



**AUTOMATIC MARK ANNOUNCING
SYSTEM**



P-1172

PROJECT REPORT

**Dissertation submitted in partial fulfilment of the
requirements for the award of the degree of**

**BACHELOR OF ENGINEERING IN INFORMATION
TECHNOLOGY
Of Bharathiar University**

Submitted by

JAYANTHI.R

KANAGARAJAN.P

KUPPURAJ.P

Guided by

Mrs.N.SUGANTHI M.E

Department of Information Technology

**KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE -641 006
TAMIL NADU
MARCH 2004**

CERTIFICATE

**DEPARTMENT OF
INFORMATION TECHNOLOGY
KUMARAGURU COLLEGE OF
TECHNOLOGY**
(Affiliated to Bharathiar University, Coimbatore)

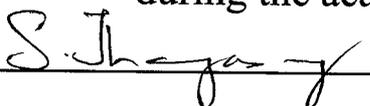


CERTIFICATE

This is to certify that the project entitled
**AUTOMATIC MARK ANNOUNCING
SYSTEM**

Done by
JAYANTHI.R (0027S0080)
KANAGARAJAN.P (0027S0083)
KUPPURAJ.P (0027S0085)

In partial fulfilment of the requirements for the award of the
degree of
Bachelor of Information Technology
Of Bharathiar University, Coimbatore,
during the academic year 2003-2004



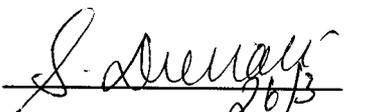
**Professor and Head of the
Department**



**Guide
Mrs.N.Suganthi M.E**

*Certified that the candidates were examined by us
in the project viva-voce examination*

held on 26.03.04



Internal Examiner

External Examiner

*AUTOMATIC MARK ANNOUNCING
SYSTEM*

DEDICATED TO OUR PARENTS
AND FRIENDS

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

Ever since the day we came up with the idea behind this project, we have been fortunate to have had the support and goodwill of several people who matter very much to us. Our gratitude to all of them is immense and we try to express as best as we can.

We are extremely grateful to Dr. K. K. Padmanabhan, B.Sc(Engg), M.Tech,Phd. Principal ,Kumaraguru college of Technology for having given us his valuable guidance and suggestions during the course of our education.

We are deeply obliged to Prof. Dr. S. Thangaswamy Phd., Head of the department of Computer Science for his guidance and useful suggestions during the course of this project.

We also extend our thanks to Mrs. S. Devaki, B.E., M.S., IT department Coordinator and Ms. P. Sudha, B.E., Lecturer, IT Department, our class Advisor for providing us their support and encouragement which helped us to come out with this project.

We are greatly indebted to Mrs.N.Suganthi, M.E., Lecturer,department of Information Technology, our Project Guide, for her helpful guidance and valuable support extended to us throughout the project.

Last but not the least we owe our gratitude and heartfelt thanks to our parents and friends for their encouraging support and Almighty for his abundant blessing for finishing this project on time.

SYNOPSIS

SYNOPSIS

The project entitles "**Automatic Mark Announcing System**", helps to retrieve the marks via telephone line.

The pulse generated when the phone connection is set up, is sent to the blocking capacitor to separate AC from DC. The opto coupler with the timer send the pulse according to the delay of the ring to the Decade Counter. At the fifth ring the clock inhibit in the decade counter will be high and the high pulse from Flip Flop is send to the relay driver to establish the call.

As the call gets established, the DTMF decoder gets enabled and the class code and roll number is entered. By using VB, the marks will be announced through the telephone line. The recorded messages from the voice processors will help the user to get the information details.

There are 5 modules used here, phone interface circuit, micro controller, voice recorder and playback circuit, power supply and system interface. The phone interface circuit consists DTMF decoder, ring detector, counter, decade counter, FlipFlop, opto coupler.

The micro controller is of AT89C51, 4 ports are used for class code checking, delay. The voice recorder and playback circuit used for announcing the results and welcome note. The powersupply is used to supply power to the circuit designed. The system interface by MAX 232 and RS 232. The coding used is embedded C and VB with MS Access.

CONTENTS

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 CURRENT STATUS OF THE PROBLEM UNDERTAKEN	
1.2 RELEVANCE AND IMPORTANCE	
2. SOFTWARE REQUIREMENTS SPECIFICATIONS	2
2.1 INTRODUCTION	
2.2 DEVELOPMENT AND OPERATING ENVIRONMENT	
2.3 MODULE DESCRIPTION	
2.4 REPORTS	
2.5 USER CHARACTERISTICS	
2.6 DEFINITIONS AND ACRONYMS	
3. BLOCK DESCRIPTION	5
3.1 BLOCK DIAGRAM	
3.2 COMPONENTS	
3.3 SOFTWARE SIDE	
4. CIRCUIT DESCRIPTION	8
4.1 PHONE INTERFACE CIRCUIT	
4.2 MICRO CONTROLLER	

4.3	VOICE RECORDER AND PLAY BACK CIRCUIT	
4.4	POWER SUPPLY	
4.5	SYSTEM INTERFACE	
5.	MICRO CONTROLLER CODING	23
6.	VB CODING	31
7.	CONCLUSION	38
8.	REFERENCE	39
9.	APPENDIX	40

INTRODUCTION

1) INTRODUCTION:

1.1 CURRENT STATUS OF THE PROBLEM TAKEN UP:

As of now the marks from the University or Exam center is been displayed in internet and via cellphone messages. But it serves for people who has cellphone and it take time to know the results in the peak time and also the costs cannot be offered by all. There may be some mismatch in marks can also take place.

1.2 RELEVANCE AND IMPORTANCE:

Our project helps to save time and expenditure involved in monitoring. It helps to hear the marks via LAN line, which is the easier way for people to retrieve marks from the database, without running out for marks. It is done serially, only one user can get the marks at a time. We also recorded the messages for results not ready stage, wrong class code, wrong roll no. which will help the user to get the correct marks of the required name and roll no. Since voice recorders cannot be directly connected to telephone line, we are using matching transformer. For serial communication, we are using RS232. along with this we are using ring detectors, ring counter, D flip flop, decoder.

*SOFTWARE REQUIREMENT
SPECIFICATION*

2) Software Requirement Specification

2.1 INTRODUCTION:

2.11 Purpose:

The purpose of our project titled “**Automatic mark announcing System**” is to monitor the retrieval of marks using DOT. The user interface developed is used to enter the details like 3 digit class code ,2 digit roll no and to announce the mark according to class code and roll no at that time. Reports are generated as and when required.

2.12 Scope:

Our project is applicable to all the users who need the marks to be read out but one at a time. Our project is easily modifiable for eg. it can be made as a parallel connection to engage many users.

2.2 DEVELOPMENT AND OPERATING ENVIRONMENT:

2.21 Hardware:

- * Processor : Pentium III
- * RAM : 128 MB
- * RS232 Serial port connector
- * PIC16F876 Microcontroller:
 - 28-Pin, 8-Bit FLASH Microcontroller
 - Operating Speed
 - 20MHz clock input
 - 200 ns instruction cycle
 - 8K FLASH Program Memory

- 368 Bytes of Data Memory
- 256 Bytes of EEPROM Data Memory
- Operating Voltage Range : 2.0V to 5.5V
- Current : 25 mA
- * MAX232 IC
- * LCD and LEDs

2.22 Software:

- * Win 3.1 and higher versions
- * Visual Basic 6.0
- * IDE : MPLAB
- * Compiler : Hi-tech
- * Embedded C.

2.3 MODULE DESCRIPTION:

2.31 Microcontroller Programming:

- * This module is used to access the signal generated by the sensor and increment the count in the microcontroller. The database is updated with the count every one minute from the start of the shift till its end.
- * The ids of the operator, component and process are retrieved and stored in the EEPROM at the start of the first shift.
- * During the start of each shift, the operator selects the required details for the shift which are sent to the PC.

2.32 Data Processing:

- * This module is used to maintain the database of the 2 digit class code to display.
- * At the start of the first shift the details are sent to the 89C51 Microcontroller.
- * The 3 digit class code is checked by the microcontroller are retrieved and updated in another database.
- * Whenever the count signal is received from the serial port, the corresponding database is updated with the count.

2.4 REPORTS:

Reports are generated using 3 digit class code determine the performance of the machine or operator as and when required.

Reports are generated in the form of voice using voice recorders with matching transformer.

2.5 USER CHARACTERISTICS:

The user has very little knowledge about computers, hence the interface (with four switches, LEDs and LCD display) is designed to be user friendly.

