding new features at interfacing system. The interfacing system can add 'print' and 'save data' features so that users can print or save the present data for documentation and references.

REFERENCES

1. A.Hamidat, B.Benyoucef, T.Hartani, (2003). Small-Scale Irrigation with Photovoltaic Water Pumping System in Sahara Regions, 28, pp.1081-1096.

2. H S Kalsi, (1995) .*Electronic Instrumentation*. Tata MGraw-Hill

3. Muhammad Ali Mazidi and Janice Gillispie Mazidi, (1999).*The 8051 Microcontroller and Embedded System.* 1st Edition. Prentice hall.

4. David I. Schneider, (2001). *Introduction to Visual Basic 6.0*. Prentice Hall, Upper Saddler River, New Jersey.

5. Thomas L.Floyd, (2002). *Electronic Devices*. 6th Edition. Prentice Hall International, Inc, New Jersey.

6. CS Rangan, GR Sarma and VSV Mani, (1997).*Instrumentation Devices & Systems*. Tata McGraw-Hill Publishing, Delhi.

7. M.Morris Mano and Charles R.Kimie, (2001). *Logic and Computer Design Fundamentals.* 2nd Edition, Prentice Hall, New Jersey.

8. www.iguanalabs.com

APPENDICES

Appendix A: Assembly Language Program for ADC Data Conversion

```
cpu "8051.tbl"
incl "8051.inc"

org 50h
volt:   dfs 1

org 0

mov tmod, #00100001B
mov scon,#01010010B
mov th1,#253
setb tr1

start:
clr p2.1          ; clear port output

setb p2.0         ; setting star conversion from ADC
nop
clr p2.0
nop
nop
nop
setb p2.0

jb p3.2,$

mov a,p1          ;get data  from ADC
```

```
cjne a,volt,xsame
jmp start

xsame:
mov volt,a
swap a
anl a,#0fh

mov dptr,#table
movc a,@a+dptr

call tx
mov a,volt
anl a,#0fh
mov dptr,#table
movc a,@a+dptr
call tx

jmp start

tx:     jnb ti,$          ; transfer data to serial port
        clr ti
        mov sbuf,a
        ret

table: dfb "0123456789ABCDEF"
```

Appendix B: Assembly Language Program for Water Monitoring System

```
cpu "8051.tbl"
incl "8051.inc"

org 0

mov tmod, #00100001B
mov scon,#01010010B
mov th1,#253
setb tr1

clr p2.1         ;output

start:           call delay
                 setb p2.0                    ; setting star conversion from ADC
                 nop
                 clr p2.0
                 nop
                 nop
                 nop
                 setb p2.0

                 jb p3.2,$
                 mov a,p1                     ; get data  from ADC

                 cjne a,#19h,xsame19          ; compare data  from ADC  with 19H
                 jmp cm25
```

```
xsame19:    jnc more19
            mov dptr,#XVALID
            call tmltr
            jmp start

more19:     cjne a,#21h,xsame21         ; compare data  from ADC  with 21H
            jmp cm20

xsame21:    jnc more21
cm25:       mov dptr,#TABLE25
            call tmltr
            setb p2.1                   ;  set output to run  pump circuit
            jmp start

more21:     cjne a,#2ch,xsame2c         ; compare data  from ADC  with 2C
            jmp cm15
xsame2c:    jnc more2c
cm20:       mov dptr,#TABLE20
            call tmltr
            setb p2.1                   ;  set output to run  pump circuit
            jmp start

more2c:     cjne a,#40h,xsame40         ; compare data  from ADC  with 40H
            jmp cm10
xsame40:    jnc more40
cm15:       mov dptr,#TABLE15
            call tmltr
            setb p2.1                   ;  set output to run  pump circuit
            jmp start

more40:     cjne a,#76h,xsame76         ; compare data  from ADC  with 76H
```

```
            jmp cm5
xsame76:    jnc more76

cm10:       clr p2.1                        ;  set output to stop pump circuit
            mov dptr,#TABLE10
            call tmltr
            jmp start

more76:     cjne a,#78h,xsame78         ; compare data  from ADC  with 78H
more78:     mov dptr,#XVALID2
            call tmltr
            jmp start
xsame78:    jnc more78

cm5:        clr p2.1                        ;  set  output to stop  pump circuit
            mov dptr,#TABLE5
            call tmltr
            jmp start

tmltr:      clr a
            movc a,@a+dptr
            cjne a,#0,next
            ret

next:       call tx
            inc dptr
            jmp tmltr

tx:         jnb ti,$                        ; transmit data to serial port
            clr ti
            mov sbuf,a
```

```
                ret

rx:             jnb ri,$
                clr ri
                mov a,sbuf
                ret

delay:          mov r2,#100
again:          mov r3,#100
here:           nop
                nop
                djnz r3,here
                djnz r2,again
                ret


XVALID:         dfb "LO",0
XVALID2:        dfb "HI",0
TABLE25:        dfb "05",0
TABLE20:        dfb "10",0
TABLE15:        dfb "15",0
TABLE10:        dfb "20",0
TABLE5:         dfb "25",0
```

Appendix C: Visual Basic Programming for Water Monitoring System

```
Dim Data As String

Private Sub Form_Load()
MSComm1.CommPort = 1                              ‘Setting Serial Port’
MSComm1.Settings = "9600,N,8,1"
MSComm1.InputLen = 0
MSComm1.PortOpen = True
End Sub



Private Sub timer1_timer()
If Shape6.FillColor = vbWhite Then
      Shape6.FillColor = vbRed
   Else
      Shape6.FillColor = vbWhite
   End If

If Shape7.FillColor = vbWhite Then


      Shape7.FillColor = vbRed
   Else
      Shape7.FillColor = vbWhite
   End If

If Shape8.FillColor = vbWhite Then


      Shape8.FillColor = vbRed
   Else
      Shape8.FillColor = vbWhite
   End If

If Shape9.FillColor = vbWhite Then

      Shape9.FillColor = vbRed
   Else
      Shape9.FillColor = vbWhite
   End If
End Sub



Private Sub Timer2_Timer()
Text1.Text = MSComm1.Input          'Commands to Received Data from Microcontroller'
Data = Text1.Text



 If Data = "LO" Then                                 'Control level Water'
   Label8.Caption = "Low Level"              'Shape will increase when level increase '
   Shape2.Height = 120                               '
   Shape2.Top = 7200
   Timer1.Enabled = False

ElseIf Data = "25" Then
   Label8.Caption = "level 1"
   Shape2.Height = 1100
   Shape2.Top = 6160
   Timer1.Enabled = False

ElseIf Data = "20" Then
```

```
    Label8.Caption = "level 2"
    Shape2.Height = 2340
    Shape2.Top = 4920
    Timer1.Enabled = False

ElseIf Data = "15" Then
    Label8.Caption = "level 3"
    Shape2.Height = 3660
    Shape2.Top = 3600
    Timer1.Enabled = False

ElseIf Data = "10" Then
    Label8.Caption = "High level"
    Shape2.Height = 4980
    Shape2.Top = 2280
    Timer1.Enabled = True

ElseIf Data = "05" Then
    Label8.Caption = "High level"
    Shape2.Height = 4980
    Shape2.Top = 2280
    Timer1.Enabled = True

ElseIf Data = "HI" Then
    Label8.Caption = "High level"
    Shape2.Height = 4980
    Shape2.Top = 2280
    Timer1.Enabled = True

End If
End Sub
```

```
Private Sub Timer3_Timer()              ‘Setting time and date’
Label3.Caption = Time
Label6.Caption = Date
End Sub
```

APPENDIX D

Data Sheet GP2D120

APPENDIX E

Data Sheet AT89S52

APPENDIX F

Data Sheet ADC0804

Features

- **Compatible with MCS-51® Products**
- **8K Bytes of In-System Programmable (ISP) Flash Memory**
  - **Endurance: 1000 Write/Erase Cycles**
- **4.0V to 5.5V Operating Range**
- **Fully Static Operation: 0 Hz to 33 MHz**
- **Three-level Program Memory Lock**
- **256 x 8-bit Internal RAM**
- **32 Programmable I/O Lines**
- **Three 16-bit Timer/Counters**
- **Eight Interrupt Sources**
- **Full Duplex UART Serial Channel**
- **Low-power Idle and Power-down Modes**
- **Interrupt Recovery from Power-down Mode**
- **Watchdog Timer**
- **Dual Data Pointer**
- **Power-off Flag**

Description

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

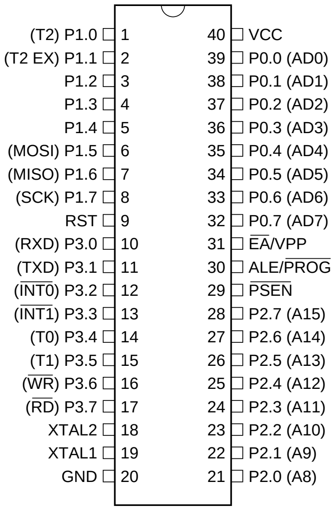
8-bit Microcontroller with 8K Bytes In-System Programmable Flash

AT89S52

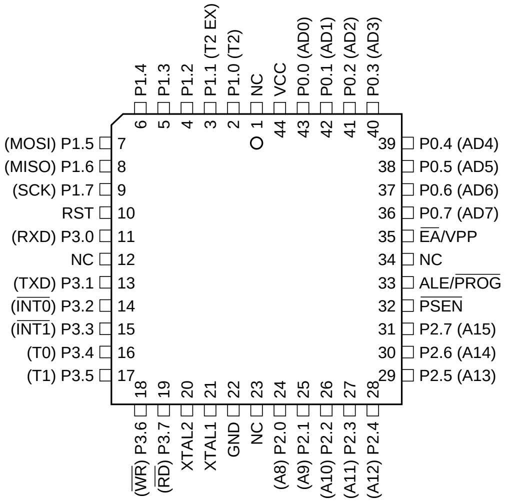
Rev. 1919A-07/01

Pin Configurations

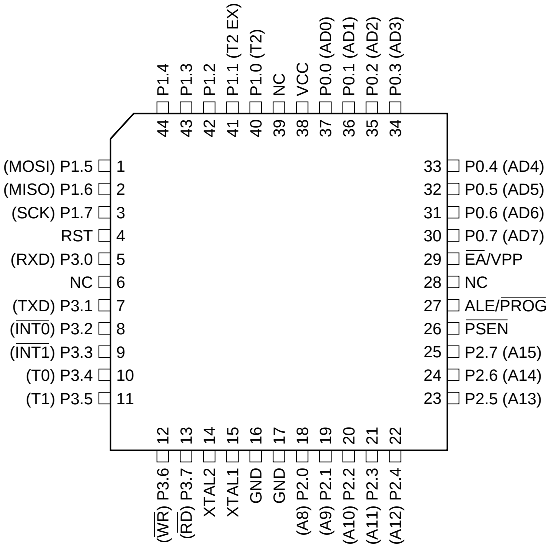
PDIP



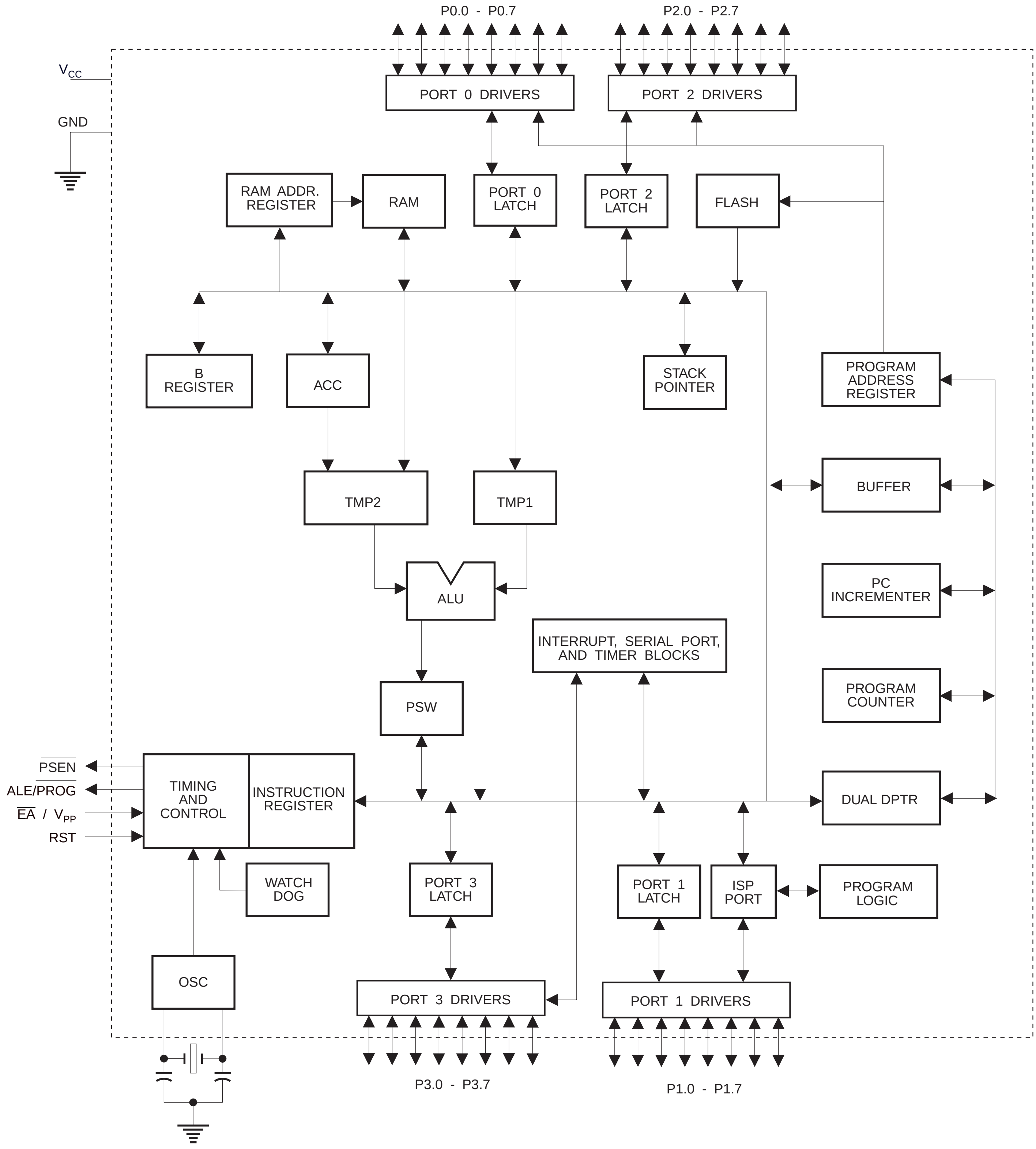
PLCC



TQFP



Block Diagram



Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 96 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/$\overline{\text{PROG}}$

Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{\text{PROG}}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$

Program Store Enable ($\overline{\text{PSEN}}$) is the read strobe to external program memory.

When the AT89S52 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA}}$/VPP

External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89S52 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXXX0				WDTRST XXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers: Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 3) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

Interrupt Registers: The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

Table 2. T2CON – Timer/Counter 2 Control Register

T2CON Address = 0C8H Reset Value = 0000 0000B

Bit Addressable

Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{\text{T2}}$	CP/$\overline{\text{RL2}}$
	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/$\overline{\text{T2}}$	Timer or counter select for Timer 2. C/$\overline{\text{T2}}$ = 0 for timer function. C/$\overline{\text{T2}}$ = 1 for external event counter (falling edge triggered).
CP/$\overline{\text{RL2}}$	Capture/Reload select. CP/$\overline{\text{RL2}}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{\text{RL2}}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Table 3a. AUXR: Auxiliary Register

AUXR Address = 8EH Reset Value = XXX00XX0B

Not Bit Addressable

	–	–	–	WDIDLE	DISRTO	–	–	DISALE
Bit	7	6	5	4	3	2	1	0

–	Reserved for future expansion	
DISALE	Disable/Enable ALE	
	DISALE	Operating Mode
	0	ALE is emitted at a constant rate of 1/6 the oscillator frequency
	1	ALE is active only during a MOVX or MOVC instruction
DISRTO	Disable/Enable Reset out	
	DISRTO	
	0	Reset pin is driven High after WDT times out
	1	Reset pin is input only
WDIDLE	Disable/Enable WDT in IDLE mode	
	WDIDLE	
	0	WDT continues to count in IDLE mode
	1	WDT halts counting in IDLE mode

Dual Data Pointer Registers: To facilitate accessing both internal and external data memory, two banks of 16-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR AUXR1 selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

Power Off Flag: The Power Off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to “1” during power up. It can be set and rest under software control and is not affected by reset.

Table 3b. AUXR1: Auxiliary Register 1

AUXR1 Address = A2H Reset Value = XXXXXXX0B

Not Bit Addressable

	–	–	–	–	–	–	–	DPS
Bit	7	6	5	4	3	2	1	0

–	Reserved for future expansion	
DPS	Data Pointer Register Select	
	DPS	
	0	Selects DPTR Registers DP0L, DP0H
	1	Selects DPTR Registers DP1L, DP1H

Memory Organization

MCS-51 devices have a separate address space for Program and Data Memory. Up to 64K bytes each of external Program and Data Memory can be addressed.

Program Memory

If the $\overline{EA}$ pin is connected to GND, all program fetches are directed to external memory.

On the AT89S52, if $\overline{EA}$ is connected to V_{CC}, program fetches to addresses 0000H through 1FFFH are directed to internal memory and fetches to addresses 2000H through FFFFH are to external memory.

Data Memory

The AT89S52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. This means that the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions which use direct addressing access of the SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
    MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
    MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

Watchdog Timer (One-time Enabled with Reset-out)

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upsets. The WDT consists of a 13-bit counter and the Watchdog Timer Reset (WDTRST) SFR. The WDT is defaulted to disable from exiting reset. To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, it will increment every machine cycle while the oscillator is running. The WDT timeout period is dependent on the external clock frequency. There is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST pin.