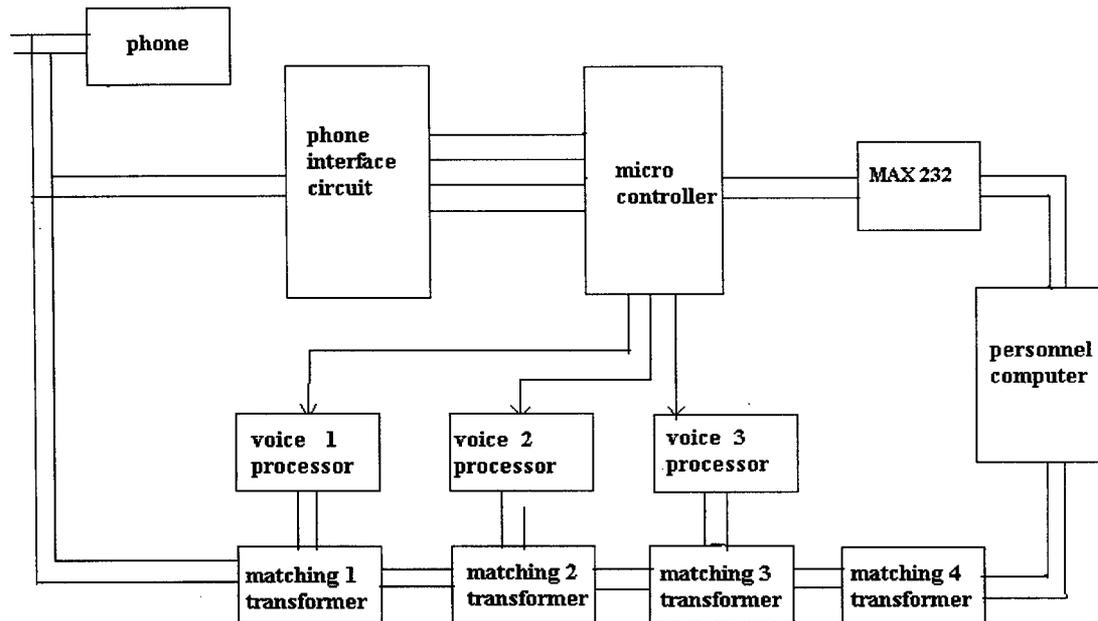
2.6 DEFINITIONS AND ACRONYMS:

- * AMAS → Automatic Mark Announcing System
- * RS232 → Serial port connector
- * LCD → Liquid Crystal Display
- * LED → Light Emitting Diode
- * MAX232 → IC used to provide compatibility between serial port and PIC Controller

BLOCK DESCRIPTION

BLOCK DESCRIPTION

3.1 BLOCK DIAGRAM



3.2 COMPONENTS

- The hardware part consists of
 - Ring detector circuit.
 - Ring counter circuit.
 - DTMF decoder circuit.
 - Micro controller.
 - Voice recorders.
 - Matching transformers.
 - Serial port driver.
 - Relay driver circuit.

RING DETECTOR

- The ring detector circuit is used to detect the ring pulse when somebody calls.
- By using an opto coupler it senses the ring signal and it gives 1 pulse for each ring to the ring counter circuit.

RING COUNTER

- The ring counter is used to set pre programmed no of rings.
- And after that no rings it automatically makes the phone to off hook condition and it enables the DTMF decoder.
- Thus it will ready to accept the DTMF tones.

DTMF DECODER

- The DTMF decoder is a chip which will give BCD o/p for the corresponding DTMF i/p.
- The BCD o/p is given to the Micro controller for further operation.

MICRO CONTROLLER

- The Micro controller is the heart of the system.
- First it checks for the valid 3 digit class code.
- Then it accepts the 2 digit roll no.
- And it sends those roll no.to the computer via serial port driver circuit.
- For identification of class it will send one alphabet for valid class code.
- It also activates the Voice recorders to give the preloaded voice msgs.

VOICE RECORDERS

- Voice recorders are used to play welcome note, how to operate the system, to know the correct class code no. Etc.
- We can store upto 30 seconds of voice in these Voice recorders.

MATCHING TRANSFORMER

- To connect the audio o/p from this Voice recorders and also from computer to the telephone line a matching transformer is used.
- It will couple the audio to the telephone line.

3.3 SOFTWARE SIDE

- The marks are stored in the database system individually.
 - A VB program is made to run to receive the BCD codes from the serial port of the computer and it opens the file in data base system automatically.
 - A text to speech component is used to convert those marks in the text format to voice.
 - The audio o/p from the pc is connected to the telephone line via matching transformer.
 - Thus the marks are announced via telephone line automatically.
 - Stk or any other text to speech component can used to convert text to speech.
- We can store any number of data's in the PC.

CIRCUIT DESCRIPTION

4.CIRCUIT DESCRIPTION

The circuit description is divided into 5 modules

- 1) Phone interface circuit
- 2) Microcontroller
- 3) Voice recorder and playback circuit
- 4) Power supply
- 5) System interface

4.1) PHONE INTERFACE CIRCUIT:

The phone interface circuit consists of

- a) coupling capacitor
- b) oscillator
- c) ring counter
- d) Decade counter
- e) flip flop
- f) dtmf decoder

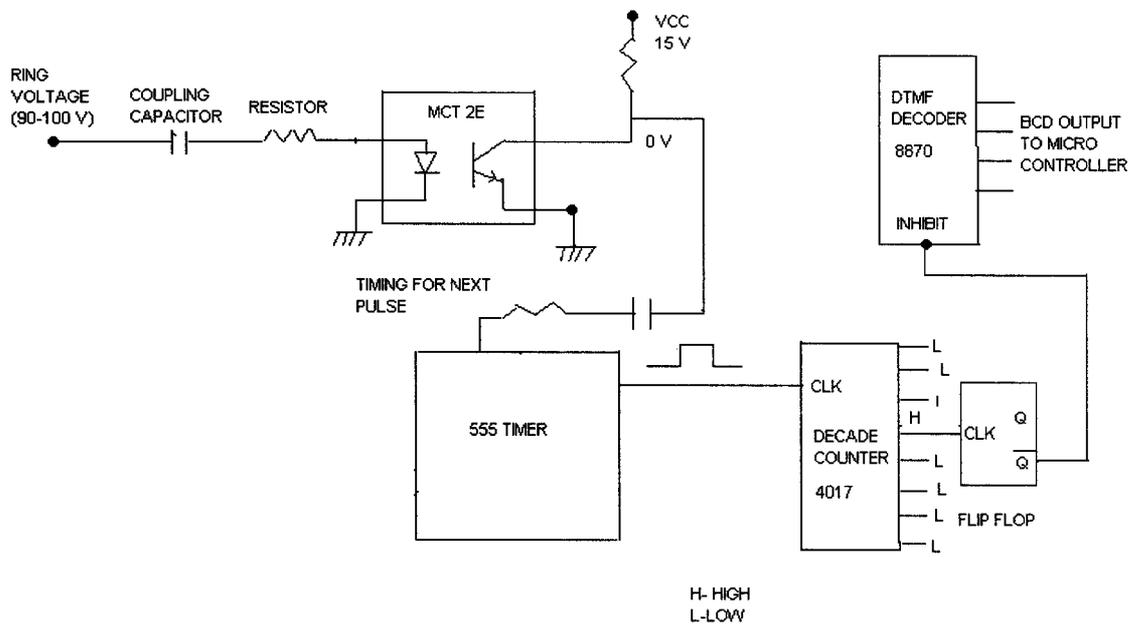
The chips used in the interface circuit are C MOS , which only take 5 V. As the ring voltage from the phone line is 90-100 V, blocking capacitor along with the resistor is used. The generated pulse is given to the OPTO coupler.

The opto coupler senses the ring signal and gives 1 pulse for each ring to the ring counter circuit. The pulse generated is given to oscillator.

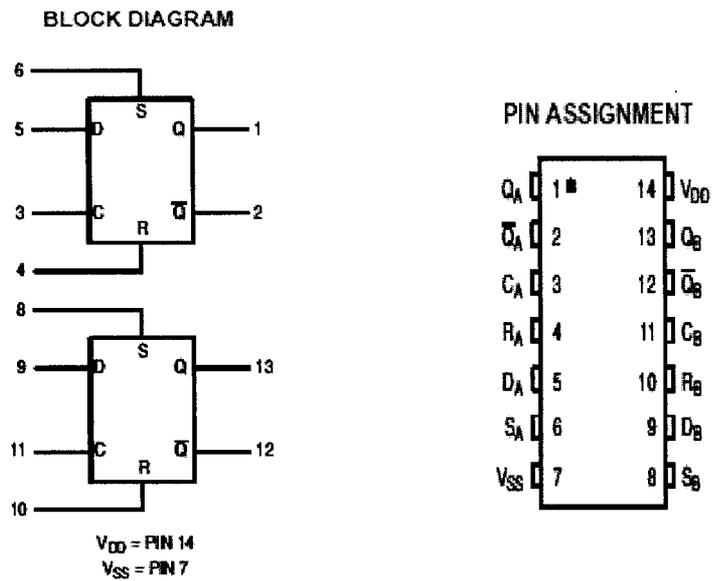
The ring counter is used to set pre programmed no of rings. Then the output is given to the 4 stage decade counter, 1 high 1 low and 2 high. The relay gets on. The fifth ring is given to the flip flop,at first it is set to 0 then when the ring gets

detected it becomes 1. the relay is of 220 ohm. The high input is given to the DTMF decoder. At once the dtmf decoder get enabled. The dtmf decoder is a chip which will give BCD o/p for the corresponding dtmf i/p. then the BCD o/p is given to the micro controller.