Using the WDT

To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 13-bit counter overflows when it reaches 8191 (1FFFH), and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 8191 machine cycles. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write-only register. The WDT counter cannot be read or written. When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is 96xTOSC, where TOSC=1/FOSC. To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

WDT During Power-down and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are two methods of exiting Power-down mode: by a hardware reset or via a level-activated external interrupt which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, servicing the WDT should occur as it normally does whenever the AT89S52 is reset. Exiting Power-down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power-down mode.

To ensure that the WDT does not overflow within a few states of exiting Power-down, it is best to reset the WDT just before entering Power-down mode.

Before going into the IDLE mode, the WDIDLE bit in SFR AUXR is used to determine whether the WDT continues to count if enabled. The WDT keeps counting during IDLE (WDIDLE bit = 0) as the default state. To prevent the WDT from resetting the AT89S52 while in IDLE mode, the user should always set up a timer that will periodically exit IDLE, service the WDT, and reenter IDLE mode.

With WDIDLE bit enabled, the WDT will stop to count in IDLE mode and resumes the count upon exit from IDLE.

UART

The UART in the AT89S52 operates the same way as the UART in the AT89C51 and AT89C52. For further information on the UART operation, refer to the ATMEL Web site (http://www.atmel.com). From the home page, select ‘Products’, then ‘8051-Architecture Flash Microcontroller’, then ‘Product Overview’.

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S52 operate the same way as Timer 0 and Timer 1 in the AT89C51 and AT89C52. For further information on the timers’ operation, refer to the ATMEL Web site (http://www.atmel.com). From the home page, select ‘Products’, then ‘8051-Architecture Flash Microcontroller’, then ‘Product Overview’.

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{\text{T2}}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Table 3. Timer 2 Operating Modes

RCLK +TCLK	CP/$\overline{\text{RL2}}$	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 5.

Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 5. Timer in Capture Mode

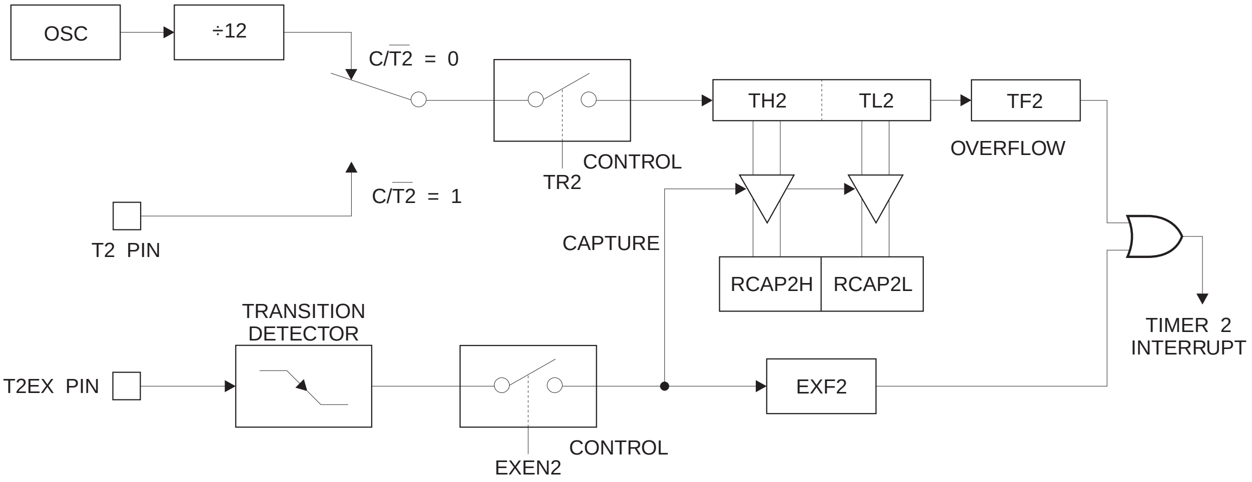


Figure 6 shows Timer 2 automatically counting up when DCEN=0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in Timer in Capture ModeRCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 6. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Figure 6. Timer 2 Auto Reload Mode (DCEN = 0)

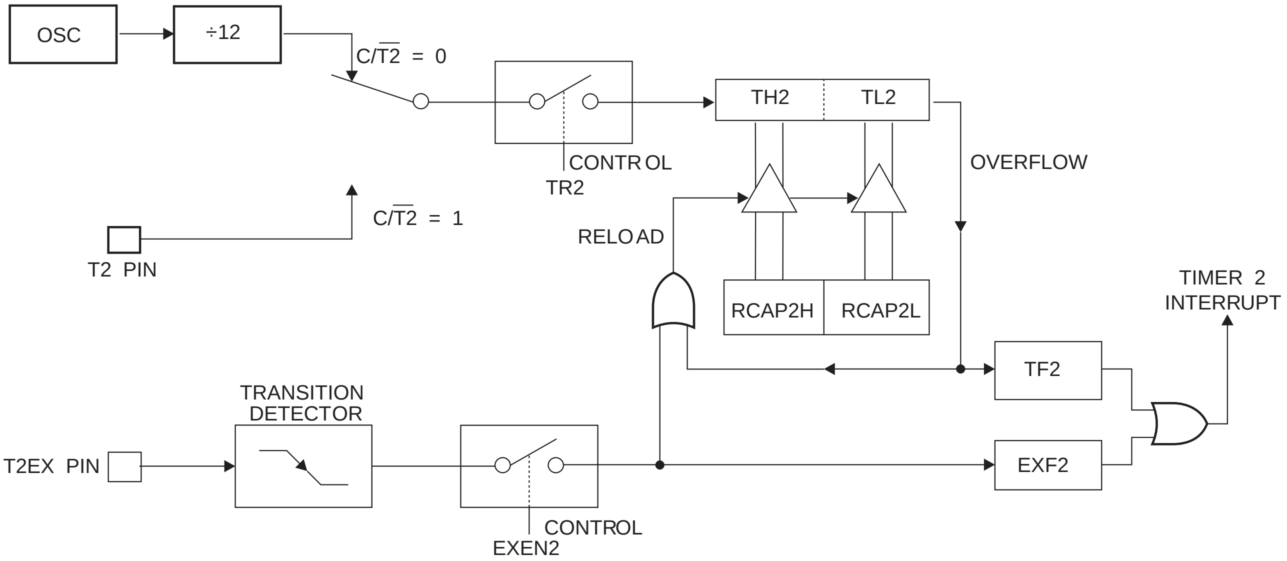


Table 4. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H Reset Value = XXXX XX00B

Not Bit Addressable

	–	–	–	–	–	–	T2OE	DCEN
Bit	7	6	5	4	3	2	1	0

Symbol	Function
–	Not implemented, reserved for future
T2OE	Timer 2 Output Enable bit
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter

Figure 7. Timer 2 Auto Reload Mode (DCEN = 1)

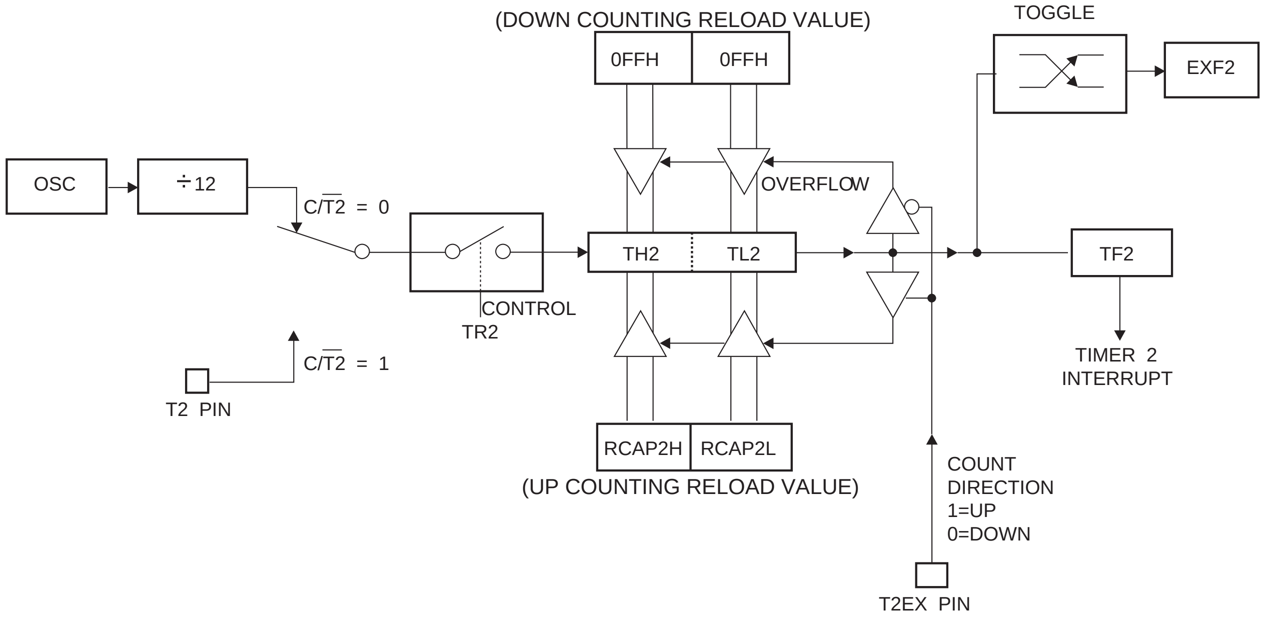
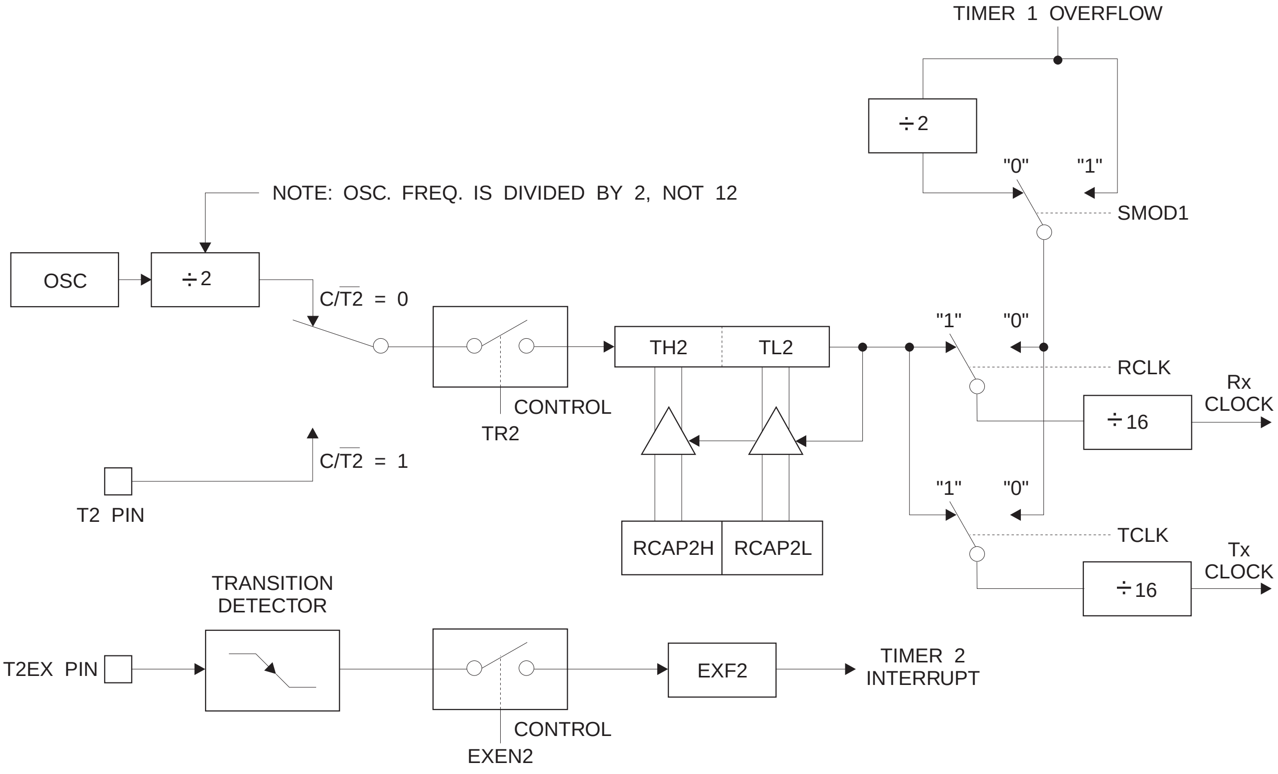


Figure 8. Timer 2 in Baud Rate Generator Mode



Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 8.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation (CP/$\overline{\text{T2}}$ = 0). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

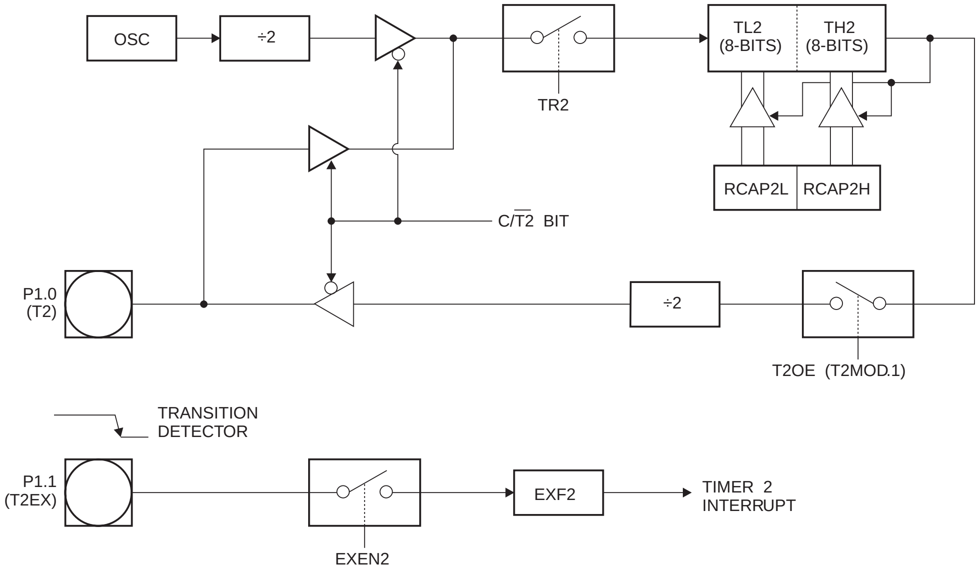
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \text{ x } [65536\text{-RCAP2H,RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 8. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus, when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Figure 9. Timer 2 in Clock-Out Mode



Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 9. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit $C/\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock-Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536-(\text{RCAP2H},\text{RCAP2L})]}$$

In the clock-out mode, Timer 2 roll-overs will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

Interrupts

The AT89S52 has a total of six interrupt vectors: two external interrupts ($\overline{\text{INT0}}$ and $\overline{\text{INT1}}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 5 shows that bit position IE.6 is unimplemented. In the AT89S52, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Table 5. Interrupt Enable (IE) Register

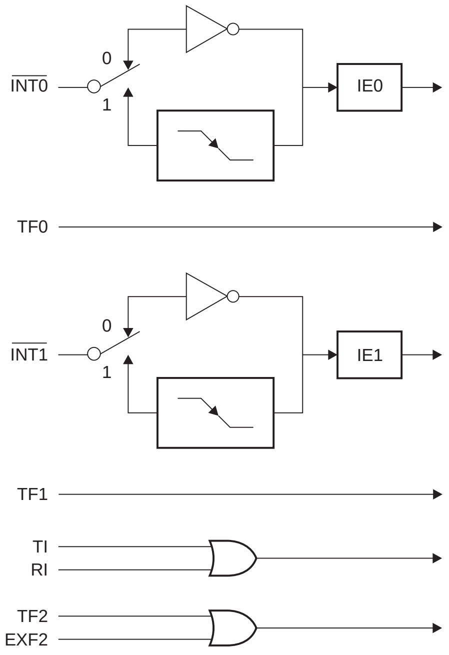
(MSB)							(LSB)
EA	–	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.
User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.		

Figure 10. Interrupt Sources



Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

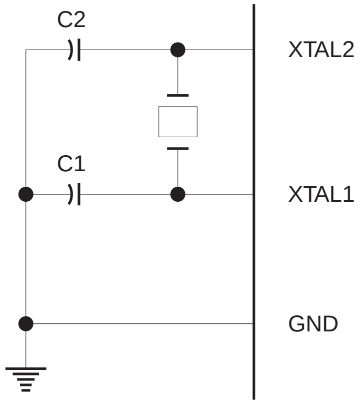
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power-down Mode

In the Power-down mode, the oscillator is stopped, and the instruction that invokes Power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power-down mode is terminated. Exit from Power-down mode can be initiated either by a hardware reset or by an enabled external interrupt. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Figure 11. Oscillator Connections



Note: C1, C2 = 30 pF ±10 pF for Crystals
= 40 pF ±10 pF for Ceramic Resonators

Figure 12. External Clock Drive Configuration

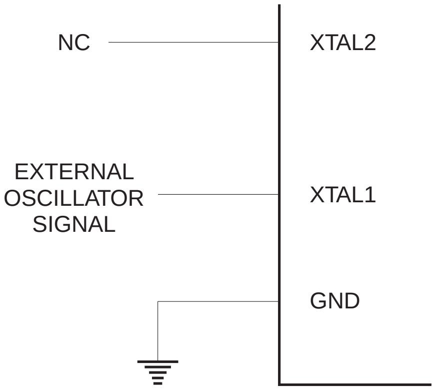


Table 6. Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Program Memory Lock Bits

The AT89S52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Table 7. Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{\text{EA}}$ is sampled and latched on reset, and further programming of the Flash memory is disabled
3	P	P	U	Same as mode 2, but verify is also disabled
4	P	P	P	Same as mode 3, but external execution is also disabled

When lock bit 1 is programmed, the logic level at the $\overline{\text{EA}}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of $\overline{\text{EA}}$ must agree with the current logic level at that pin in order for the device to function properly.