CIRCUIT DIAGRAM OF PHONE INTERFACE CIRCUIT



D- FLIP FLOP



4.2) MICRO CONTROLLER:

The micro controller used here is IC 89C51. The features are as follows

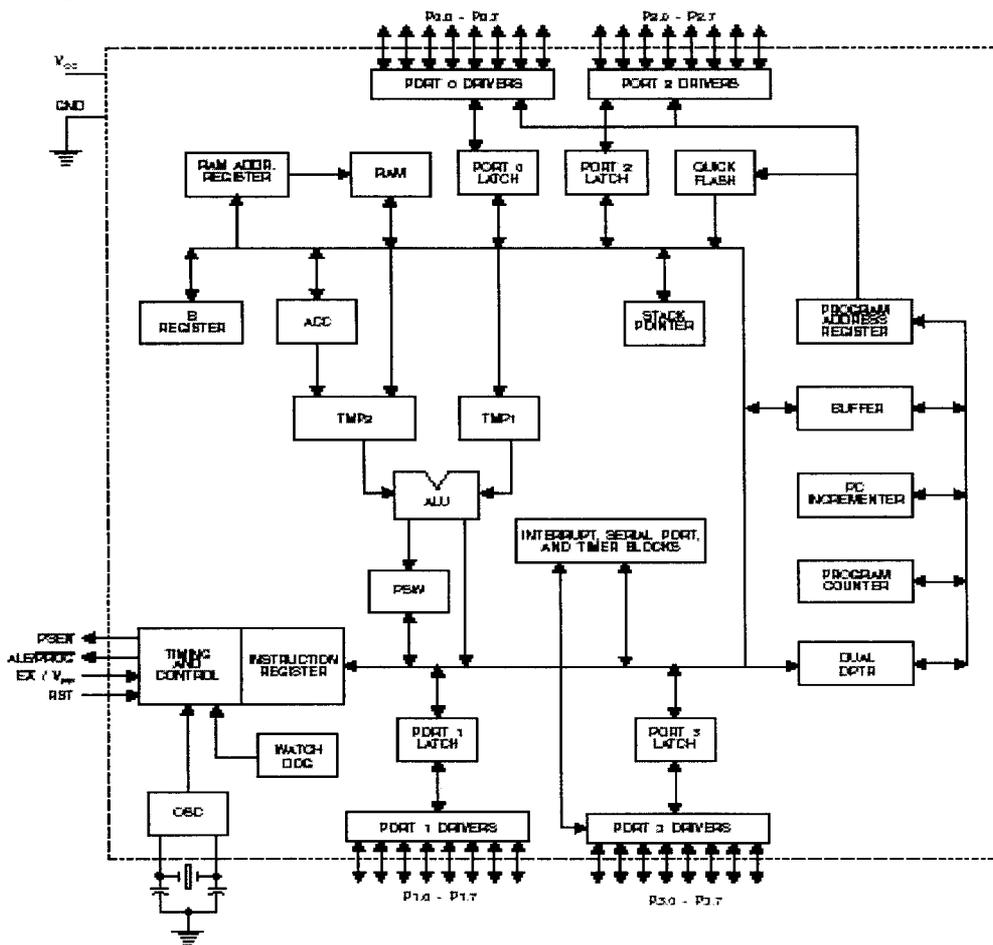
- ✓ 8051 based architecture
- ✓ 4-K bytes of on-chip Reprogrammable Flash memory
- ✓ 128 x 8 RAM
- ✓ two 16-bit timer/counter
- ✓ full duplex serial channel
- ✓ Boolean processor
- ✓ Four 8 bit I/O ports, 32 I/O lines
- ✓ Memory addressing capability
- ✓ Program memory lock
- ✓ Six interrupt sources
- ✓ Most instructions execute in 0.3 sec

PIN CONFIGURATION

PDIP

(T2) P1.0	1	40	VCC
(T2EX) P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
P1.5	6	35	P0.4 (AD4)
P1.6	7	34	P0.5 (AD5)
P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	EA/VPP
(TXD) P3.1	11	30	ALE/PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
(WR) P3.6	16	25	P2.4 (A12)
(RD) P3.7	17	24	P2.3 (A11)
XTAL2	18	23	P2.2 (A10)
XTAL1	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)

Block Diagram



AT89C51RC

GENERAL DESCRIPTION:

The IS89C51 is a high performance microcontroller fabricated using high density CMOS technology. The CMOS IS89C31 is functionally compatible with the industry standard 80C51 microcontrollers.

The IS89C51 is designed with 4-K bytes of flash memory, 128 x 8 RAM; 32 programmable I/O lines; a serial I/O port for either multiprocessor communications, I/O expansion or full duplex USART; two 16 bit timer/counters;

An six-source, two priority-level, nested interrupt structure; and an onchip oscillator and clock circuit. The IS89C31 can be expanded using standard TTL compatible memory.

OPERATING DESCRIPTION:

The detail description of IS89C51 included in this description are

- memory map
- timer/counters
- flash memory

MEMORY MAP:

The IS89C51 has separate address spaces for program and data memory. The program and data memory can be up to 64 K bytes long. The lower 4 K program memory can reside on-chip.

The IS89C51 has 128 bytes of on-chip RAM, plus numbers of special function registers. The lower 128 bytes can be accessed either by direct addressing or by indirect addressing. The lower 128 bytes of RAM can be divided into three segments.

1. **register banks 0-3:** locations 00H through 1FH. The device after reset defaults to register bank 0. To use the other register banks, the user must select them in the software. Each register bank contains eight 1-byte registers R0-R7. Reset initializes the stack point to location 07H, and is incremented once to start from 08H, which is the first register of the second register bank.
2. **bit addressable area:** 16 bytes have been assigned for this segment 20H-2FH. Each one of the 128 bits of this segment can be directly addressed. Each of the 16 bytes in this segment can also be addressed as a byte.
3. **scratch pad area:** 30H-7FH are available to the user as data RAM. However, if the data pointer has been initialised to this area, enough bytes should be left aside to prevent SP data destruction.

TIMER/COUNTERS:

The IS89C51 has two 16-bit Timer/Counter registers: Timer 0 and Timer 1. All two can be configured to operate either as Timers or event counters. As a Timer, the register is incremented every machine cycle. Thus, the register counts the machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of

the oscillator frequency. As a counter, the register is incremented in response to a 1-0 transition at its corresponding external input pin, T0 to T1. The external is sampled during S5P2 of every machine cycle.

In addition to the Timer or Counter functions, Timer 0 and Timer 1 have four operating modes: 13-bit timer, 16-bit timer, 8-bit auto reload, split Timer.

TIMER 0 AND TIMER 1:

The timer or counter function is selected by control bits C/T in the special function register TMOD. These two timers/counters have four operating modes, which are selected by bit parts (M1, M0) in TMOD. Modes 0, 1 and 2 are the same for both timer/counters, but mode 3 is different.

The four modes are described in the following sections.

MODE 0:

Both timers in mode 0 are 8 bit counters with a divide by 32 prescaler. In this mode, the timer register is configured as a 13 bit register. As the count rolls over from all of 1s to all 0s, it sets the timer interrupt flag TF1. The counted input is enabled to the timer when TR1=1 and either GATE=0 or INT1=1. Setting GATE=1 allows the timer to be controlled by external input INT1, to facilitate pulse width measurements. TR1 is a control bit in the special function register TCON. Gate is in TMOD. The 13 bit register consists of all eight bits of TH1 and the lower five bits of TL1. The upper three bits of TL1 are indeterminate and should be ignored. Setting the run flag does not clear the registers. There are two different GATE bits, one for Timer 1 and one for Timer 0.

MODE 1:

Mode 1 is same as Mode 0, except that the time register is run with all 16 bits. The clock is applied to the combined high and low timer registers. As clock pulses are received, the timer counts up:0000H,0001H,0002H,etc. an overflow flag. The timer continues to count. The overflow flag is the TF1 bit in TCON that is read or written by software.

MODE 2:

Mode 2 configures the Timer register as an 8-bit Counter with automatic reload. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves the TH1 unchanged. Mode 2 operation is the same for Timer/Counter 0.