Programming the Flash – Parallel Mode

The AT89S52 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S52 code memory array is programmed byte-by-byte.

Programming Algorithm: Before programming the AT89S52, the address, data, and control signals should be set up according to the Flash programming mode table and Figures 13 and 14. To program the AT89S52, take the following steps:

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{\text{EA}}/V_{PP}$ to 12V.
5. Pulse ALE/$\overline{\text{PROG}}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 50 µs.

Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{\text{Data}}$ Polling: The AT89S52 features $\overline{\text{Data}}$ Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{\text{Data}}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{\text{Busy}}$: The progress of byte programming can also be monitored by the RDY/$\overline{\text{BSY}}$ output signal. P3.0 is pulled low after ALE goes high during programming to indicate $\overline{\text{BUSY}}$. P3.0 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The status of the individual lock bits can be verified directly by reading them back.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 100H, and 200H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(100H) = 52H indicates 89S52
(200H) = 06H

Chip Erase: In the parallel programming mode, a chip erase operation is initiated by using the proper combination of control signals and by pulsing ALE/$\overline{\text{PROG}}$ low for a duration of 200 ns - 500 ns.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 500 ms.

During chip erase, a serial read from any address location will return 00H at the data output.

Programming the Flash – Serial Mode

The Code memory array can be programmed using the serial ISP interface while RST is pulled to V_{CC}. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before other operations can be executed. Before a reprogramming sequence can occur, a Chip Erase operation is required.

The Chip Erase operation turns the content of every memory location in the Code array into FFH.

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

Serial Programming Algorithm

To program and verify the AT89S52 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:

   Apply power between VCC and GND pins.

   Set RST pin to "H".

   If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.

3. The Code array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The write cycle is self-timed and typically takes less than 1 ms at 5V.

4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.

5. At the end of a programming session, RST can be set low to commence normal device operation.

Power-off sequence (if needed):

Set XTAL1 to "L" (if a crystal is not used).

Set RST to "L".

Turn V_{CC} power off.

$\overline{\text{Data}}$ Polling: The $\overline{\text{Data}}$ Polling feature is also available in the serial mode. In this mode, during a write cycle an attempted read of the last byte written will result in the complement of the MSB of the serial output byte on MISO.

Serial Programming Instruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 10.

Programming Interface – Parallel Mode

Every code byte in the Flash array can be programmed by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Table 8. Flash Programming Modes

Mode	V_{CC}	RST	$\overline{PSEN}$	ALE/ $\overline{PROG}$	$\overline{EA}$/ V_{PP}	P2.6	P2.7	P3.3	P3.6	P3.7	P0.7-0 Data	P2.4-0	P1.7-0
												Address	
Write Code Data	5V	H	L	⎺⎽⎺ (2)	12V	L	H	H	H	H	D_{IN}	A12-8	A7-0
Read Code Data	5V	H	L	H	H	L	L	L	H	H	D_{OUT}	A12-8	A7-0
Write Lock Bit 1	5V	H	L	⎺⎽⎺ (3)	12V	H	H	H	H	H	X	X	X
Write Lock Bit 2	5V	H	L	⎺⎽⎺ (3)	12V	H	H	H	L	L	X	X	X
Write Lock Bit 3	5V	H	L	⎺⎽⎺ (3)	12V	H	L	H	H	L	X	X	X
Read Lock Bits 1, 2, 3	5V	H	L	H	H	H	H	L	H	L	P0.2, P0.3, P0.4	X	X
Chip Erase	5V	H	L	⎺⎽⎺ (1)	12V	H	L	H	L	L	X	X	X
Read Atmel ID	5V	H	L	H	H	L	L	L	L	L	1EH	X 0000	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	52H	X 0001	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	06H	X 0010	00H

Notes:
1. Each $\overline{PROG}$ pulse is 200 ns - 500 ns for Chip Erase.
2. Each $\overline{PROG}$ pulse is 200 ns - 500 ns for Write Code Data.
3. Each $\overline{PROG}$ pulse is 200 ns - 500 ns for Write Lock Bits.
4. RDY/$\overline{BSY}$ signal is output on P3.0 during programming.
5. X = don’t care.

Figure 13. Programming the Flash Memory (Parallel Mode)

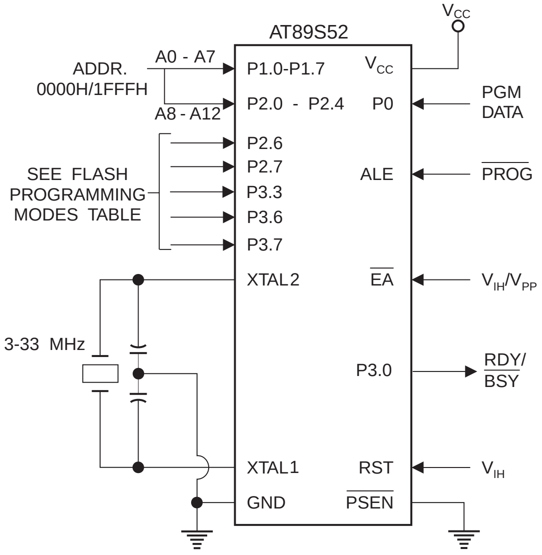
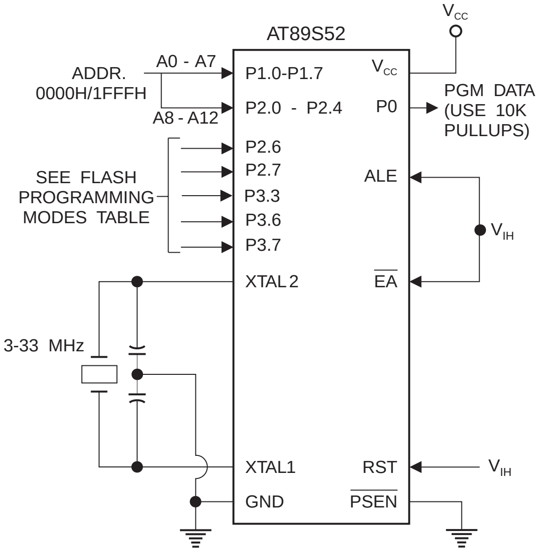


Figure 14. Verifying the Flash Memory (Parallel Mode)



Flash Programming and Verification Characteristics (Parallel Mode)

T_A = 20°C to 30°C, V_{CC} = 4.5 to 5.5V

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	11.5	12.5	V
I_{PP}	Programming Supply Current		10	mA
I_{CC}	V_{CC} Supply Current		30	mA
$1/t_{CLCL}$	Oscillator Frequency	3	33	MHz
t_{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{\text{ENABLE}}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		µs
t_{GHSL}	V_{PP} Hold After $\overline{\text{PROG}}$	10		µs
t_{GLGH}	$\overline{\text{PROG}}$ Width	0.2	1	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		50	µs

Figure 15. Flash Programming and Verification Waveforms – Parallel Mode

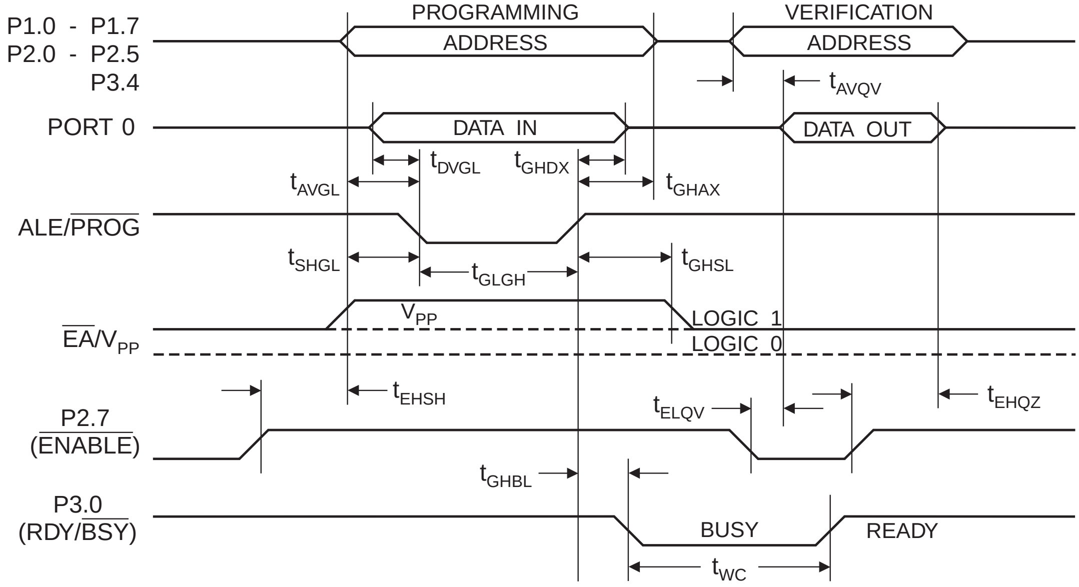
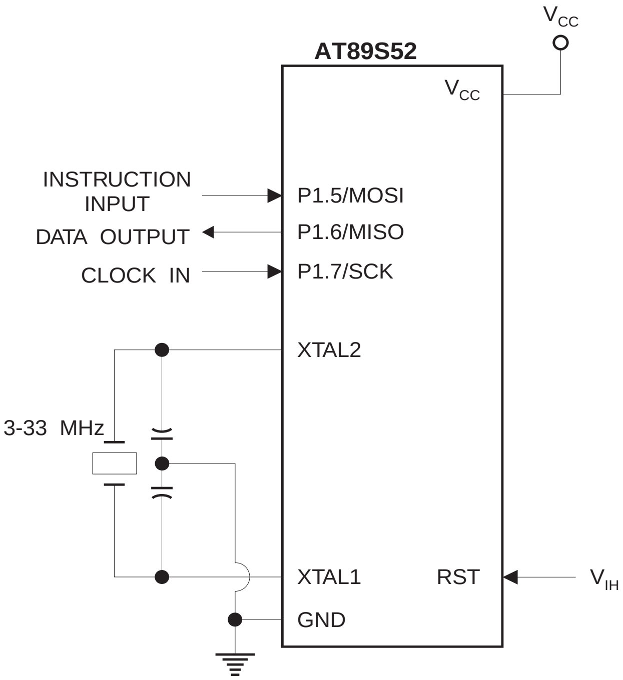
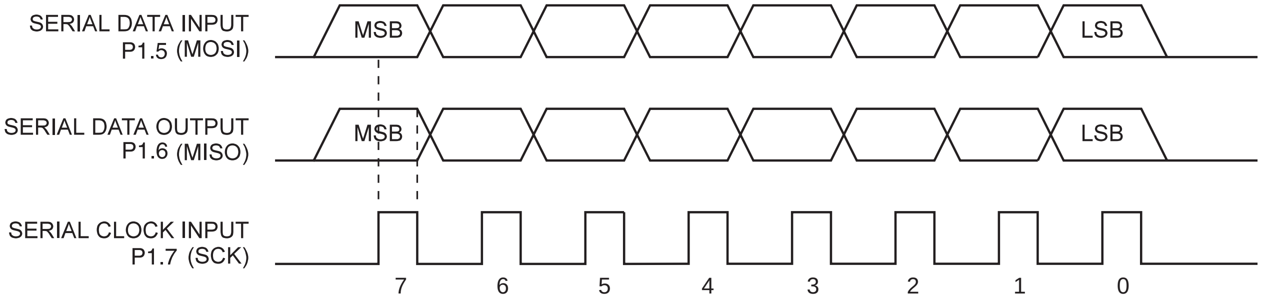


Figure 16. Flash Memory Serial Downloading



Flash Programming and Verification Waveforms – Serial Mode

Figure 17. Serial Programming Waveforms



SERIAL DATA INPUT P1.5 (MOSI)
MSB
LSB
SERIAL DATA OUTPUT P1.6 (MISO)
MSB
LSB
SERIAL CLOCK INPUT P1.7 (SCK)
7
6
5
4
3
2
1
0

Table 9. Serial Programming Instruction Set

	Instruction Format				
Instruction	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Operation**
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Write data to Program memory in the byte mode
Write Lock Bits[(2)]	1010 1100	1110 00 B1 B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (2).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xx LB3 LB2 LB1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a '1')
Read Signature Bytes[(1)]	0010 1000	xxx A5 A4 A3 A2 A1 A0	xxx xxxx	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

Notes: 1. The signature bytes are not readable in Lock Bit Modes 3 and 4.

2. B1 = 0, B2 = 0 ---> Mode 1, no lock protection
B1 = 0, B2 = 1 ---> Mode 2, lock bit 1 activated
B1 = 1, B2 = 0 ---> Mode 3, lock bit 2 activated
B1 = 1, B1 = 1 ---> Mode 4, lock bit 3 activated

} Each of the lock bits needs to be activated sequentially before Mode 4 can be executed.

After Reset signal is high, SCK should be low for at least 64 system clocks before it goes high to clock in the enable data bytes. No pulsing of Reset signal is necessary. SCK should be no faster than 1/16 of the system clock at XTAL1.

For Page Read/Write, the data always starts from byte 0 to 255. After the command byte and upper address byte are latched, each byte thereafter is treated as data until all 256 bytes are shifted in/out. Then the next instruction will be ready to be decoded.

Serial Programming Characteristics

Figure 18. Serial Programming Timing

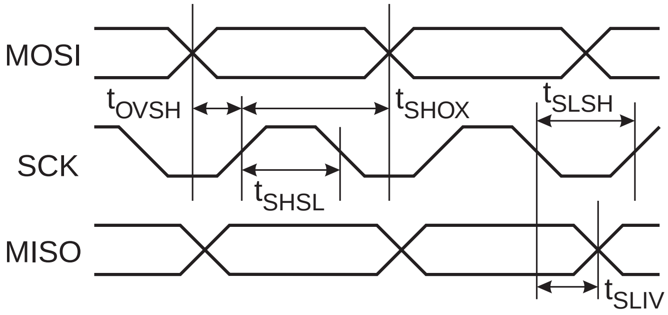


Table 10. Serial Programming Characteristics, T_A = -40° C to 85° C, V_{CC} = 4.0 - 5.5V (Unless otherwise noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0		33	MHz
t_{CLCL}	Oscillator Period	30			ns
t_{SHSL}	SCK Pulse Width High	2 t_{CLCL}			ns
t_{SLSH}	SCK Pulse Width Low	2 t_{CLCL}			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	2 t_{CLCL}			ns
t_{SLIV}	SCK Low to MISO Valid	10	16	32	ns
t_{ERASE}	Chip Erase Instruction Cycle Time			500	ms
t_{SWC}	Serial Byte Write Cycle Time			64 t_{CLCL} + 400	µs

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 4.0V to 5.5V, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	$0.45 < V_{IN} < V_{CC}$		±10	µA
RRST	Reset Pulldown Resistor		10	30	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode[1]	V_{CC} = 5.5V		50	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port:
Port 0: 26 mA Ports 1, 2, 3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power-down is 2V.

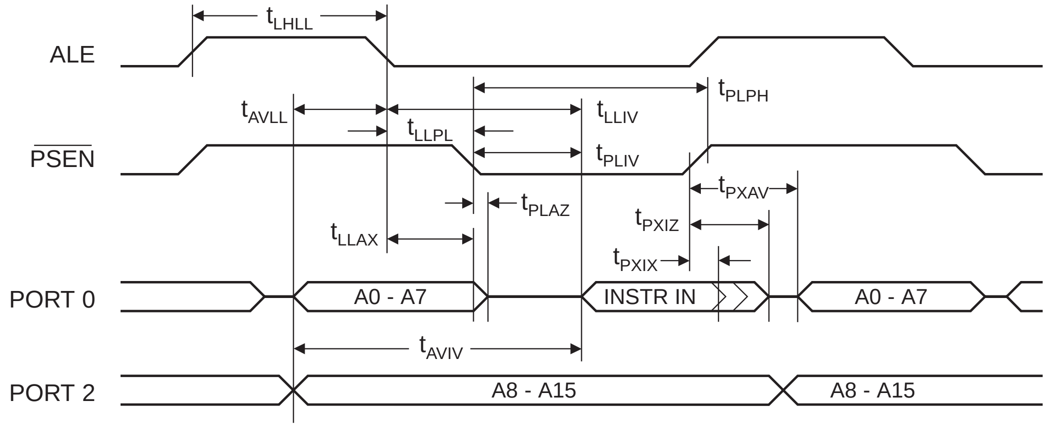
AC Characteristics

Under operating conditions, load capacitance for Port 0, ALE/$\overline{\text{PROG}}$, and $\overline{\text{PSEN}}$ = 100 pF; load capacitance for all other outputs = 80 pF.