MODE 3:

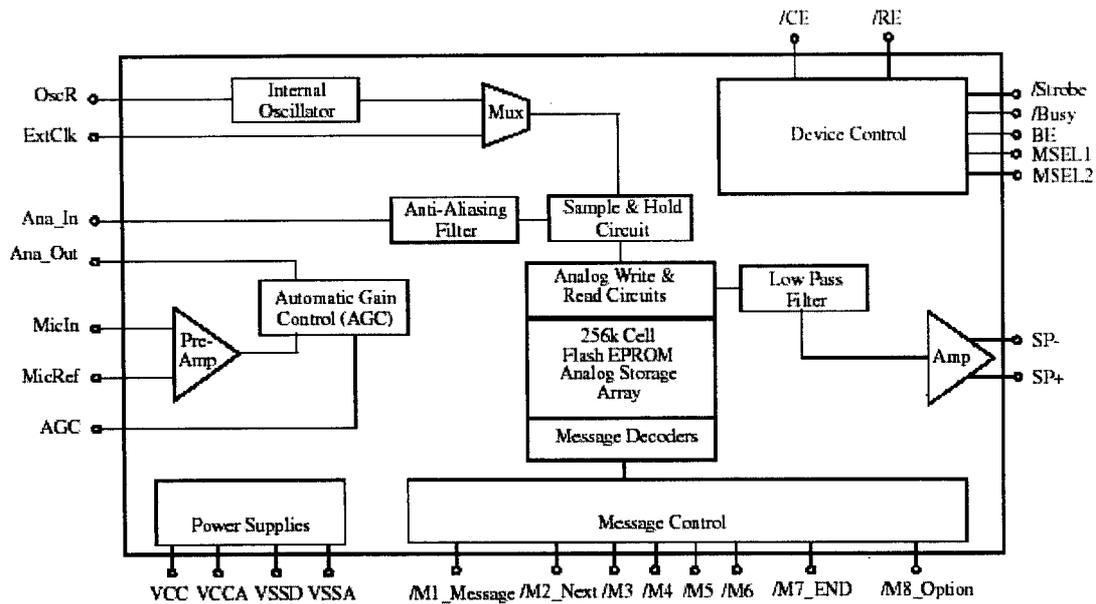
Timer 1 in mode 3 simply holds its count. The effect is the same as setting TR1=0. timer 0 in mode 3 establishes TL0 and TH0 as two separate counters. TL0 uses the timer 0 control bits: C/T,GATE,TR0,INT0 and TF0.

4.3) VOICE RECORDER AND PLAYBACK CIRCUIT:

Features

- Single-chip, high-quality voice recording & playback solution
- No external ICs required
- Minimum external components
- Non-volatile Flash memory technology
- No battery backup required
- User-Selectable messaging options
- Random access of multiple fixed-duration messages
- Sequential access of multiple variable-duration messages
- User-friendly, easy-to-use operation
- Programming & development systems not required
- Level-activated recording & edge-activated playback switches
- Low power consumption
- Operating current: 25 mA typical
- Standby current: 1mA typical
- Automatic power-down
- Chip Enable pin for simple message expansion

BLOCK DIAGRAM:



Functional Description

The APR9600 block diagram is included in order to give understanding of the APR9600 internal architecture. At the left hand side of the diagram are the analog inputs. A differential microphone amplifier, including integrated AGC, is included on-chip for applications requiring its use. The amplified microphone signal is fed into the device by connecting the Ana_Out pin to the Ana_In pin through an external DC blocking capacitor. Recording can be fed directly into the Ana_In pin through a DC blocking capacitor, however, the connection between Ana_In and Ana_Out is still required for playback. The next block encountered by the input signal is the internal anti-aliasing filter. The filter automatically adjusts its response according to the sampling frequency selected so Shannon's Sampling Theorem is satisfied. After anti-aliasing filtering is accomplished the signal is ready to be clocked into the memory array. This storage is accomplished through a combination of the Sample and Hold circuit and the Analog Write/Read circuit. These circuits are clocked by either the

Internal Oscillator or an external clock source. When playback is desired the previously stored recording is retrieved from memory, low pass filtered, and amplified as shown on the right hand side of the diagram. The signal can be heard by connecting a speaker to the SP+ and SP- pins. Chip-wide management is accomplished through the device control block shown in the upper right hand corner. Message management is controlled through the message control block represented in the lower center of the block diagram. More detail on actual device application can be found in the Sample Applications section. More detail on sampling control can be found in the Sample Rate and Voice Quality section. More detail on message management and device control can be found in the Message Management section.

Message Management General Description:

Playback and record operations are managed by on chip circuitry. There are several available messaging modes depending upon desired operation. These message modes determine message management style, message length, and external parts count. Therefore, the designer must select the appropriate operating mode before beginning the design. Operating modes do not affect voice quality; for information on factors affecting quality refer to the Sampling Rate & Voice Quality section. The device supports three message management modes

- Random access mode with 2, 4, or 8 fixed-duration messages
- Tape mode, with multiple variable-duration messages, provides two options:
 - Auto rewind
 - Normal

Modes cannot be mixed. Switching of modes after the device has recorded an initial message is not recommended. If modes are switched after an initial recording has

been made some unpredictable message fragments from the previous mode may remain present, and be audible on playback, in the new mode. These fragments will disappear after a record operation in the newly selected mode. Table 1 defines the decoding necessary to choose the desired mode. An important feature of the APR9600 message management capabilities is the ability to audibly prompt the user to changes in the device's status through the use of "beeps" superimposed on the device's output. This feature is enabled by asserting a logic high level on the BE pin.

4.4) POWER SUPPLY:

OVERALL WORKING OF POWER SUPPLY:

The 230 V line input is stepped down to 12 V by the step down transformer. It works on the principle of mutual inductance between two inductively coupled coils . the two windings are coupled by magnetic induction. One of the windings called the primary is energized by sinusoidal voltage. The secondary voltage feeds the load. This is then given to the bridge rectifier. It rectifies ac input into pulsating dc using the diodes D1,D2,D3,D4.the diodes D1 & D3 rectify the positive half cycle and diodes D2&d4 rectify the negative half cycle. This is then filtered using the capacitor filter which takes in the ac component of the wave. The capacitor let only ac through it and block the dc input. The capacitor is placed in parallel in the circuit here. The output is a smooth dc output. This is then given to the voltage regulator which maintains the voltage at a particular level. It uses the principles of line/ input regulation,load regulation, ripple rejection. Here there are three levels 5V,6V,12V.according to the voltage level, the corresponding LED's ill glow. If any led does nt glow, it means there is a fault in the voltage regulator related to it. If all LED's doesn't glow it means there is a fault in the beginning of the circuit ie the transformer, rectifier or capacitor filters.

ELECTRICITY BOARD:

Electricity board provides the main supply of electricity of the circuit. It is connected to a transformer which steps down the 230 V line voltage to the 12 V required for the circuit. The power goes to the power supply board which is where the ic chips other components are embedded. It has a capacitor , resistor, bridge rectifier, voltage regulator, potentiometer,heat sink and LED.

4.5) SYSTEM INTERFACE:

The approach through interface with the micro controller and input/output device is determined by peripheral input output transfer. The device is identified by an eight addresses and data is transferred.

Following interface is used in the project:

- RS232 serial port interface for PC communication.

RS 232 SERIAL PORT INTERFACE:

In the early days of computing, although data could be transmitted at high speed, it could not be read and processed continuously. So , a set of handshaking lines and protocols were developed for what became known as RS-232 Serial Communications.

The typical packet contains seven bits,(which is the number of bits that each ASCII character contained). This simplifies the transmission of man-readable text,but makes sending object code and data(which were arranged as bytes) more complex because each byte would have to be split up into two nibbles. Further complication is that the first 32 characters of the ASCII character set are defined as" special"

characters. This means the data nibbles would have to be converted into valid characters. With this protocol, to send a single byte of data, two bytes would have to be sent. As pointed out, modern asynchronous serial data transmission normally occurs eight bits at a time, which will avoid this problem and allow transmission of full bytes without breaking them up or converting them. In RS 232 different equipment is wired according to the function they perform.

TYPICAL RS-232 WIRING:

DTE is meant to be the connector used for computers. DCE was meant for modems and terminals that transfer the data.

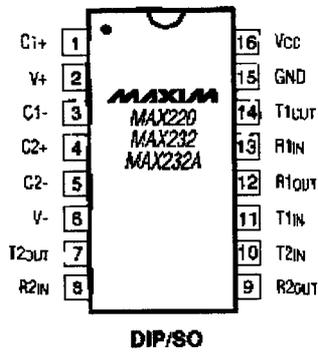
The Request To Send and Clear To Send lines are used to control data flow between the computer and the micro controller. When the PC is ready to send data, it activates the RTS line. If the DTE device is capable of receiving data, it activates the CTS line. If the PC is unable to receive data, it will de-activate the RTS line to notify the DTE device that it cannot receive any additional information.

The Data Transmission Ready and Data Set Ready lines are used to establish communications. When the PC is ready to accept data, it will assert DSR to notify the computer that the link is up and ready for data transmission.

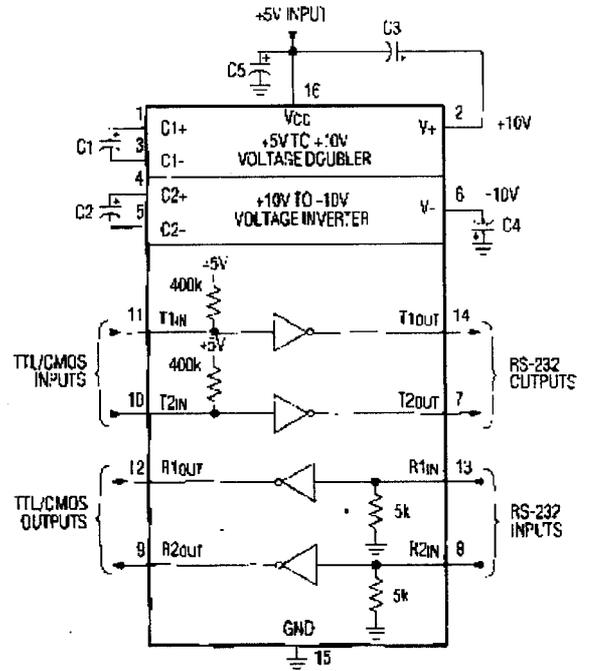
MAX 232:

The MAX232 is used to regulate the voltage for the serial communication via RS 232. It uses TTL chips which is of +5 V and -5 V. But PC accepts +12 V and -12 V. The input through MAX 232 is TTL input AND the output is RS 232 output. The capacitors used in MAX 232 are used to give +12 V and -12 V to PC. A LED in the MAX 232 glows whenever the valid data is passed.

MAXIM MAX 232 RS-232 CONNECTION



CAPACITANCE (pF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1



***MICRO CONTROLLER
CODING***

5. CODING

```
# include <stdio.h>
# include <reg51.h>
#define TV0 0xfe //TIMER 0 INTERRUPT CALL
#define TVL0 0x0b
#define PORT P0//(P00-P03)
#define dialport P1

    unsigned char temp,dl[12],totdig;
    void delay();
    void clock();
    void dialloop();

bit flagt;
bit flagr;
bit count_flag;
bit flag;
bit flag1;
bit half_flag;
sbit STD=P04;
sbit sw_ip=P10;

void dial();
void password();
void password1();
void delay5();
void delay1();
void delay2();
void voice_trans();
unsigned char p,rec_reg,cell,sec,min,g,pulse,a,k,num[10],psw[4],i;
unsigned int h,misec,f;
```

```

void main()
{
PCON=0x0;
IE=0x1a;
IP=0x10;
TMOD=0x21;
SCON=0x50;
TL1=0xe8;
TH1=0xe8;
TCON=0x40;
TI=0;
RI=0;
flagt=0;
flagr=0;
rec_reg=0;
EA=1;
count_flag=0;
misec=0;
flag=1;
flag1=0;
P20=0;
EA=1;
while(1)
{
    if(sw_ip==1)
    {
        password();
        if(half_flag==0)
        {
            P21=0;
            password1();
        }
        P21=1;
    }
}

```

```

else
{
    P22=0;
    delay();
    P22=1;
    delay();
    delay();
    delay();
    delay();
    delay();
    delay();
    delay();
    delay();
    delay();
    delay();
}
}

```



```

}
void timer0() interrupt 1
{
    EA=0;

    TR0=0;
    TH0=TV0;//0xfe;
    TL0=TVL0;//0x0b;
    msec=msec+1;

    if(msec ==950)
    {
        msec=0;
        sec=sec+1;

        if (sec ==60) // check for 60 seconds
        {
            sec=0;
            min=min +1;
        }
    }
}

```

```

        if(min ==1)
        {
            min=0;
            count_flag=1;
        }
    }
    EA=1;
    TR0=1;
}

serint ()interrupt 4
{
    EA=0;
    switch (TI)
    {
        case 1:
            {TI=0;g
            flagt=1;
            break;
            }
        case 0:
            {
            switch (RI)
            {
                case 0:
                    break;
                case 1:
                    {rec_reg=SBUF;
                    RI=0;
                    flagr=1;
                    break;}
            }
            }
    }
    EA=1;
}

```

```

void password()
{
    for(i=1;i<=3;)
    {
        do{;}while(STD==0);
        do{;}while(STD==1);
        psw[i]=PORT & 0x0f;
        delay2();
        i++;
    }

    if(psw[1]==0x01 && psw[2] ==0x02 && psw[3]==0x03)
    {
        SBUF=0x0a;
        do{;}while(flagt==0);
        flagt=0;
        half_flag=0;
        P24=0;
        delay();
        P24=1;
    }
    else if(psw[1]==0x01 && psw[2] ==0x03 && psw[3]==0x03)
    {
        SBUF=0x0b;
        do{;}while(flagt==0);
        flagt=0;
        half_flag=0;
        P24=0;
        delay();
        P24=1;
    }
    else if(psw[1]==0x01 && psw[2] ==0x04 && psw[3]==0x03)
    {
        SBUF=0x0c;
    }
}

```

```

do{;}while(flagt==0);
flagt=0;
half_flag=0;
P24=0;
delay();
P24=1;

}

else if(psw[1]==0x01 && psw[2] ==0x05 && psw[3]==0x03)
{
SBUF=0x0d;
do{;}while(flagt==0);
flagt=0;
half_flag=0;
P24=0;
delay();
P24=1;
}
else
{
P27=0;
delay();
P27=1;
half_flag=1;

}
}

```

```

void password1()
{
for(i=1;i<=2;)
{
do{;}while(STD==0);

```

```
do{;}while(STD==1);  
psw[i]=PORT & 0x0f;
```

```
delay2();  
i++;
```

```
}
```

```
cell=psw[1];  
h=psw[2]<<4;  
cell=cell|h;
```

```
SBUF=cell;  
do{;}while(flagt==0);  
flagt=0;
```

```
do{;}while(STD==0);  
do{;}while(STD==1);  
psw[1]=PORT & 0x0f;  
if(psw[1]==0x0b)
```

```
{
```

```
    P26=0;  
    delay();  
    P26=1;
```

```
}
```

```
else if(psw[1]==0x0c)
```

```
{
```

```
    P25=0;  
    delay();  
    P25=1;
```

```
}
```

```
}
```

```
void delay5()
{
for(f=1;f<=5000;)//5000
{f++;}
}
```

```
void delay1()
{
for(f=1;f<=5000;)//20000
{f++;}

}
```

```
void delay2()
{for (f=1;f<=1000;)//15000
  {f++;}
}
```

```
void delay()
{
    unsigned int z;
    for (z=0;z<=60000;z++)
    {}
}
```

VB CODING

6.CODING

```
Dim v_dept As String
Dim rs_msg As Recordset
Dim rs_mark As Recordset
Dim db As Database
Private Sub Command1_Click()
    Set rs_mark = db.OpenRecordset("select * from marklist where ucase(trim(dept))=""
    & UCase(Trim(txt_dept.Text)) & "" and ucase(trim(rollno1))="" &
    UCase(Trim(txt_rollno1.Text)) & """, dbOpenDynaset)
    If rs_mark.RecordCount > 0 Then
        rs_mark.Edit
    Else
        rs_mark.AddNew
    End If
    rs_mark!dept = txt_dept.Text
    rs_mark!Name = txt_name.Text
    rs_mark!rollno1 = txt_rollno1.Text
    rs_mark!rollno2 = txt_rollno2.Text
    rs_mark!sub_details = r_text.Text
    rs_mark.Update
    Set rs_msg = db.OpenRecordset("select * from msg", dbOpenDynaset)
    If rs_msg.RecordCount > 0 Then
        rs_msg.Edit
    Else
        rs_msg.AddNew
    End If
    rs_msg!end_msg = txt_rem.Text
    rs_msg.Update
    clear
    txt_dept.Text = ""
    txt_rollno1.Text = ""
End Sub
```

```
Private Sub Command2_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
    Set rs_mark = db.OpenRecordset("select * from marklist where ucase(trim(dept))=""  
& UCase(Trim(txt_dept.Text)) & "" ucase(trim(rollno1))="" &  
UCase(Trim(txt_rollno1.Text)) & """, dbOpenDynaset)
```

```
    If rs_mark.RecordCount > 0 Then
```

```
        rs_mark.Delete
```

```
    End If
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Set db = DBEngine.Workspaces(0).OpenDatabase("c:\jai\Psg\marklist.mdb")
```

```
    If MSComm1.PortOpen = True Then
```

```
        MSComm1.PortOpen = False
```

```
    End If
```

```
    MSComm1.InputMode = comInputModeText
```

```
    MSComm1.InputLen = 0
```

```
    MSComm1.InBufferSize = 1024
```

```
    MSComm1.Handshaking = comNone
```

```
    MSComm1.RThreshold = 1
```

```
    MSComm1.SThreshold = 0
```

```
    MSComm1.ParityReplace = ""
```

```
    MSComm1.Settings = "1200,n,8,1"
```

```
    MSComm1.CommPort = 1
```

```
    MSComm1.PortOpen = True
```

```
End Sub
```

```

Public Sub clear()
    txt_name.Text = ""
    txt_rollno2.Text = ""
    r_text.Text = ""
    txt_rem.Text = ""
    Set rs_msg = db.OpenRecordset("select * from msg", dbOpenDynaset)
    If rs_msg.RecordCount > 0 Then
        If IsNull(rs_msg!end_msg) = False Then
            txt_rem.Text = rs_msg!end_msg
        End If
    End If
End Sub