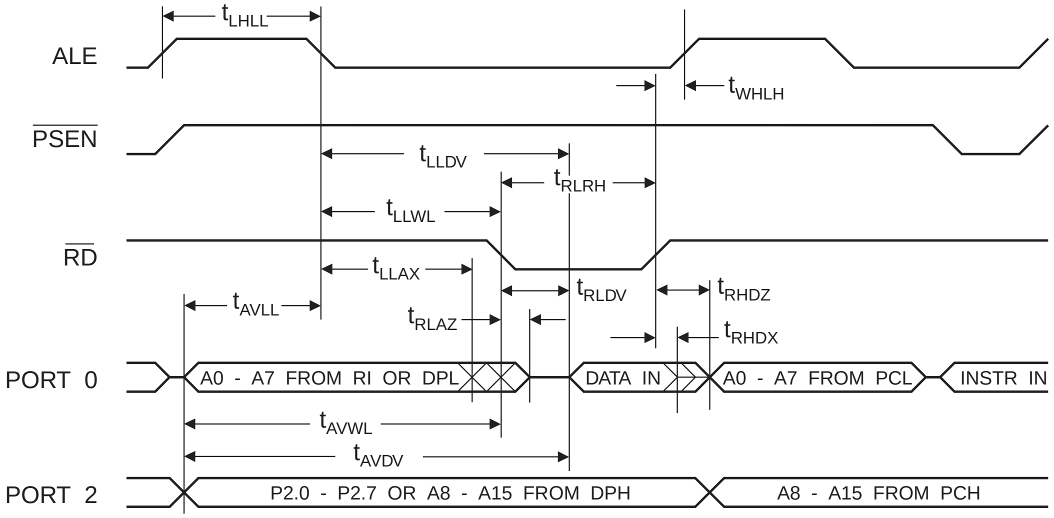
External Program and Data Memory Characteristics

		12 MHz Oscillator		Variable Oscillator		
Symbol	Parameter	Min	Max	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency			0	33	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-25		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-25		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{\text{PSEN}}$ Low	43		t_{CLCL}-25		ns
t_{PLPH}	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{CLCL}$-45		ns
t_{PLIV}	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{CLCL}$-60	ns
t_{PXIX}	Input Instruction Hold After $\overline{\text{PSEN}}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{\text{PSEN}}$		59		t_{CLCL}-25	ns
t_{PXAV}	$\overline{\text{PSEN}}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-80	ns
t_{PLAZ}	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{\text{RD}}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{\text{WR}}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{\text{RD}}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{\text{RD}}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{\text{RD}}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{\text{WR}}$ Transition	23		t_{CLCL}-30		ns
t_{QVWH}	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{CLCL}$-130		ns
t_{WHQX}	Data Hold After $\overline{\text{WR}}$	33		t_{CLCL}-25		ns
t_{RLAZ}	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	t_{CLCL}-25	t_{CLCL}+25	ns

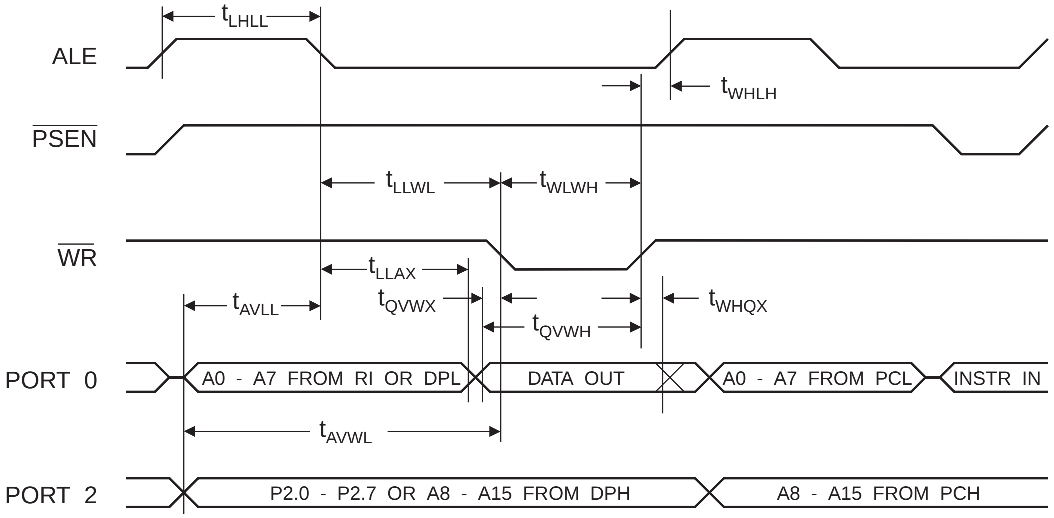
External Program Memory Read Cycle



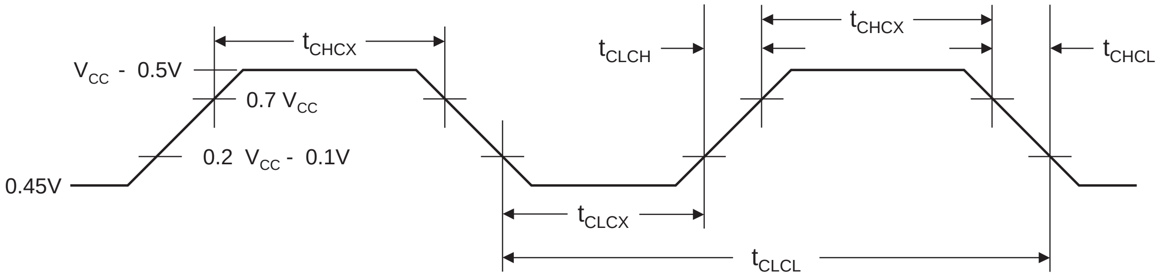
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

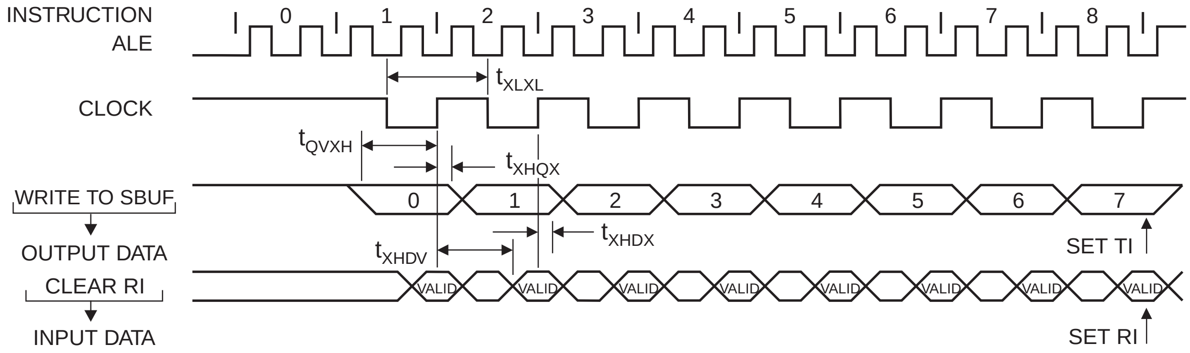
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	33	MHz
t_{CLCL}	Clock Period	30		ns
t_{CHCX}	High Time	12		ns
t_{CLCX}	Low Time	12		ns
t_{CLCH}	Rise Time		5	ns
t_{CHCL}	Fall Time		5	ns

Serial Port Timing: Shift Register Mode Test Conditions

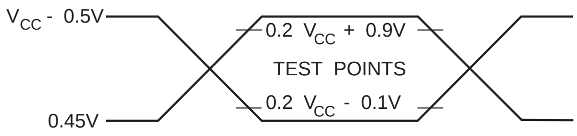
The values in this table are valid for V_{CC} = 4.0V to 5.5V and Load Capacitance = 80 pF.

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-80		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

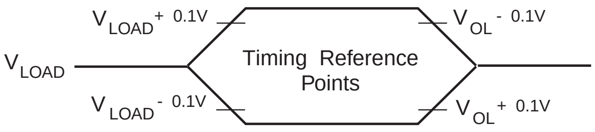


AC Testing Input/Output Waveforms(1)



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms(1)



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S52-24AC	44A	Commercial
		AT89S52-24JC	44J	(0° C to 70° C)
		AT89S52-24PC	40P6	
		AT89S52-24AI	44A	Industrial
		AT89S52-24JI	44J	(-40° C to 85° C)
		AT89S52-24PI	40P6	
33	4.5V to 5.5V	AT89S52-33AC	44A	Commercial
		AT89S52-33JC	44J	(0° C to 70° C)
		AT89S52-33PC	40P6	

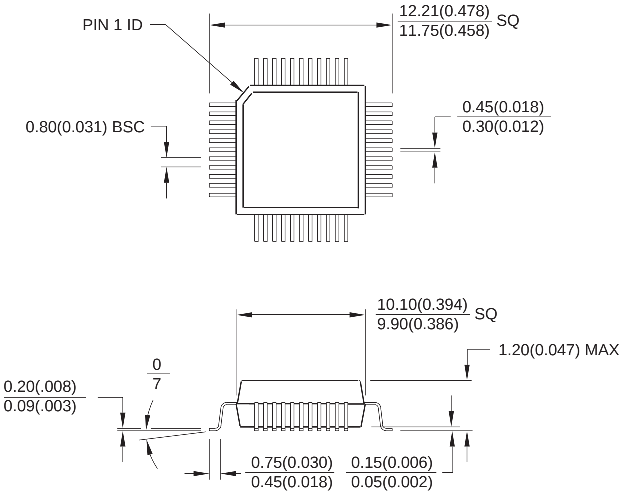
▨ = Preliminary Availability

Package Type	
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)

Packaging Information

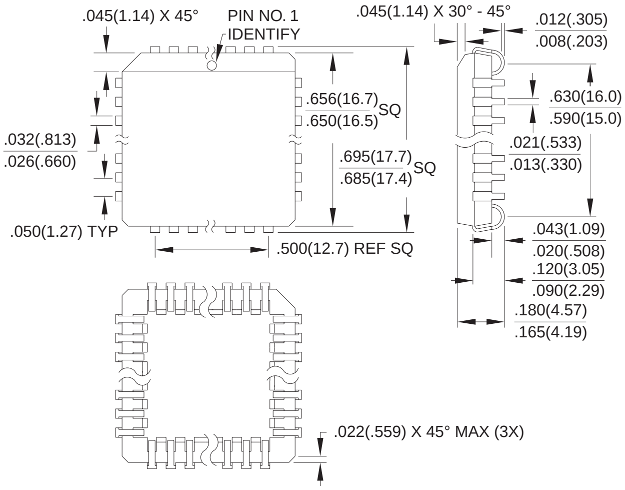
44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP) Dimensions in Millimeters and (Inches)*	**44J,** 44-lead, Plastic J-leaded Chip Carrier (PLCC) Dimensions in Inches and (Millimeters)

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
Dimensions in Millimeters and (Inches)*

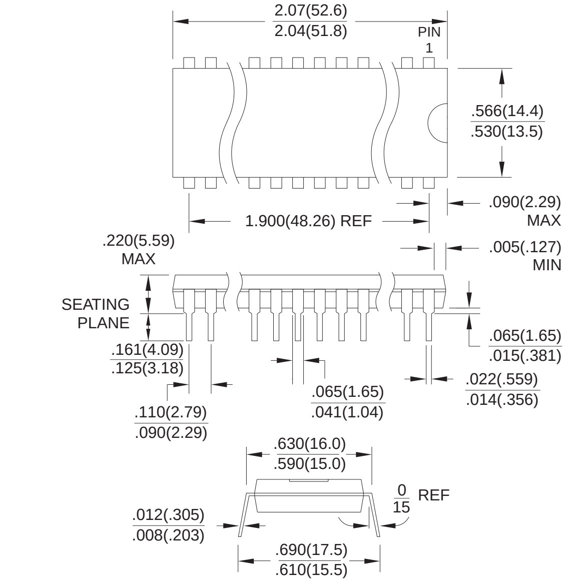


*Controlling dimension: millimeters

44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)



40P6, 40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-011 AC



Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Product Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

Atmel Heilbronn
Theresienstrasse 2
POB 3535
D-74025 Heilbronn, Germany
TEL (49) 71 31 67 25 94
FAX (49) 71 31 67 24 23

Atmel Nantes
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 0 2 40 18 18 18
FAX (33) 0 2 40 18 19 60

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Atmel Smart Card ICs
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Printed on recycled paper.

HARRIS Semiconductor

ADC0802, ADC0803 ADC0804

8-Bit, Microprocessor-Compatible, A/D Converters

August 1997

Features

- 80C48 and 80C80/85 Bus Compatible - No Interfacing Logic Required
- Conversion Time < 100µs
- Easy Interface to Most Microprocessors
- Will Operate in a “Stand Alone” Mode
- Differential Analog Voltage Inputs
- Works with Bandgap Voltage References
- TTL Compatible Inputs and Outputs
- On-Chip Clock Generator
- 0V to 5V Analog Voltage Input Range (Single + 5V Supply)
- No Zero-Adjust Required

Description

The ADC0802 family are CMOS 8-Bit, successive-approximation A/D converters which use a modified potentiometric ladder and are designed to operate with the 8080A control bus via three-state outputs. These converters appear to the processor as memory locations or I/O ports, and hence no interfacing logic is required.

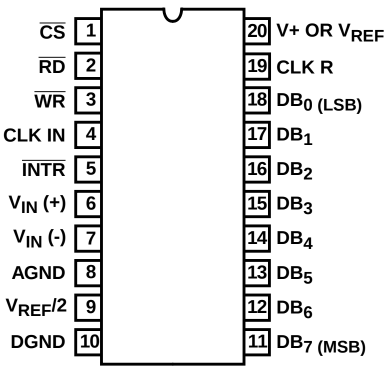
The differential analog voltage input has good common-mode-rejection and permits offsetting the analog zero-input-voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

Ordering Information

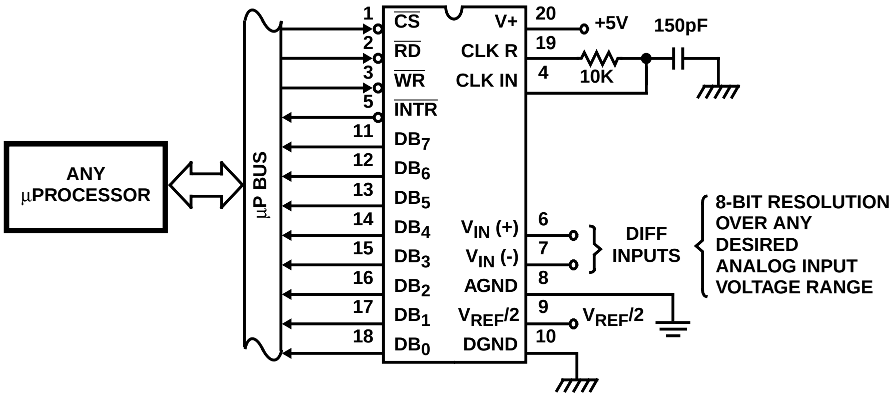
PART NUMBER	ERROR	EXTERNAL CONDITIONS	TEMP. RANGE (^{o}C)	PACKAGE	PKG. NO
ADC0802LCN	±$^1/_2$ LSB	$V_{REF}/2 = 2.500V_{DC}$ (No Adjustments)	0 to 70	20 Ld PDIP	E20.3
ADC0802LCD	±$^3/_4$ LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0802LD	±1 LSB		-55 to 125	20 Ld CERDIP	F20.3
ADC0803LCN	±$^1/_2$ LSB	$V_{REF}/2$ Adjusted for Correct Full Scale Reading	0 to 70	20 Ld PDIP	E20.3
ADC0803LCD	±$^3/_4$ LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0803LCWM	±1 LSB		-40 to 85	20 Ld SOIC	M20.3
ADC0803LD	±1 LSB		-55 to 125	20 Ld CERDIP	F20.3
ADC0804LCN	±1 LSB	$V_{REF}/2 = 2.500V_{DC}$ (No Adjustments)	0 to 70	20 Ld PDIP	E20.3
ADC0804LCD	±1 LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0804LCWM	±1 LSB		-40 to 85	20 Ld SOIC	M20.3

Pinout

ADC0802, ADC0803, ADC0804
(PDIP, CERDIP)
TOP VIEW

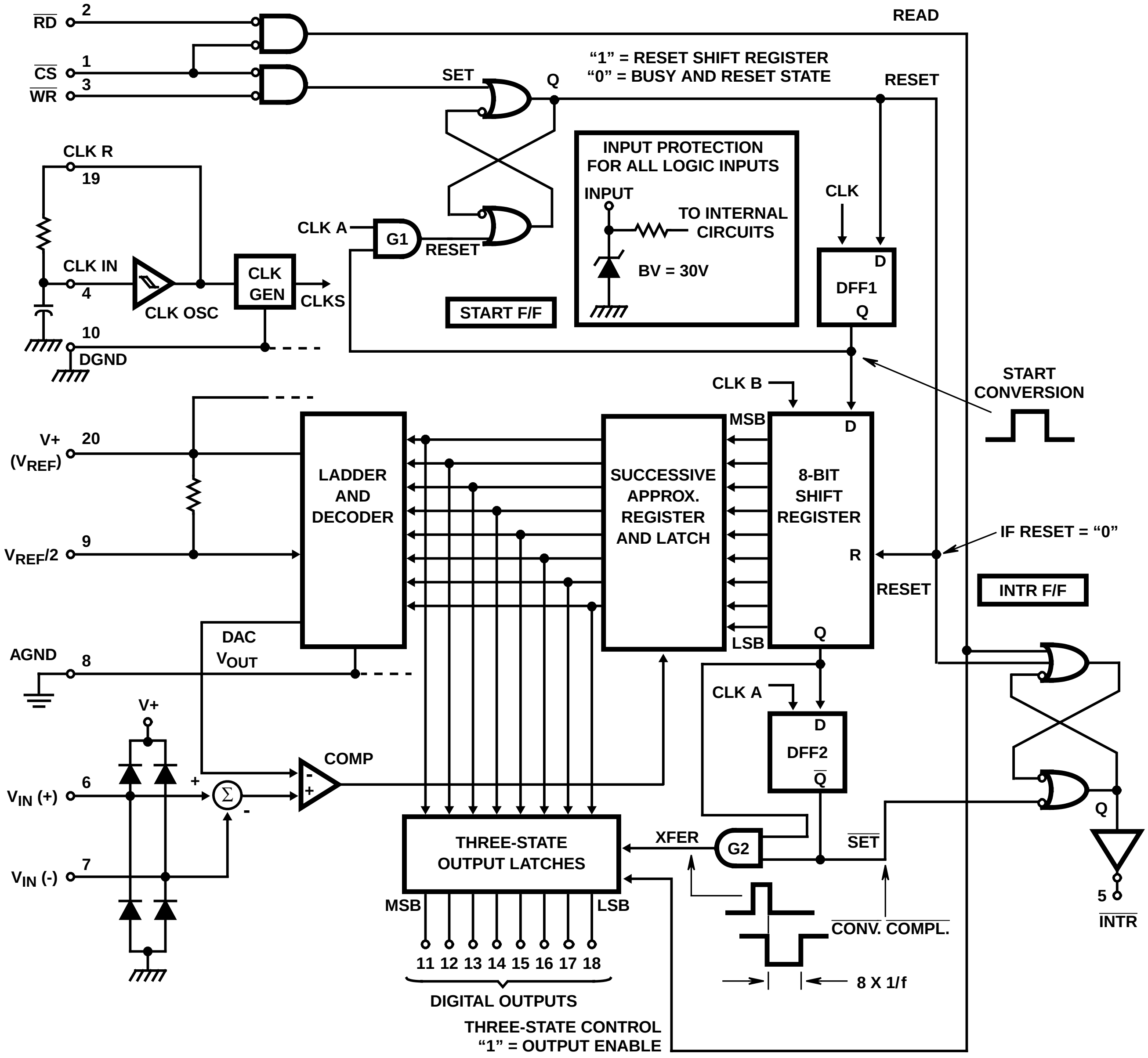


Typical Application Schematic



CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper IC Handling Procedures.