```

```

Private Sub MSComm1_OnComm()
    Select Case MSComm1.CommEvent

```

```

        Case 2 'comEvReceive

```

```

            Dim v_In As String

```

```

            Dim v_Roll As String

```

```

            v_In = ""

```

```

            v_In = MSComm1.Input

```

```

            If v_In = Chr(10) Then

```

```

                v_dept = "CT"

```

```

            ElseIf v_In = Chr(11) Then

```

```

                v_dept = "EC"

```

```

            ElseIf v_In = Chr(12) Then

```

```

                v_dept = "IT"

```

```

            ElseIf v_In = Chr(13) Then

```

```

                v_dept = "EEE"

```

```

            ElseIf v_In <> Chr(10) And v_In <> Chr(11) And v_In <> Chr(12) And v_In <>

```

```

Chr(13) Then

```

```

                v_Roll = Format(Search_no(v_In), "00")

```

```

                txt_dept.Text = v_dept

```

```

                txt_rollno1.Text = v_Roll

```

```

    "" Check & Display The Record In That Particular Department & RollNo.
    Set rs_mark = db.OpenRecordset("select * from marklist where
ucase(trim(dept))="" & UCase(Trim(txt_dept.Text)) & "" and ucase(trim(rollno1))="" &
UCase(Trim(txt_rollno1.Text)) & """, dbOpenDynaset)
    If rs_mark.RecordCount > 0 Then
        retrieve
        DirectSS2.Speed = 120
        DirectSS2.Pitch = 222
        DirectSS2.Speak "Department. " & txt_dept.Text
        DirectSS2.Speak "Name . " & txt_name.Text
        DirectSS2.Speak "Roll Number . " & txt_rollno2.Text
        DirectSS2.Speak r_text.Text
        DirectSS2.Speak txt_rem.Text
    Else
        DirectSS2.Speak "Sorry Your roll Number Does Not Exist . "
        clear
    End If
End If
End Select
End Sub

```

```

Private Sub txt_dept_DropDown()
    txt_dept.clear
    txt_dept.AddItem "CT"
    txt_dept.AddItem "EC"
    txt_dept.AddItem "IT"
    txt_dept.AddItem "EEE"
End Sub

```

```

Private Sub txt_dept_KeyPress(KeyAscii As Integer)
    KeyAscii = 0
End Sub

```

```

Private Sub txt_rollno1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        Set rs_mark = db.OpenRecordset("select * from marklist where
ucase(trim(dept))="" & UCase(Trim(txt_dept.Text)) & "" and ucase(trim(rollno1))="" &
UCase(Trim(txt_rollno1.Text)) & """, dbOpenDynaset)
        clear
        If rs_mark.RecordCount > 0 Then
            retrieve
        End If
    End If
End Sub

```

```

Public Sub retrieve()
    clear
    txt_dept.Text = rs_mark!dept
    If IsNull(rs_mark!Name) = False Then
        txt_name.Text = rs_mark!Name
    End If
    If IsNull(rs_mark!rollno1) = False Then
        txt_rollno1.Text = rs_mark!rollno1
    End If
    If IsNull(rs_mark!rollno2) = False Then
        txt_rollno2.Text = rs_mark!rollno2
    End If
    If IsNull(rs_mark!sub_details) = False Then
        r_text.Text = rs_mark!sub_details
    End If
    Set rs_msg = db.OpenRecordset("select * from msg", dbOpenDynaset)
    If rs_msg.RecordCount > 0 Then
        If IsNull(rs_msg!end_msg) = False Then
            txt_rem.Text = rs_msg!end_msg
        End If
    End If
End Sub

```

```

Public Function Search_no(v_chr As Variant)
    Dim v_no As Integer
    a1 = "26,42,58,74,90,106,122,138,154,161"
    a2 = "17,33,49,65,81,97,113,129,145,162"
    a3 = "18,34,50,66,82,98,114,130,146,163"
    a4 = "19,35,51,67,83,99,115,131,147,164"
    a5 = "20,36,52,68,84,100,116,132,148,165"
    a6 = "21,37,53,69,85,101,117,133,149,166"
    a7 = "22,38,54,70,86,102,118,134,150,167"
    a8 = "23,39,55,71,87,103,119,135,151,168"
    a9 = "24,40,56,72,88,104,120,136,152,169"
    a10 = "25,41,57,73,89,105,121,137,153,170"
    v_no = Asc(v_chr)
    If InStr(a1, v_no) > 0 Then
        Search_no = Find_roll(Trim(a1), v_no, 0)
    ElseIf InStr(a2, v_no) > 0 Then
        Search_no = Find_roll(Trim(a2), v_no, 10)
    ElseIf InStr(a3, v_no) > 0 Then
        Search_no = Find_roll(Trim(a3), v_no, 20)
    ElseIf InStr(a4, v_no) > 0 Then
        Search_no = Find_roll(Trim(a4), v_no, 30)
    ElseIf InStr(a5, v_no) > 0 Then
        Search_no = Find_roll(Trim(a5), v_no, 40)
    ElseIf InStr(a6, v_no) > 0 Then
        Search_no = Find_roll(Trim(a6), v_no, 50)
    ElseIf InStr(a7, v_no) > 0 Then
        Search_no = Find_roll(Trim(a7), v_no, 60)
    ElseIf InStr(a8, v_no) > 0 Then
        Search_no = Find_roll(Trim(a8), v_no, 70)
    ElseIf InStr(a9, v_no) > 0 Then
        Search_no = Find_roll(Trim(a9), v_no, 80)

```

```
Elseif InStr(a10, v_no) > 0 Then
    Search_no = Find_roll(Trim(a10), v_no, 90)
End If
End Function
```

```
Public Function Find_roll(v_str As String, Roll_no As Integer, pos As Integer)
    a = Split(v_str, ",")
    For i = 0 To UBound(a)
        If a(i) = Roll_no Then
            Find_roll = i + 1 + pos
            Exit Function
        End If
    Next i
End Function
```

CONCLUSION

7. CONCLUSION AND FUTURE OUTLOOK

This project undertaken is a simulation of retrieving marks from the LAN line without any more expenses such as internet, cellphone. It is fairly simple and more efficient than that one using in the present system. Since this is done in serial , one can use the line at a time. The line gets hooked after the fifth ring then the line will be engaged. This can be used by every people,whoever has the LAN line(most of them). It can be further developed parallel for the maximum number of users who can retrieve the marks at the same time. Since we have done the system using database , which can be expanded ,that it can be connected to internet to retrieve the marks as soon as the university results been displayed. The recorded messages will help the user to enter the correct class code and roll no.

REFERENCES

8. REFERENCES:

- Millman & Halkias," Electronic devices and circuits ", Tata McGraw-Hill Edition , 2000.
- D.Roy Choudhury & Shail Jain , " Linear Integrated Circuits", New Age International(p) Limited,Publishers , 1999.
- Barry . B. Brey , "Intel microprocessor ", Prentice Hall of India pvt ltd, 2000.
- Microcontroller 8051 Data Sheet Manual
- Database Management System by - Elmasri, Navathe
- Visual Basic Graphics Programming by - Rod Stevens

WEB SITES:

- www.microchip.com
- www.readplease.com

APPENDIX

Features

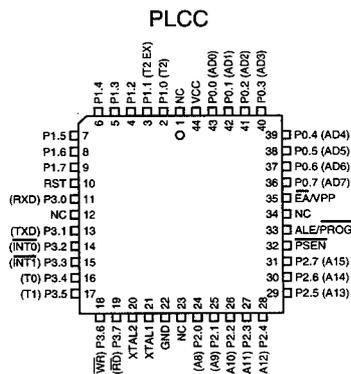
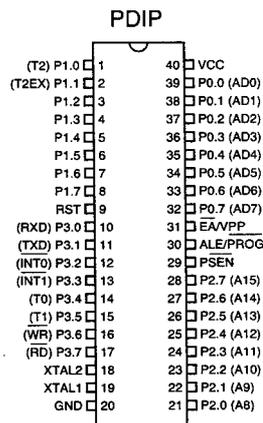
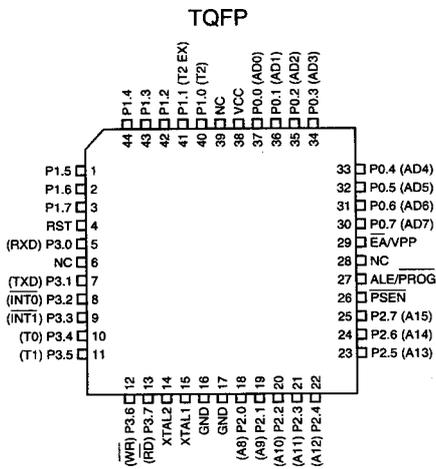
- Compatible with MCS-51™ Products
- 32K Bytes of Reprogrammable Flash Memory
- Endurance: 1000 Write/Erase Cycles
- 4V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 512 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Hardware Watchdog Timer
- Dual Data Pointer
- Power-off Flag

Description

The AT89C51RC is a low-power, high-performance CMOS 8-bit microcomputer with 32K bytes of Flash programmable read only memory and 512 bytes of RAM. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 and 80C52 instruction set and

(continued)

Pin Configurations



8-bit Microcontroller with 32K Bytes Flash

AT89C51RC

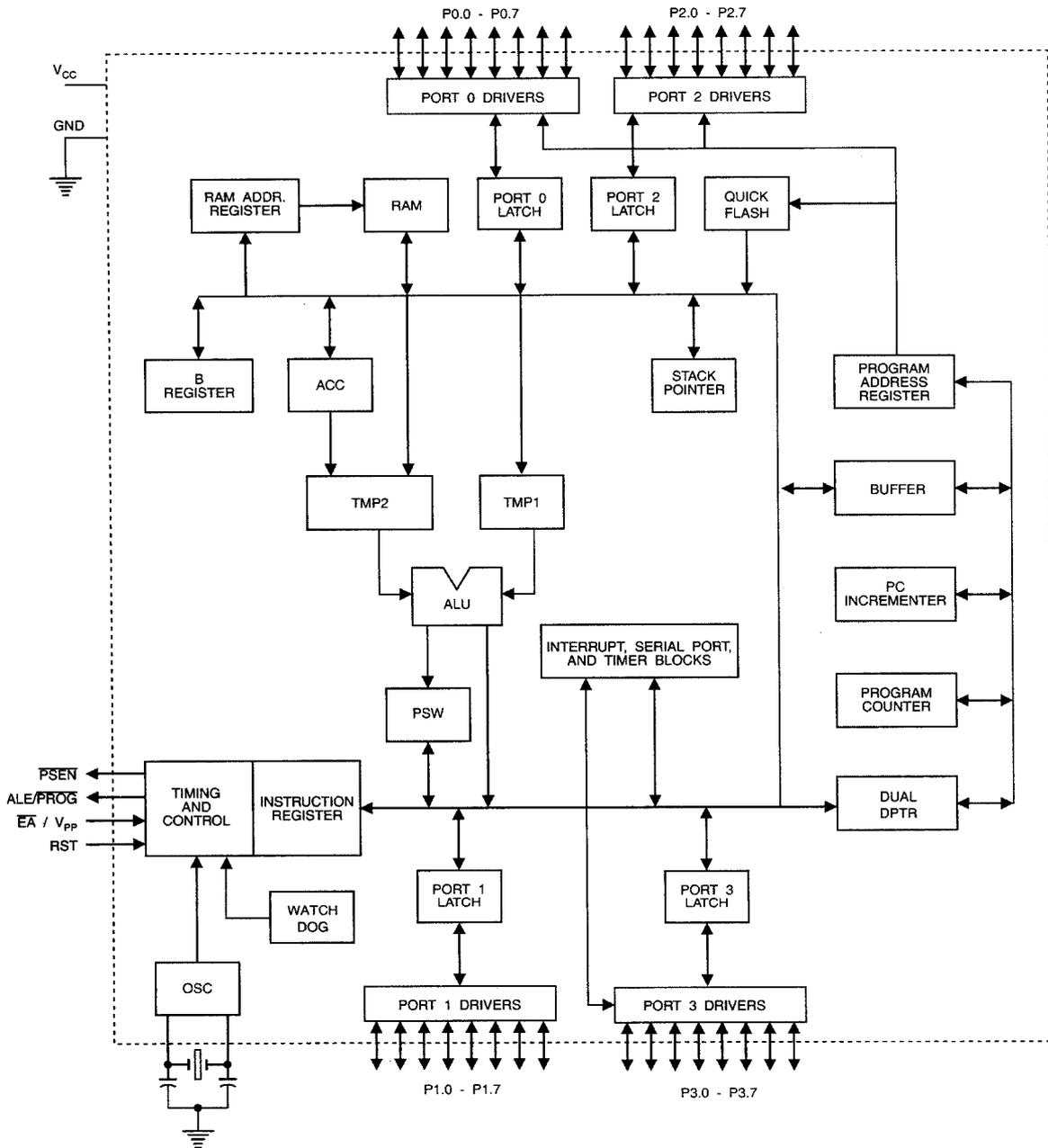




nout. The on-chip Flash allows the program memory to be user programmed by a conventional nonvolatile memory programmer. A total of 512 bytes of internal RAM are available in the AT89C51RC. The 256-byte expanded internal RAM is accessed via MOVX instructions after clearing bit 1 of the SFR located at address 8EH. The other 256-byte

RAM segment is accessed the same way as the Atmel AT89-series and other 8052-compatible products. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51RC is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Block Diagram



The AT89C51RC provides the following standard features: 2K bytes of Flash, 512 bytes of RAM, 32 I/O lines, three 8-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and lock circuitry. In addition, the AT89C51RC 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 external interrupt or hardware reset.

Pin Description

V_{CC}

Supply voltage.

3ND

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 pull-ups.

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

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. 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 pull-ups 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 pull-ups.

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)

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. 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 pull-ups 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 pull-ups.

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 pull-ups 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 pull-ups. 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 pull-ups 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 pull-ups.

Port 3 also serves the functions of various special features of the AT89C51RC, 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 DISTRO, the RESET HIGH out feature is enabled.

ALE/PROG

Address Latch Enable 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 is the read strobe to external program memory.

When the AT89C51RC 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. AT89C51RC 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				WDRST XXXXXXX	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 XXX00000	8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	PCON 0XXX0000	87H

Timer

NE/SA/SE555/SE555C

DESCRIPTION

The 555 monolithic timing circuit is a highly stable controller capable of producing accurate time delays, or oscillation. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For a stable operation as an oscillator, the free running frequency and the duty cycle are both accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output structure can source or sink up to 200mA.

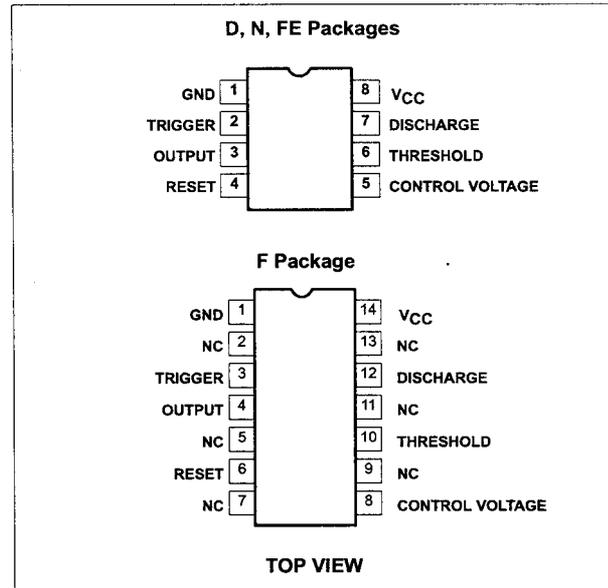
FEATURES

- Turn-off time less than 2µs
- Max. operating frequency greater than 500kHz
- Timing from microseconds to hours
- Operates in both astable and monostable modes
- High output current
- Adjustable duty cycle
- TTL compatible
- Temperature stability of 0.005% per °C

APPLICATIONS

- Precision timing
- Pulse generation
- Sequential timing
- Time delay generation
- Pulse width modulation