Functional Diagram



Absolute Maximum Ratings

Supply Voltage	6.5V
Voltage at Any Input	-0.3V to (V$^+$ +0.3V)

Operating Conditions

Temperature Range	
ADC0802/03LD	-55^{o}C to 125^{o}C
ADC0802/03/04LCD	-40^{o}C to 85^{o}C
ADC0802/03/04LCN	0^{o}C to 70^{o}C
ADC0803/04LCWM	-40^{o}C to 85^{o}C

Thermal Information

Thermal Resistance (Typical, Note 1)	θ_{JA} (^{o}C/W)	θ_{JC} (^{o}C/W)
PDIP Package	125	N/A
CERDIP Package	80	20
SOIC Package	120	N/A

Maximum Junction Temperature	
Hermetic Package	175^{o}C
Plastic Package	150^{o}C
Maximum Storage Temperature Range	-65^{o}C to 150^{o}C
Maximum Lead Temperature (Soldering, 10s)	300^{o}C
(SOIC - Lead Tips Only)	

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

1. θ_{JA} is measured with the component mounted on an evaluation PC board in free air.

Electrical Specifications (Notes 1, 7)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
CONVERTER SPECIFICATIONS V+ = 5V, T_A = 25^{o}C and f_{CLK} = 640kHz, Unless Otherwise Specified					
Total Unadjusted Error					
ADC0802	V_{REF}/2 = 2.500V	-	-	$\pm{}^1/_2$	LSB
ADC0803	V_{REF}/2 Adjusted for Correct Full Scale Reading	-	-	$\pm{}^1/_2$	LSB
ADC0804	V_{REF}/2 = 2.500V	-	-	±1	LSB
V_{REF}/2 Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	kΩ
Analog Input Voltage Range	(Note 2)	GND-0.05	-	(V+) + 0.05	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	$\pm{}^1/_{16}$	$\pm{}^1/_8$	LSB
Power Supply Sensitivity	V+ = 5V ±10% Over Allowed Input Voltage Range	-	$\pm{}^1/_{16}$	$\pm{}^1/_8$	LSB
CONVERTER SPECIFICATIONS V+ = 5V, 0^{o}C to 70^{o}C and f_{CLK} = 640kHz, Unless Otherwise Specified					
Total Unadjusted Error					
ADC0802	V_{REF}/2 = 2.500V	-	-	$\pm{}^1/_2$	LSB
ADC0803	V_{REF}/2 Adjusted for Correct Full Scale Reading	-	-	$\pm{}^1/_2$	LSB
ADC0804	V_{REF}/2 = 2.500V	-	-	±1	LSB
V_{REF}/2 Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	kΩ
Analog Input Voltage Range	(Note 2)	GND-0.05	-	(V+) + 0.05	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	$\pm{}^1/_8$	$\pm{}^1/_4$	LSB
Power Supply Sensitivity	V+ = 5V ±10% Over Allowed Input Voltage Range	-	$\pm{}^1/_{16}$	$\pm{}^1/_8$	LSB
CONVERTER SPECIFICATIONS V+ = 5V, -25^{o}C to 85^{o}C and f_{CLK} = 640kHz, Unless Otherwise Specified					
Total Unadjusted Error					
ADC0802	V_{REF}/2 = 2.500V	-	-	$\pm{}^3/_4$	LSB
ADC0803	V_{REF}/2 Adjusted for Correct Full Scale Reading	-	-	$\pm{}^3/_4$	LSB
ADC0804	V_{REF}/2 = 2.500V	-	-	±1	LSB
V_{REF}/2 Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	kΩ
Analog Input Voltage Range	(Note 2)	GND-0.05	-	(V+) + 0.05	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	$\pm{}^1/_8$	$\pm{}^1/_4$	LSB
Power Supply Sensitivity	V+ = 5V ±10% Over Allowed Input Voltage Range	-	$\pm{}^1/_{16}$	$\pm{}^1/_8$	LSB

Electrical Specifications (Notes 1, 7) **(Continued)**

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
CONVERTER SPECIFICATIONS V+ = 5V, -55^{o}C to 125^{o}C and f_{CLK} = 640kHz, Unless Otherwise Specified					
Total Unadjusted Error					
ADC0802	V_{REF}/2 = 2.500V	-	-	±1	LSB
ADC0803	V_{REF}/2 Adjusted for Correct Full Scale Reading	-	-	±1	LSB
V_{REF}/2 Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	kΩ
Analog Input Voltage Range	(Note 2)	GND-0.05	-	(V+) + 0.05	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	$\pm{}^{1}/_{8}$	$\pm{}^{1}/_{4}$	LSB
Power Supply Sensitivity	V+ = 5V ±10% Over Allowed Input Voltage Range	-	$\pm{}^{1}/_{8}$	$\pm{}^{1}/_{4}$	LSB
AC TIMING SPECIFICATIONS V+ = 5V, and T_A = 25^{o}C, Unless Otherwise Specified					
Clock Frequency, f_{CLK}	V+ = 6V (Note 3)	100	640	1280	kHz
	V+ = 5V	100	640	800	kHz
Clock Periods per Conversion (Note 4), t_{CONV}		62	-	73	Clocks/Conv
Conversion Rate In Free-Running Mode, CR	$\overline{\text{INTR}}$ tied to $\overline{\text{WR}}$ with $\overline{\text{CS}}$ = 0V, f_{CLK} = 640kHz	-	-	8888	Conv/s
Width of $\overline{\text{WR}}$ Input (Start Pulse Width), $t_{W(WR)I}$	$\overline{\text{CS}}$ = 0V (Note 5)	100	-	-	ns
Access Time (Delay from Falling Edge of $\overline{\text{RD}}$ to Output Data Valid), t_{ACC}	C_L = 100pF (Use Bus Driver IC for Larger $C_{L)}$	-	135	200	ns
Three-State Control (Delay from Rising Edge of $\overline{\text{RD}}$ to HI-Z State), t_{1H}, t_{0H}	C_L = 10pF, R_L= 10K (See Three-State Test Circuits)	-	125	250	ns
Delay from Falling Edge of $\overline{\text{WR}}$ to Reset of $\overline{\text{INTR}}$, t_{WI}, t_{RI}		-	300	450	ns
Input Capacitance of Logic Control Inputs, C_{IN}		-	5	-	pF
Three-State Output Capacitance (Data Buffers), C_{OUT}		-	5	-	pF
DC DIGITAL LEVELS AND DC SPECIFICATIONS V+ = 5V, and T_{MIN} to T_{MAX}, Unless Otherwise Specified					
CONTROL INPUTS (Note 6)					
Logic “1“ Input Voltage (Except Pin 4 CLK IN), V_{INH}	V+ = 5.25V	2.0	-	V+	V
Logic “0“ Input Voltage (Except Pin 4 CLK IN), V_{INL}	V+ = 4.75V	-	-	0.8	V
CLK IN (Pin 4) Positive Going Threshold Voltage, $V+_{CLK}$		2.7	3.1	3.5	V
CLK IN (Pin 4) Negative Going Threshold Voltage, $V-_{CLK}$		1.5	1.8	2.1	V
CLK IN (Pin 4) Hysteresis, V_H		0.6	1.3	2.0	V
Logic “1” Input Current (All Inputs), I_{INHI}	V_{IN} = 5V	-	0.005	1	µA
Logic “0” Input Current (All Inputs), I_{INLO}	V_{IN} = 0V	-1	-0.005	-	µA
Supply Current (Includes Ladder Current), I+	f_{CLK} = 640kHz,T_A = 25^{o}C and $\overline{\text{CS}}$ = HI	-	1.3	2.5	mA
DATA OUTPUTS AND $\overline{\text{INTR}}$					
Logic “0” Output Voltage, V_{OL}	I_O = 1.6mA, V+ = 4.75V	-	-	0.4	V

Electrical Specifications (Notes 1, 7) **(Continued)**

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
Logic “1” Output Voltage, V_{OH}	I_O = -360µA, V+ = 4.75V	2.4	-	-	V
Three-State Disabled Output Leakage (All Data Buffers), I_{LO}	V_{OUT} = 0V	-3	-	-	µA
	V_{OUT} = 5V	-	-	3	µA
Output Short Circuit Current, I_{SOURCE}	V_{OUT} Short to Gnd T_A = 25^{o}C	4.5	6	-	mA
Output Short Circuit Current, I_{SINK}	V_{OUT} Short to V+ T_A = 25^{o}C	9.0	16	-	mA

NOTES:

1. All voltages are measured with respect to GND, unless otherwise specified. The separate AGND point should always be wired to the DGND, being careful to avoid ground loops.
2. For $V_{IN(-)} \geq V_{IN(+)}$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see Block Diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V+ supply. Be careful, during testing at low V+ levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct - especially at elevated temperatures, and cause errors for analog inputs near full scale. As long as the analog V_{IN} does not exceed the supply voltage by more than 50mV, the output code will be correct. To achieve an absolute 0V to 5V input voltage range will therefore require a minimum supply voltage of 4.950V over temperature variations, initial tolerance and loading.
3. With V+ = 6V, the digital logic interfaces are no longer TTL compatible.
4. With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process.
5. The $\overline{CS}$ input is assumed to bracket the $\overline{WR}$ strobe input so that timing is dependent on the $\overline{WR}$ pulse width. An arbitrarily wide pulse width will hold the converter in a reset mode and the start of conversion is initiated by the low to high transition of the $\overline{WR}$ pulse (see Timing Diagrams).
6. CLK IN (pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately.
7. None of these A/Ds requires a zero-adjust. However, if an all zero code is desired for an analog input other than 0V, or if a narrow full scale span exists (for example: 0.5V to 4V full scale) the $V_{IN(-)}$ input can be adjusted to achieve this. See the Zero Error description in this data sheet.

Timing Waveforms

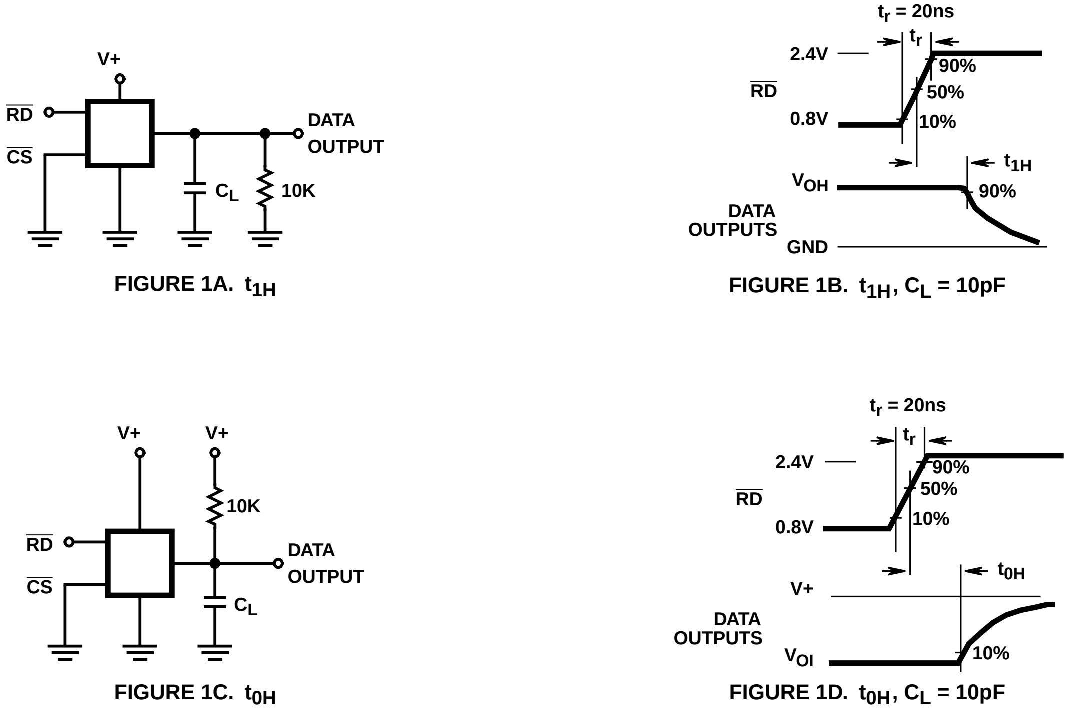


FIGURE 1A. t_{1H}

FIGURE 1B. t_{1H}, C_L = 10pF

FIGURE 1C. t_{0H}

FIGURE 1D. t_{0H}, C_L = 10pF

FIGURE 1. THREE-STATE CIRCUITS AND WAVEFORMS

Typical Performance Curves

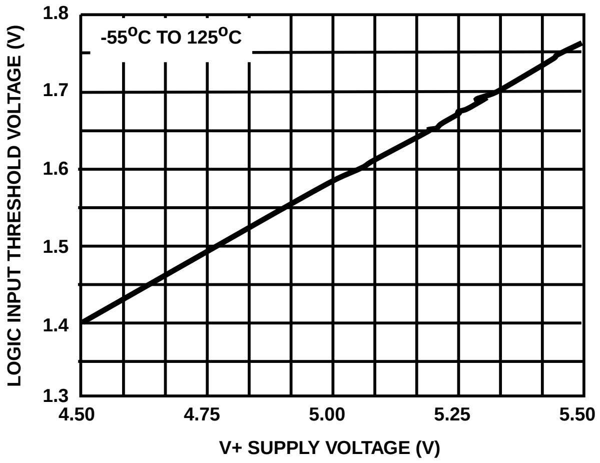


FIGURE 2. LOGIC INPUT THRESHOLD VOLTAGE vs SUPPLY VOLTAGE

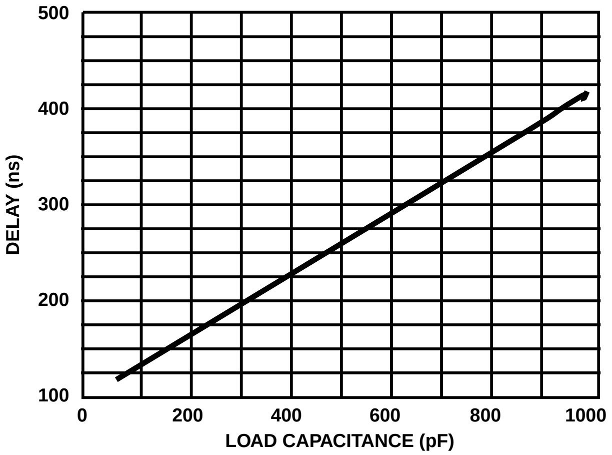


FIGURE 3. DELAY FROM FALLING EDGE OF $\overline{RD}$ TO OUTPUT DATA VALID vs LOAD CAPACITANCE

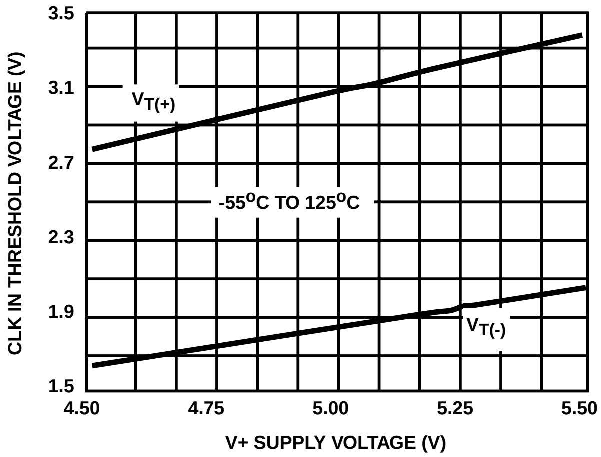


FIGURE 4. CLK IN SCHMITT TRIP LEVELS vs SUPPLY VOLTAGE

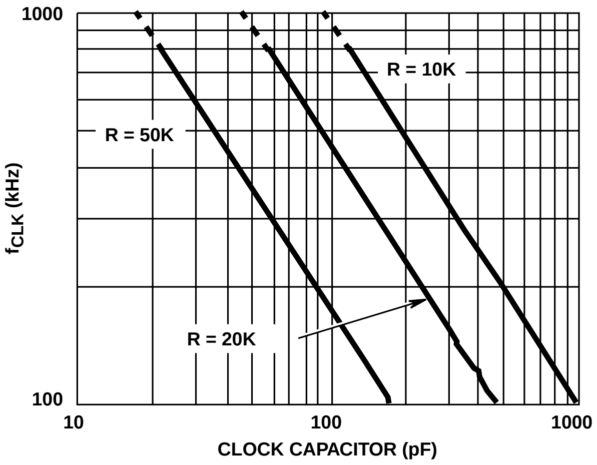


FIGURE 5. f_{CLK} vs CLOCK CAPACITOR

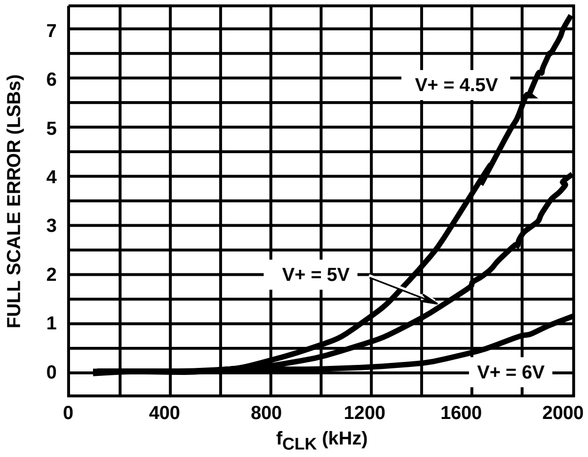


FIGURE 6. FULL SCALE ERROR vs f_{CLK}

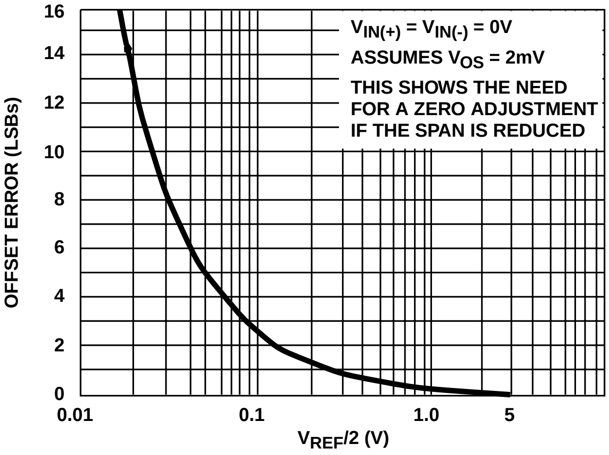


FIGURE 7. EFFECT OF UNADJUSTED OFFSET ERROR

Typical Performance Curves (Continued)

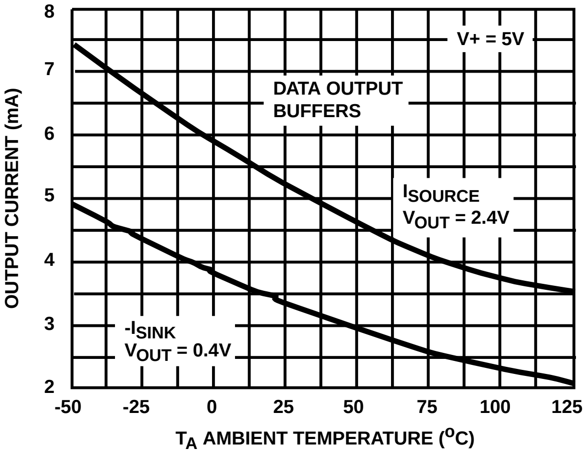


FIGURE 8. OUTPUT CURRENT vs TEMPERATURE

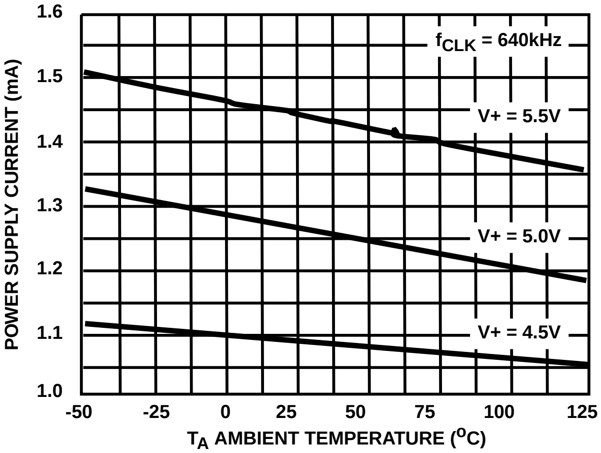


FIGURE 9. POWER SUPPLY CURRENT vs TEMPERATURE

Timing Diagrams

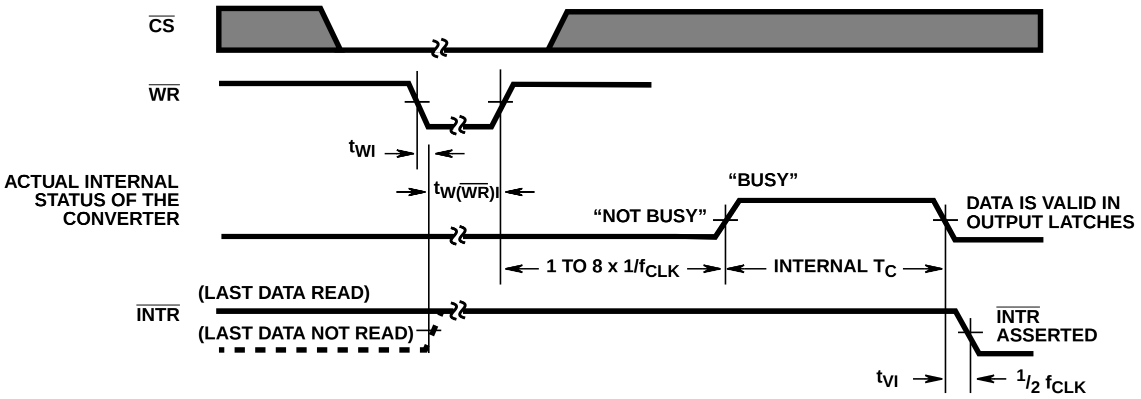


FIGURE 10A. START CONVERSION

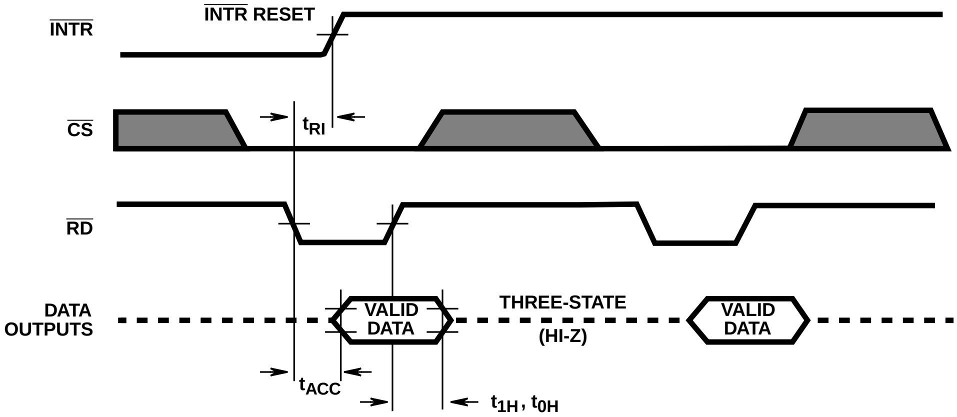
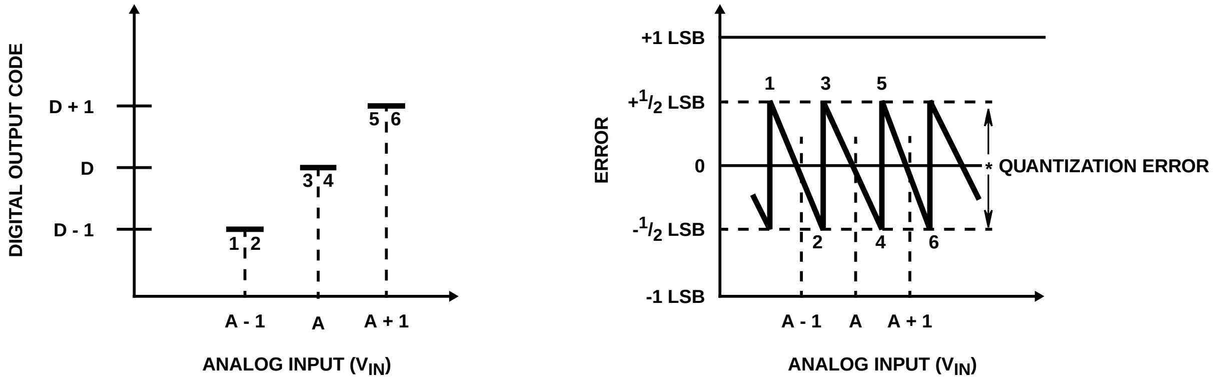


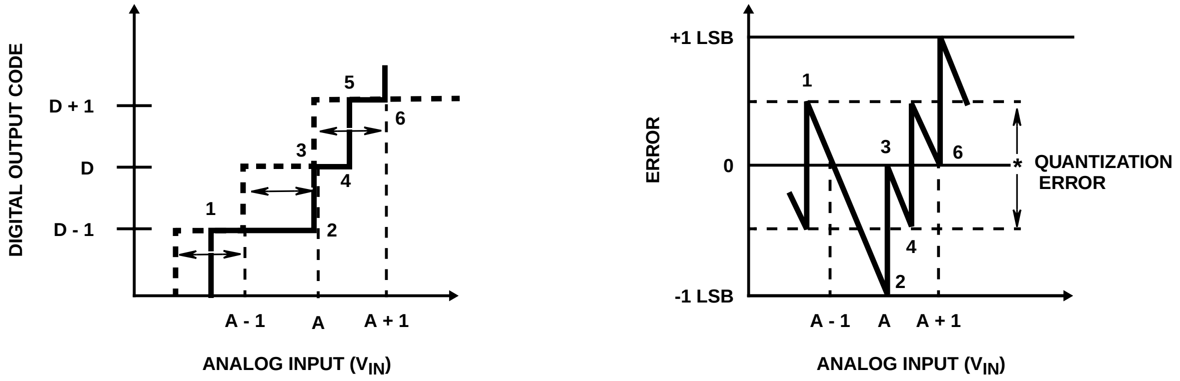
FIGURE 10B. OUTPUT ENABLE AND RESET $\overline{INTR}$



TRANSFER FUNCTION

ERROR PLOT

FIGURE 11A. ACCURACY = ±0 LSB; PERFECT A/D



TRANSFER FUNCTION

ERROR PLOT

FIGURE 11B. ACCURACY = ±$^1/_2$ LSB

FIGURE 11. CLARIFYING THE ERROR SPECS OF AN A/D CONVERTER

Understanding A/D Error Specs

A perfect A/D transfer characteristic (staircase wave-form) is shown in Figure 11A. The horizontal scale is analog input voltage and the particular points labeled are in steps of 1 LSB (19.53mV with 2.5V tied to the V_{REF}/2 pin). The digital output codes which correspond to these inputs are shown as D-1, D, and D+1. For the perfect A/D, not only will center-value (A - 1, A, A + 1, . . .) analog inputs produce the correct output digital codes, but also each riser (the transitions between adjacent output codes) will be located ±$^1/_2$ LSB away from each center-value. As shown, the risers are ideal and have no width. Correct digital output codes will be provided for a range of analog input voltages which extend ±$^1/_2$ LSB from the ideal center-values. Each tread (the range of analog input voltage which provides the same digital output code) is therefore 1 LSB wide.

The error curve of Figure 11B shows the worst case transfer function for the ADC0802. Here the specification guarantees that if we apply an analog input equal to the LSB analog voltage center-value, the A/D will produce the correct digital code.

Next to each transfer function is shown the corresponding error plot. Notice that the error includes the quantization uncertainty of the A/D. For example, the error at point 1 of Figure 11A is +$^1/_2$ LSB because the digital code appeared $^1/_2$ LSB in advance of the center-value of the tread. The error plots always have a constant negative slope and the abrupt upside steps are always 1 LSB in magnitude, unless the device has missing codes.

Detailed Description

The functional diagram of the ADC0802 series of A/D converters operates on the successive approximation principle (see Application Notes AN016 and AN020 for a more detailed description of this principle). Analog switches are closed sequentially by successive-approximation logic until the analog differential input voltage [$V_{IN(+)}$ - $V_{IN(-)}$] matches a voltage derived from a tapped resistor string across the reference voltage. The most significant bit is tested first and after 8 comparisons (64 clock cycles), an 8-bit binary code (1111 1111 = full scale) is transferred to an output latch.

The normal operation proceeds as follows. On the high-to-low transition of the WR input, the internal SAR latches and the shift-register stages are reset, and the INTR output will be set high. As long as the CS input and WR input remain low, the A/D will remain in a reset state. Conversion will start from 1 to 8 clock periods after at least one of these inputs makes a low-to-high transition. After the requisite number of clock pulses to complete the conversion, the INTR pin will make a high-to-low transition. This can be used to interrupt a processor, or otherwise signal the availability of a new conversion. A RD operation (with CS low) will clear the INTR line high again.

The device may be operated in the free-running mode by connecting INTR to the WR input with CS = 0. To ensure start-up under all possible conditions, an external WR pulse is required during the first power-up cycle. A conversion-in-process can be interrupted by issuing a second start command.

Digital Operation

The converter is started by having CS and WR simultaneously low. This sets the start flip-flop (F/F) and the resulting “1” level resets the 8-bit shift register, resets the Interrupt (INTR) F/F and inputs a “1” to the D flip-flop, DFF1, which is at the input end of the 8-bit shift register. Internal clock signals then transfer this “1” to the Q output of DFF1. The AND gate, G1, combines this “1” output with a clock signal to provide a reset signal to the start F/F. If the set signal is no longer present (either WR or CS is a “1”), the start F/F is reset and the 8-bit shift register then can have the “1” clocked in, which starts the conversion process. If the set signal were to still be present, this reset pulse would have no effect (both outputs of the start F/F would be at a “1” level) and the 8-bit shift register would continue to be held in the reset mode. This allows for asynchronous or wide CS and WR signals.

After the “1” is clocked through the 8-bit shift register (which completes the SAR operation) it appears as the input to DFF2. As soon as this “1” is output from the shift register, the AND gate, G2, causes the new digital word to transfer to the Three-State output latches. When DFF2 is subsequently clocked, the $\overline{Q}$ output makes a high-to-low transition which causes the INTR F/F to set. An inverting buffer then supplies the $\overline{INTR}$ output signal.

When data is to be read, the combination of both $\overline{CS}$ and $\overline{RD}$ being low will cause the INTR F/F to be reset and the three-state output latches will be enabled to provide the 8-bit digital outputs.

Digital Control Inputs

The digital control inputs (CS, RD, and WR) meet standard TTL logic voltage levels. These signals are essentially equivalent to the standard A/D Start and Output Enable control signals, and are active low to allow an easy interface to microprocessor control busses. For non-microprocessor based applications, the CS input (pin 1) can be grounded and the standard A/D Start function obtained by an active low pulse at the WR input (pin 3). The Output Enable function is achieved by an active low pulse at the RD input (pin 2).

Analog Operation

The analog comparisons are performed by a capacitive charge summing circuit. Three capacitors (with precise ratioed values) share a common node with the input to an auto-zeroed comparator. The input capacitor is switched between $V_{IN(+)}$ and $V_{IN(-)}$, while two ratioed reference capacitors are switched between taps on the reference voltage divider string. The net charge corresponds to the weighted difference between the input and the current total value set by the successive approximation register. A correction is made to offset the comparison by $^1/_2$ LSB (see Figure 11A).

Analog Differential Voltage Inputs and Common-Mode Rejection

This A/D gains considerable applications flexibility from the analog differential voltage input. The $V_{IN(-)}$ input (pin 7) can be used to automatically subtract a fixed voltage value from the input reading (tare correction). This is also useful in 4mA - 20mA current loop conversion. In addition, common-mode noise can be reduced by use of the differential input.

The time interval between sampling $V_{IN(+)}$ and $V_{IN(-)}$ is $4^1/_2$ clock periods. The maximum error voltage due to this slight time difference between the input voltage samples is given by:

$$\Delta V_E(MAX) = (V_{PEAK})(2\pi f_{CM})\left[\frac{4.5}{f_{CLK}}\right]$$

where:

ΔV_E is the error voltage due to sampling delay,

V_{PEAK} is the peak value of the common-mode voltage,

f_{CM} is the common-mode frequency.

For example, with a 60Hz common-mode frequency, f_{CM}, and a 640kHz A/D clock, f_{CLK}, keeping this error to $^1/_4$ LSB (~5mV) would allow a common-mode voltage, V_{PEAK}, given by:

$$V_{PEAK} = \frac{\left[\Delta V_{E(MAX)(f_{CLK})}\right]}{(2\pi f_{CM})(4.5)} ,$$

or

$$V_{PEAK} = \frac{(5\times10^{-3})(640\times10^{3})}{(6.28)(60)(4.5)} \cong 1.9V.$$

The allowed range of analog input voltage usually places more severe restrictions on input common-mode voltage levels than this.

An analog input voltage with a reduced span and a relatively large zero offset can be easily handled by making use of the differential input (see Reference Voltage Span Adjust).

Analog Input Current

The internal switching action causes displacement currents to flow at the analog inputs. The voltage on the on-chip capacitance to ground is switched through the analog differential input voltage, resulting in proportional currents entering the $V_{IN(+)}$ input and leaving the $V_{IN(-)}$ input. These current transients occur at the leading edge of the internal clocks. They rapidly decay and do not inherently cause errors as the on-chip comparator is strobed at the end of the clock perIod.