PIN CONFIGURATIONS



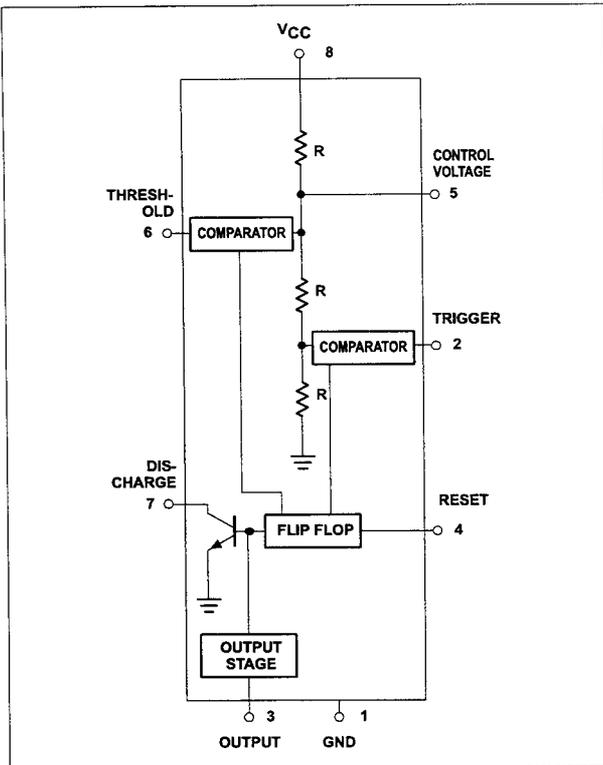
ORDERING INFORMATION

DESCRIPTION	TEMPERATURE RANGE	ORDER CODE	DWG #
8-Pin Plastic Small Outline (SO) Package	0 to +70°C	NE555D	0174C
8-Pin Plastic Dual In-Line Package (DIP)	0 to +70°C	NE555N	0404B
8-Pin Plastic Dual In-Line Package (DIP)	-40°C to +85°C	SA555N	0404B
8-Pin Plastic Small Outline (SO) Package	-40°C to +85°C	SA555D	0174C
8-Pin Hermetic Ceramic Dual In-Line Package (CERDIP)	-55°C to +125°C	SE555CFE	
8-Pin Plastic Dual In-Line Package (DIP)	-55°C to +125°C	SE555CN	0404B
14-Pin Plastic Dual In-Line Package (DIP)	-55°C to +125°C	SE555N	0405B
8-Pin Hermetic Cerdip	-55°C to +125°C	SE555FE	
14-Pin Ceramic Dual In-Line Package (CERDIP)	0 to +70°C	NE555F	0581B
14-Pin Ceramic Dual In-Line Package (CERDIP)	-55°C to +125°C	SE555F	0581B
14-Pin Ceramic Dual In-Line Package (CERDIP)	-55°C to +125°C	SE555CF	0581B

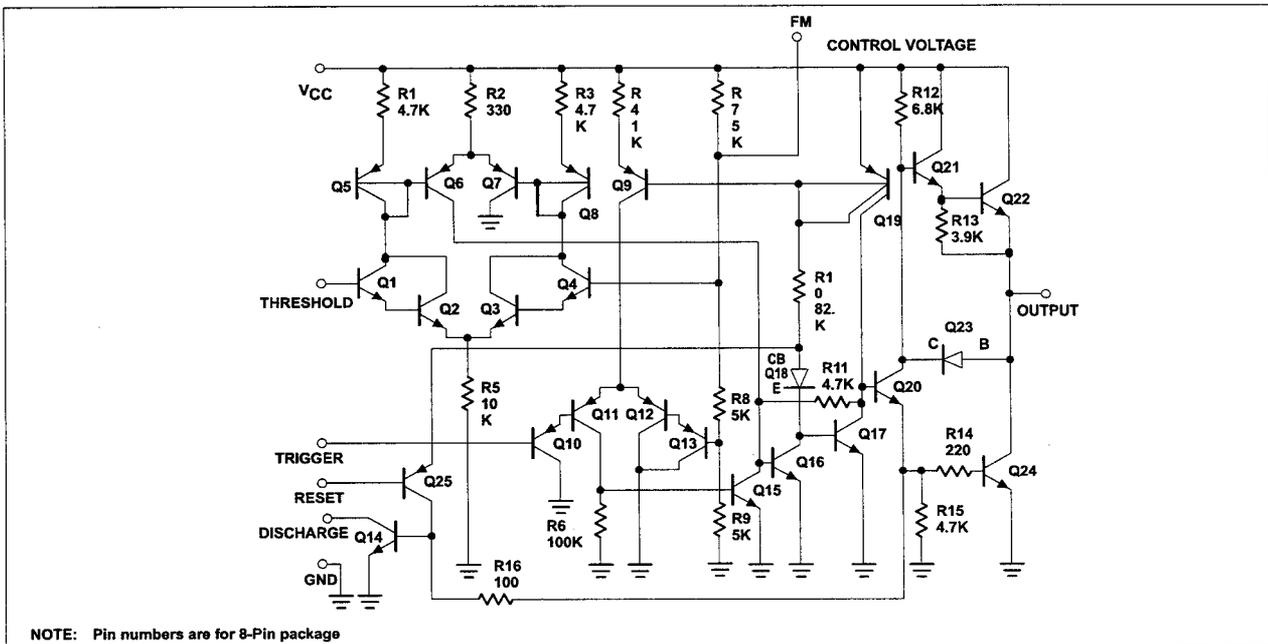
Timer

NE/SA/SE555/SE555C

BLOCK DIAGRAM



EQUIVALENT SCHEMATIC



NOTE: Pin numbers are for 8-Pin package

CD4017B, CD4022B Types

CMOS Counter/Dividers

High-Voltage Types (20-Volt Rating)

CD4017B—Decade Counter with
10 Decoded Outputs

CD4022B—Octal Counter with
8 Decoded Outputs

■ CD4017B and CD4022B are 5-stage and 4-stage Johnson counters having 10 and 8 decoded outputs, respectively. Inputs include a CLOCK, a RESET, and a CLOCK INHIBIT signal. Schmitt trigger action in the CLOCK input circuit provides pulse shaping that allows unlimited clock input pulse rise and fall times.

These counters are advanced one count at the positive clock signal transition if the CLOCK INHIBIT signal is low. Counter advancement via the clock line is inhibited when the CLOCK INHIBIT signal is high. A high RESET signal clears the counter to its zero count. Use of the Johnson counter configuration permits high-speed operation, 2-input decode-gating and spike-free decoded outputs. Anti-lock gating is provided, thus assuring proper counting sequence. The decoded outputs are normally low and go high only at their respective decoded time slot. Each decoded output remains high for one full clock cycle. A CARRY-OUT signal completes one cycle every 10 clock input cycles in the CD4017B or every 8 clock input cycles in the CD4022B and is used to

Features:

- Fully static operation
- Medium-speed operation . . . 10 MHz (typ.) at $V_{DD} = 10\text{ V}$
- Standardized, symmetrical output characteristics
- 100% tested for quiescent current at 20 V
- 5-V, 10-V, and 15-V parametric ratings
- Meets all requirements of JEDEC Tentative Standard No. 13A, "Standard Specifications for Description of 'B' Series CMOS Devices"

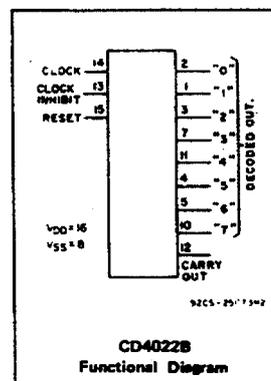
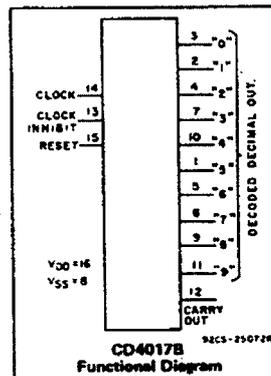
Applications:

- Decade counter/decimal decode display (CD4017B)
- Binary counter/decoder
- Frequency division
- Counter control/timers
- Divide-by-N counting

■ For further application information, see ICAN-6166 "COS/MOS MSI Counter and Register Design and Applications"

ripple-clock the succeeding device in a multi-device counting chain.

The CD4017B and CD4022B-series types are supplied in 16-lead hermetic dual-in-line ceramic packages (D and F suffixes), 16-lead dual-in-line plastic package (E suffix), 16-lead ceramic flat packages (K suffix), and in chip form (H suffix).



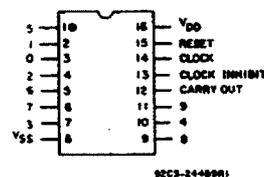
COMMERCIAL CMOS
HIGH VOLTAGE ICs

RECOMMENDED OPERATING CONDITIONS

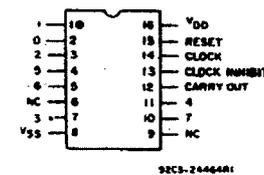
For maximum reliability, nominal operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTICS	V_{DD} (V)	LIMITS		UNITS
		Min.	Max.	
Supply-Voltage Range (For T_A = Full Package-Temperature Range)		3	18	V
Clock Input Frequency, f_{CL}	5	—	2.5	MHz
	10	—	5	
	15	—	5.5	
Clock Pulse Width, t_W	5	200	—	ns
	10	90	—	
	15	60	—	
Clock Rise & Fall Time, t_{rCL} , t_{fCL}	5	UNLIMITED*		
	10	UNLIMITED*		
	15	UNLIMITED*		
Clock Inhibit Setup Time, t_s	5	230	—	ns
	10	100	—	
	15	70	—	
Reset Pulse Width, t_{RW}	5	260	—	ns
	10	110	—	
	15	60	—	
Reset Removal Time, t_{rem}	5	400	—	ns
	10	280	—	
	15	150	—	

*Only if Pin 14 is used as the clock input. If Pin 13 is used as the clock input and Pin 14 is tied high (for advancing count on negative transition of the clock), rise and fall time should be $\leq 15\ \mu\text{s}$.



TOP VIEW
CD4017B
TERMINAL DIAGRAM



TOP VIEW
NC - no connection
CD4022B
TERMINAL DIAGRAM

CD4017B, CD4022B Types