Input Bypass Capacitors

Bypass capacitors at the inputs will average these charges and cause a DC current to flow through the output resistances of the analog signal sources. This charge pumping action is worse for continuous conversions with the $V_{IN(+)}$ input voltage at full scale. For a 640kHz clock frequency with the $V_{IN(+)}$ input at 5V, this DC current is at a maximum of approximately 5µA. Therefore, **bypass capacitors should not be used at the analog inputs or the V_{REF}/2** pin for high resistance sources (>1kΩ). If input bypass capacitors are necessary for noise filtering and high source resistance is desirable to minimize capacitor size, the effects of the voltage drop across this input resistance, due to the average value of the input current, can be compensated by a full scale adjustment while the given source resistor and input bypass capacitor are both in place. This is possible because the average value of the input current is a precise linear function of the differential input voltage at a constant conversion rate.

Input Source Resistance

Large values of source resistance where an input bypass capacitor is not used will not cause errors since the input currents settle out prior to the comparison time. If a low-pass filter is required in the system, use a low-value series resistor (≤1kΩ) for a passive RC section or add an op amp RC active low-pass filter. For low-source-resistance applications (≤1kΩ), a 0.1μF bypass capacitor at the inputs will minimize EMI due to the series lead inductance of a long wire. A 100Ω series resistor can be used to isolate this capacitor (both the R and C are placed outside the feedback loop) from the output of an op amp, if used.

Stray Pickup

The leads to the analog inputs (pins 6 and 7) should be kept as short as possible to minimize stray signal pickup (EMI). Both EMI and undesired digital-clock coupling to these inputs can cause system errors. The source resistance for these inputs should, in general, be kept below 5kΩ. Larger values of source resistance can cause undesired signal pickup. Input bypass capacitors, placed from the analog inputs to ground, will eliminate this pickup but can create analog scale errors as these capacitors will average the transient input switching currents of the A/D (see Analog Input Current). This scale error depends on both a large source resistance and the use of an input bypass capacitor. This error can be compensated by a full scale adjustment of the A/D (see Full Scale Adjustment) with the source resistance and input bypass capacitor in place, and the desired conversion rate.

Reference Voltage Span Adjust

For maximum application flexibility, these A/Ds have been designed to accommodate a 5V, 2.5V or an adjusted voltage reference. This has been achieved in the design of the IC as shown in Figure 12.

Notice that the reference voltage for the IC is either $^1/_2$ of the voltage which is applied to the V+ supply pin, or is equal to the voltage which is externally forced at the $V_{REF}/2$ pin. This allows for a pseudo-ratiometric voltage reference using, for the V+ supply, a 5V reference voltage. Alternatively, a voltage less than 2.5V can be applied to the $V_{REF}/2$ input. The internal gain to the $V_{REF}/2$ input is 2 to allow this factor of 2 reduction in the reference voltage.

Such an adjusted reference voltage can accommodate a reduced span or dynamic voltage range of the analog input voltage. If the analog input voltage were to range from 0.5V to 3.5V, instead of 0V to 5V, the span would be 3V. With 0.5V applied to the $V_{IN(-)}$ pin to absorb the offset, the reference voltage can be made equal to $^1/_2$ of the 3V span or 1.5V. The A/D now will encode the $V_{IN(+)}$ signal from 0.5V to 3.5V with the 0.5V input corresponding to zero and the 3.5V input corresponding to full scale. The full 8 bits of resolution are therefore applied over this reduced analog input voltage range. The requisite connections are shown in Figure 13. For expanded scale inputs, the circuits of Figures 14 and 15 can be used.

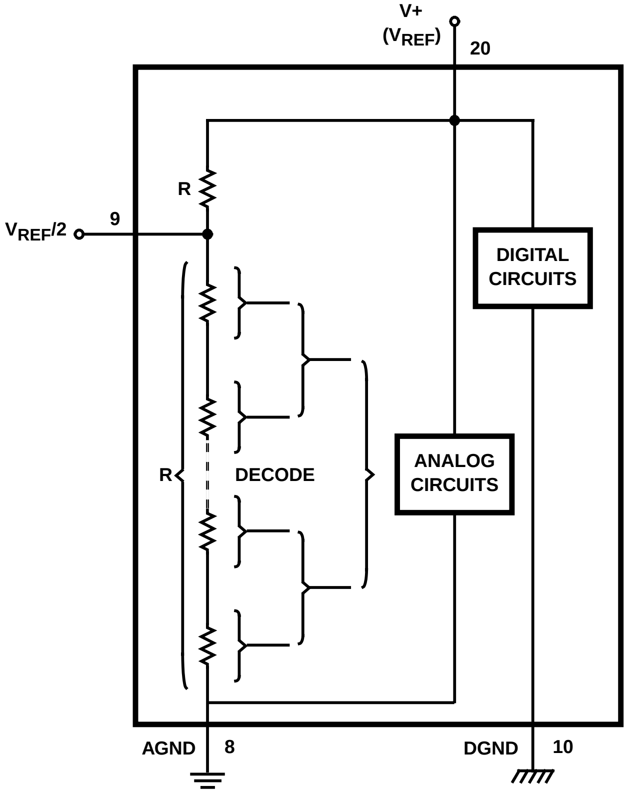


FIGURE 12. THE $V_{REFERENCE}$ DESIGN ON THE IC

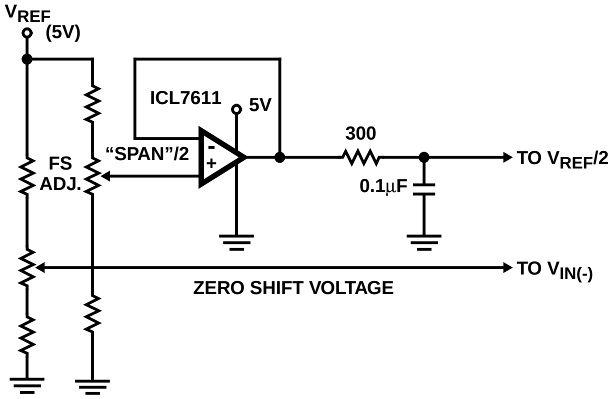


FIGURE 13. OFFSETTING THE ZERO OF THE ADC0802 AND PERFORMING AN INPUT RANGE (SPAN) ADJUSTMENT

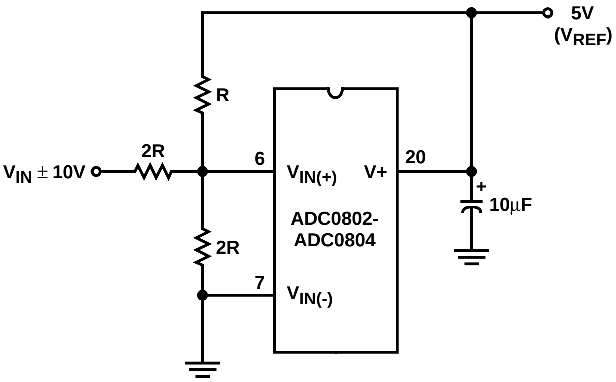


FIGURE 14. HANDLING ±10V ANALOG INPUT RANGE

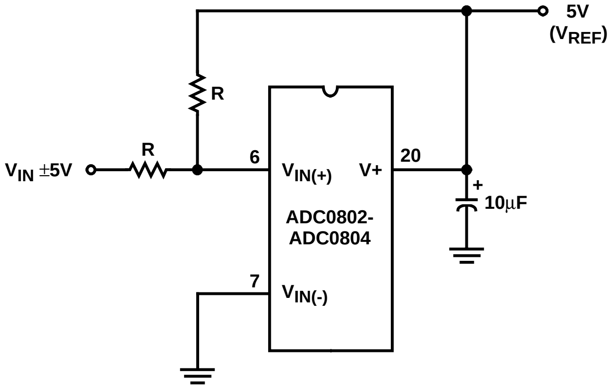


FIGURE 15. HANDLING ±5V ANALOG INPUT RANGE

Reference Accuracy Requirements

The converter can be operated in a pseudo-ratiometric mode or an absolute mode. In ratiometric converter applications, the magnitude of the reference voltage is a factor in both the output of the source transducer and the output of the A/D converter and therefore cancels out in the final digital output code. In absolute conversion applicatIons, both the initial value and the temperature stability of the reference voltage are important accuracy factors in the operation of the A/D converter. For $V_{REF}/2$ voltages of 2.5V nominal value, initial errors of ±10mV will cause conversion errors of ±1 LSB due to the gain of 2 of the $V_{REF}/2$ input. In reduced span applications, the initial value and the stability of the $V_{REF}/2$ input voltage become even more important. For example, if the span is reduced to 2.5V, the analog input LSB voltage value is correspondingly reduced from 20mV (5V span) to 10mV and 1 LSB at the $V_{REF}/2$ input becomes 5mV. As can be seen, this reduces the allowed initial tolerance of the reference voltage and requires correspondingly less absolute change with temperature variations. Note that spans smaller than 2.5V place even tighter requirements on the initial accuracy and stability of the reference source.

In general, the reference voltage will require an initial adjustment. Errors due to an improper value of reference voltage appear as full scale errors in the A/D transfer function. IC voltage regulators may be used for references if the ambient temperature changes are not excessive.

Zero Error

The zero of the A/D does not require adjustment. If the minimum analog input voltage value, $V_{IN(MIN)}$, is not ground, a zero offset can be done. The converter can be made to output 0000 0000 digital code for this minimum input voltage by biasing the A/D $V_{IN(-)}$ input at this $V_{IN(MIN)}$ value (see Applications section). This utilizes the differential mode operation of the A/D.

The zero error of the A/D converter relates to the location of the first riser of the transfer function and can be measured by grounding the $V_{IN(-)}$ input and applying a small magnitude positive voltage to the $V_{IN(+)}$ input. Zero error is the difference between the actual DC input voltage which is necessary to just cause an output digital code transition from 0000 0000 to 0000 0001 and the ideal $^1/_2$ LSB value ($^1/_2$ LSB = 9.8mV for $V_{REF}/2$ = 2.500V).

Full Scale Adjust

The full scale adjustment can be made by applying a differential input voltage which is $1^1/_2$ LSB down from the desired analog full scale voltage range and then adjusting the magnitude of the $V_{REF}/2$ input (pin 9) for a digital output code which is just changing from 1111 1110 to 1111 1111. When offsetting the zero and using a span-adjusted $V_{REF}/2$ voltage, the full scale adjustment is made by inputting V_{MIN} to the $V_{IN(-)}$ input of the A/D and applying a voltage to the $V_{IN(+)}$ input which is given by:

$$V_{IN(+)}f_{SADJ} = V_{MAX} - 1.5\left[\frac{(V_{MAX} - V_{MIN})}{256}\right],$$

where:

V_{MAX} = the high end of the analog input range,

and

V_{MIN} = the low end (the offset zero) of the analog range. (Both are ground referenced.)

Clocking Option

The clock for the A/D can be derived from an external source such as the CPU clock or an external RC network can be added to provIde self-clocking. The CLK IN (pin 4) makes use of a Schmitt trigger as shown in Figure 16.

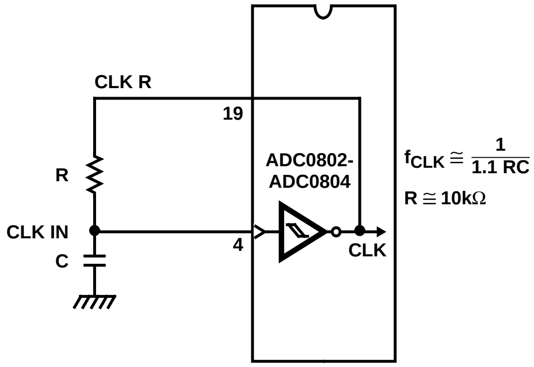


FIGURE 16. SELF-CLOCKING THE A/D

Heavy capacitive or DC loading of the CLK R pin should be avoided as this will disturb normal converter operation. Loads less than 50pF, such as driving up to 7 A/D converter clock inputs from a single CLK R pin of 1 converter, are allowed. For larger clock line loading, a CMOS or low power TTL buffer or PNP input logic should be used to minimize the loading on the CLK R pin (do not use a standard TTL buffer).

Restart During a Conversion

If the A/D is restarted ($\overline{CS}$ and $\overline{WR}$ go low and return high) during a conversion, the converter is reset and a new conversion is started. The output data latch is not updated if the conversion in progress is not completed. The data from the previous conversion remain in this latch.

Continuous Conversions

In this application, the $\overline{CS}$ input is grounded and the $\overline{WR}$ input is tied to the $\overline{INTR}$ output. This $\overline{WR}$ and $\overline{INTR}$ node should be momentarily forced to logic low following a power-up cycle to insure circuit operation. See Figure 17 for details.

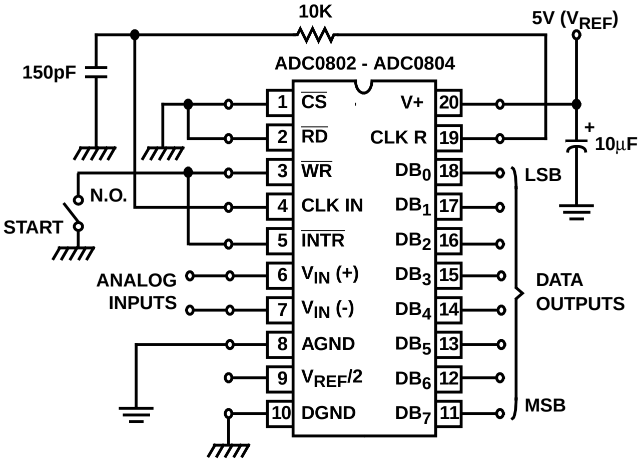


FIGURE 17. FREE-RUNNING CONNECTION

Driving the Data Bus

This CMOS A/D, like MOS microprocessors and memories, will require a bus driver when the total capacitance of the data bus gets large. Other circuItry, which is tied to the data bus, will add to the total capacitive loading, even in three-state (high-impedance mode). Back plane busing also greatly adds to the stray capacitance of the data bus.

There are some alternatives available to the designer to handle this problem. Basically, the capacitive loading of the data bus slows down the response time, even though DC specifications are still met. For systems operating with a relatively slow CPU clock frequency, more time is available in which to establish proper logic levels on the bus and therefore higher capacitive loads can be driven (see Typical Performance Curves).

At higher CPU clock frequencies time can be extended for I/O reads (and/or writes) by inserting wait states (8080) or using clock-extending circuits (6800).

Finally, if time is short and capacitive loading is high, external bus drivers must be used. These can be three-state buffers (low power Schottky is recommended, such as the 74LS240 series) or special higher-drive-current products which are designed as bus drivers. High-current bipolar bus drivers with PNP inputs are recommended.

Power Supplies

Noise spikes on the V+ supply line can cause conversion errors as the comparator will respond to this noise. A low-inductance tantalum filter capacitor should be used close to the converter V+ pin, and values of 1µF or greater are recommended. If an unregulated voltage is available in the system, a separate 5V voltage regulator for the converter (and other analog circuitry) will greatly reduce digital noise on the V+ supply. An ICL7663 can be used to regulate such a supply from an input as low as 5.2V.

Wiring and Hook-Up Precautions

Standard digital wire-wrap sockets are not satisfactory for breadboarding with this A/D converter. Sockets on PC boards can be used. All logic signal wires and leads should be grouped and kept as far away as possible from the analog signal leads. Exposed leads to the analog inputs can cause undesired digital noise and hum pickup; therefore, shielded leads may be necessary in many applications.

A single-point analog ground should be used which is separate from the logic ground points. The power supply bypass capacitor and the self-clockIng capacitor (if used) should both be returned to digital ground. Any V_{REF}/2 bypass capacitors, analog input filter capacitors, or input signal shielding should be returned to the analog ground point. A test for proper grounding is to measure the zero error of the A/D converter. Zero errors in excess of $^1/_4$ LSB can usually be traced to improper board layout and wiring (see Zero Error for measurement). Further information can be found in Application Note AN018.

Testing the A/D Converter

There are many degrees of complexity associated with testing an A/D converter. One of the simplest tests is to apply a known analog input voltage to the converter and use LEDs to display the resulting digital output code as shown in Figure 18.

For ease of testing, the V_{REF}/2 (pin 9) should be supplied with 2.560V and a V+ supply voltage of 5.12V should be used. This provides an LSB value of 20mV.

If a full scale adjustment is to be made, an analog input voltage of 5.090V (5.120 - $1^1/_2$ LSB) should be applied to the $V_{IN(+)}$ pin with the $V_{IN(-)}$ pin grounded. The value of the V_{REF}/2 input voltage should be adjusted until the digital output code is just changing from 1111 1110 to 1111 1111. This value of V_{REF}/2 should then be used for all the tests.

The digital-output LED display can be decoded by dividing the 8 bits into 2 hex characters, one with the 4 most-significant bits (MS) and one with the 4 least-significant bits (LS). The output is then interpreted as a sum of fractions times the full scale voltage:

$$V_{OUT} = \left(\frac{MS}{16} + \frac{LS}{256}\right)(5.12)V\ .$$

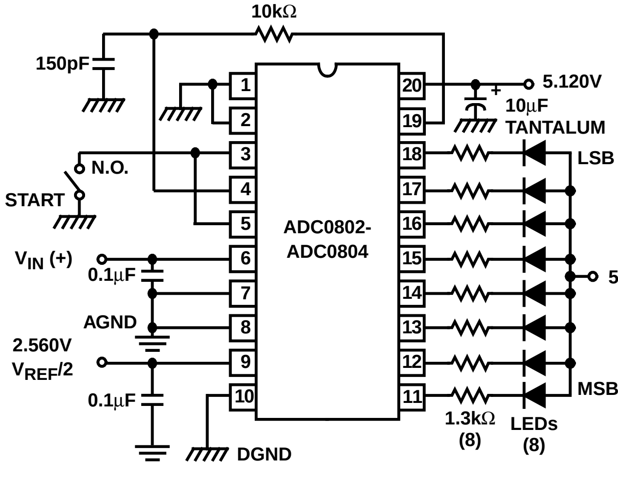


FIGURE 18. BASIC TESTER FOR THE A/D

For example, for an output LED display of 1011 0110, the MS character is hex B (decimal 11) and the LS character is hex (and decimal) 6, so:

$$V_{OUT} = \left(\frac{11}{16} + \frac{6}{256}\right)(5.12) = 3.64V.$$

Figures 19 and 20 show more sophisticated test circuits.

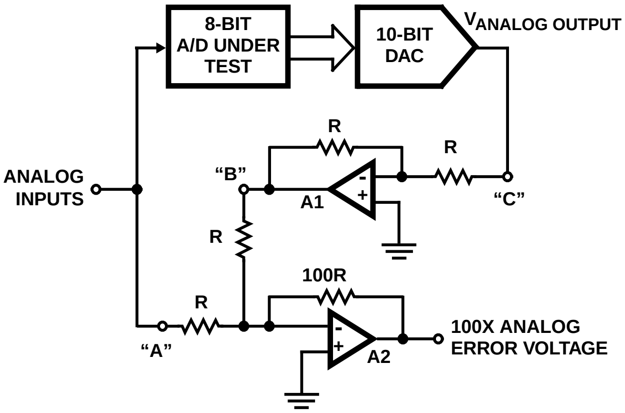


FIGURE 19. A/D TESTER WITH ANALOG ERROR OUTPUT. THIS CIRCUIT CAN BE USED TO GENERATE "ERROR PLOTS" OF FIGURE 11.

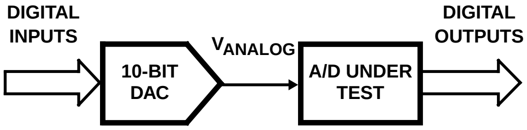


FIGURE 20. BASIC "DIGITAL" A/D TESTER

Typical Applications

Interfacing 8080/85 or Z-80 Microprocessors

This converter has been designed to directly interface with 8080/85 or Z-80 Microprocessors. The three-state output capability of the A/D eliminates the need for a peripheral interface device, although address decoding is still required to generate the appropriate $\overline{CS}$ for the converter. The A/D can be mapped into memory space (using standard memory-address decoding for $\overline{CS}$ and the $\overline{MEMR}$ and $\overline{MEMW}$ strobes) or it can be controlled as an I/O device by using the $\overline{I/OR}$ and $\overline{I/OW}$ strobes and decoding the address bits A0 → A7 (or address bits A8 → A15, since they will contain the same 8-bit address information) to obtain the $\overline{CS}$ input. Using the I/O space provides 256 additional addresses and may allow a simpler 8-bit address decoder, but the data can only be input to the accumulator. To make use of the additional memory reference instructions, the A/D should be mapped into memory space. See AN020 for more discussion of memory-mapped vs I/O-mapped interfaces. An example of an A/D in I/O space is shown in Figure 21.

The standard control-bus signals of the 8080 ($\overline{CS}$, $\overline{RD}$ and $\overline{WR}$) can be directly wired to the digital control inputs of the A/D, since the bus timing requirements, to allow both starting the converter, and outputting the data onto the data bus, are met. A bus driver should be used for larger microprocessor systems where the data bus leaves the PC board and/or must drive capacitive loads larger than 100pF.

It is useful to note that in systems where the A/D converter is 1 of 8 or fewer I/O-mapped devices, no address-decoding circuitry is necessary. Each of the 8 address bits (A0 to A7) can be directly used as $\overline{CS}$ inputs, one for each I/O device.

Interfacing the Z-80 and 8085

The Z-80 and 8085 control buses are slightly different from that of the 8080. General $\overline{RD}$ and $\overline{WR}$ strobes are provided and separate memory request, $\overline{MREQ}$, and I/O request, $\overline{IORQ}$, signals have to be combined with the generalized strobes to provide the appropriate signals. An advantage of operating the A/D in I/O space with the Z-80 is that the CPU will automatically insert one wait state (the $\overline{RD}$ and $\overline{WR}$ strobes are extended one clock period) to allow more time for the I/O devices to respond. Logic to map the A/D in I/O space is shown in Figure 22. By using $\overline{MREQ}$ in place of $\overline{IORQ}$, a memory-mapped configuration results.

Additional I/O advantages exist as software DMA routines are available and use can be made of the output data transfer which exists on the upper 8 address lines (A8 to A15) during I/O input instructions. For example, MUX channel selection for the A/D can be accomplished with this operating mode.

The 8085 also provides a generalized $\overline{RD}$ and $\overline{WR}$ strobe, with an IO/$\overline{M}$ line to distinguish I/O and memory requests. The circuit of Figure 22 can again be used, with IO/$\overline{M}$ in place of $\overline{IORQ}$ for a memory-mapped interface, and an extra inverter (or the logic equivalent) to provide $\overline{IO}$/M for an I/O-mapped connection.

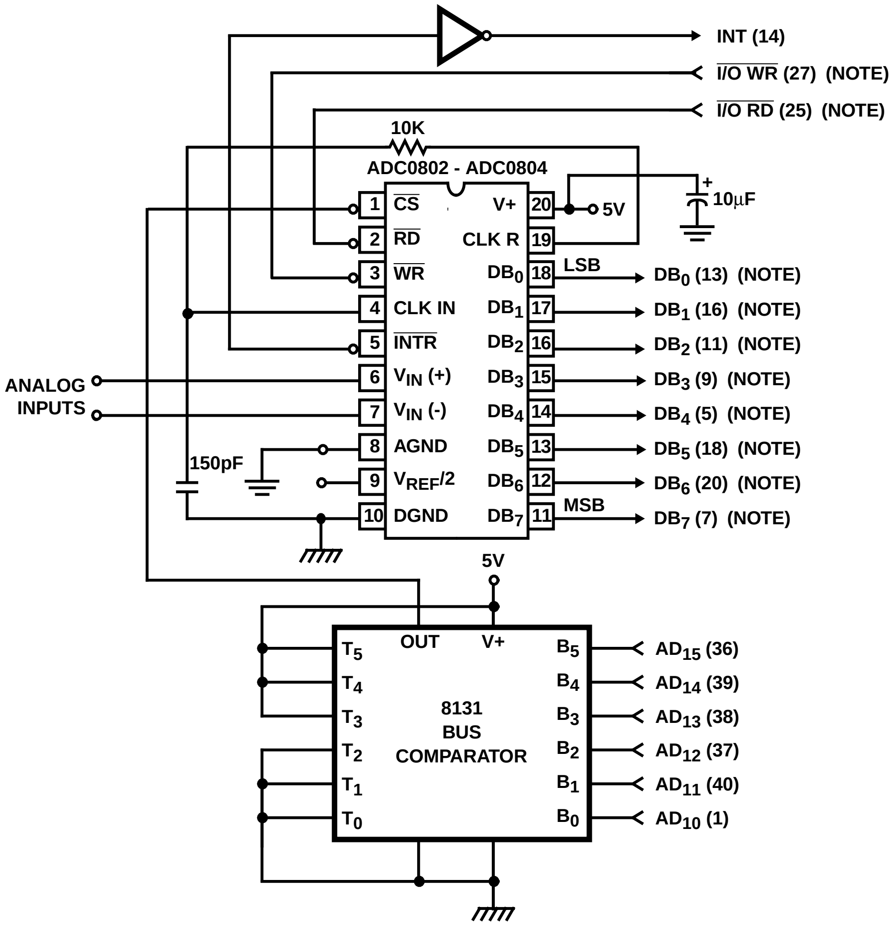
Interfacing 6800 Microprocessor Derivatives (6502, etc.)

The control bus for the 6800 microprocessor derivatives does not use the RD and WR strobe signals. Instead it employs a single R/W line and additional timing, if needed, can be derived from the ϕ2 clock. All I/O devices are memory-mapped in the 6800 system, and a special signal, VMA, indicates that the current address is valid. Figure 23 shows an interface schematic where the A/D is memory-mapped in the 6800 system. For simplicity, the CS decoding is shown using $^1/_2$ DM8092. Note that in many 6800 systems, an already decoded $\overline{4/5}$ line is brought out to the common bus at pin 21. This can be tied directly to the CS pin of the A/D, provided that no other devices are addressed at HEX ADDR: 4XXX or 5XXX.

In Figure 24 the ADC0802 series is interfaced to the MC6800 microprocessor through (the arbitrarily chosen) Port B of the MC6820 or MC6821 Peripheral Interface Adapter (PIA). Here the CS pin of the A/D is grounded since the PIA is already memory-mapped in the MC6800 system and no CS decoding is necessary. Also notice that the A/D output data lines are connected to the microprocessor bus under program control through the PIA and therefore the A/D RD pin can be grounded.

Application Notes

NOTE #	DESCRIPTION	AnswerFAX DOC. #
AN016	"Selecting A/D Converters"	9016
AN018	"Do's and Don'ts of Applying A/D Converters"	9018
AN020	"A Cookbook Approach to High Speed Data Acquisition and Microprocessor Interfacing"	9020
AN030	"The ICL7104 - A Binary Output A/D Converter for Microprocessors"	9030



NOTE: Pin numbers for 8228 System Controller: Others are 8080A.

FIGURE 21. ADC0802 TO 8080A CPU INTERFACE

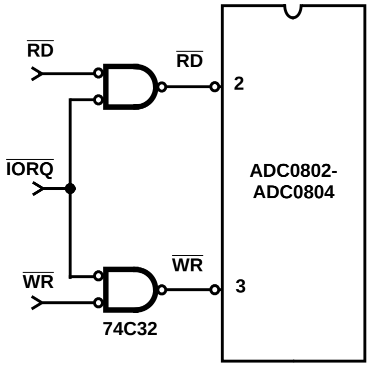
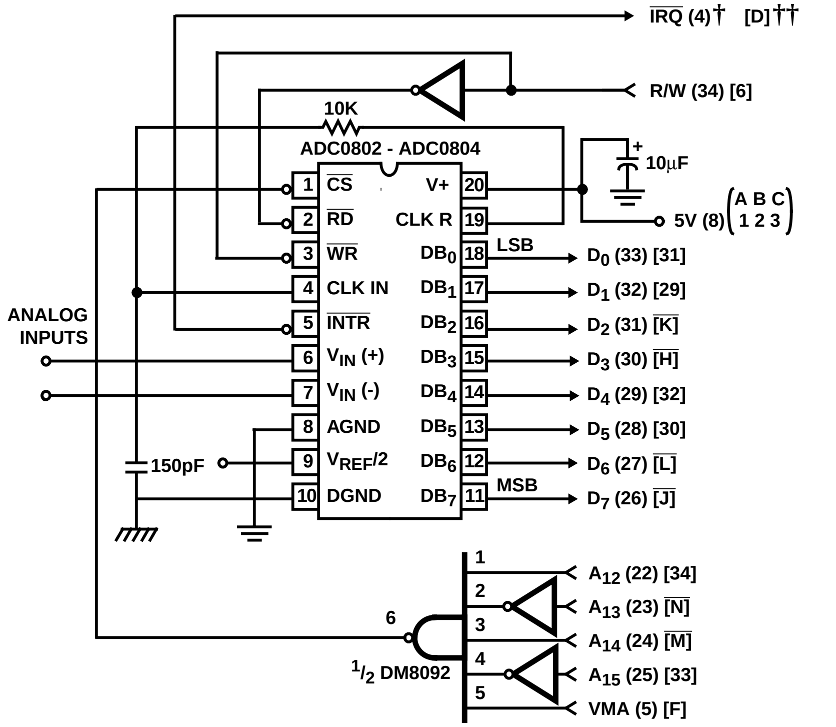


FIGURE 22. MAPPING THE A/D AS AN I/O DEVICE FOR USE WITH THE Z-80 CPU



† Numbers in parentheses refer to MC6800 CPU Pinout.

†† Numbers or letters in brackets refer to standard MC6800 System Common Bus Code.

FIGURE 23. ADC0802 TO MC6800 CPU INTERFACE

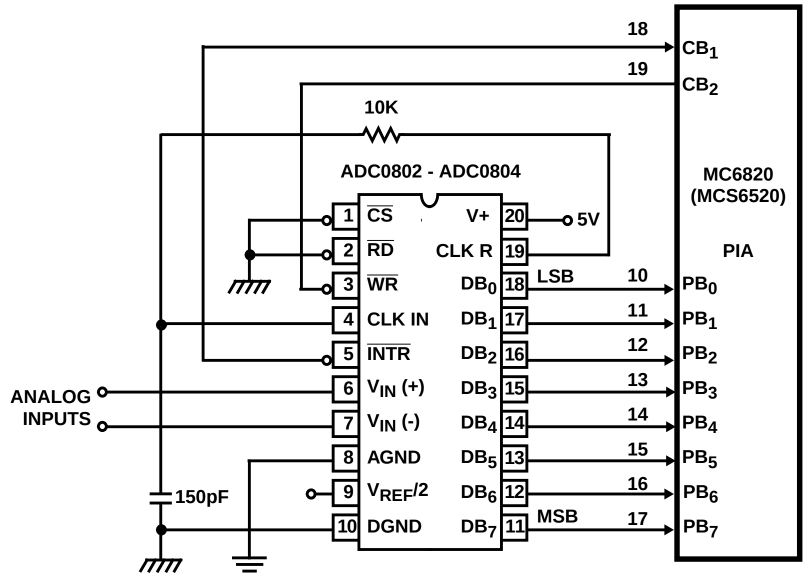


FIGURE 24. ADC0802 TO MC6820 PIA INTERFACE

Die Characteristics

DIE DIMENSIONS:

(101 mils x 93 mils) x 525µm x 25µm

METALLIZATION:

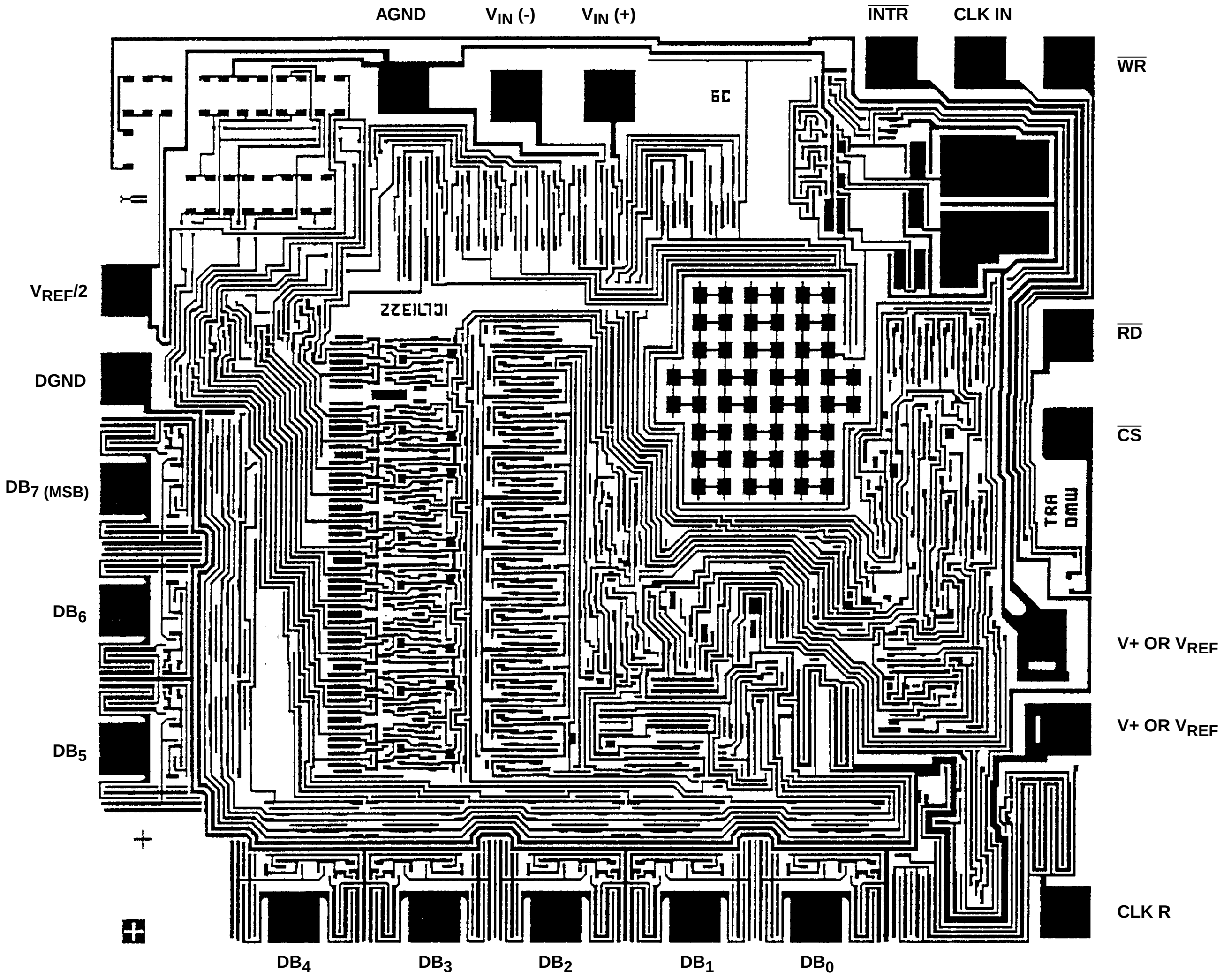
Type: Al
Thickness: 10kÅ ±1kÅ

PASSIVATION:

Type: Nitride over Silox
Nitride Thickness: 8kÅ
Silox Thickness: 7kÅ

Metallization Mask Layout

ADC0802, ADC0803, ADC0804


AGND
V_{IN} (-)
V_{IN} (+)
$\overline{INTR}$
CLK IN
$\overline{WR}$
$V_{REF}/2$
DGND
$\overline{RD}$
$\overline{CS}$
DB_7 (MSB)
DB_6
V+ OR V_{REF}
V+ OR V_{REF}
DB_5
CLK R
DB_4
DB_3
DB_2
DB_1
DB_0