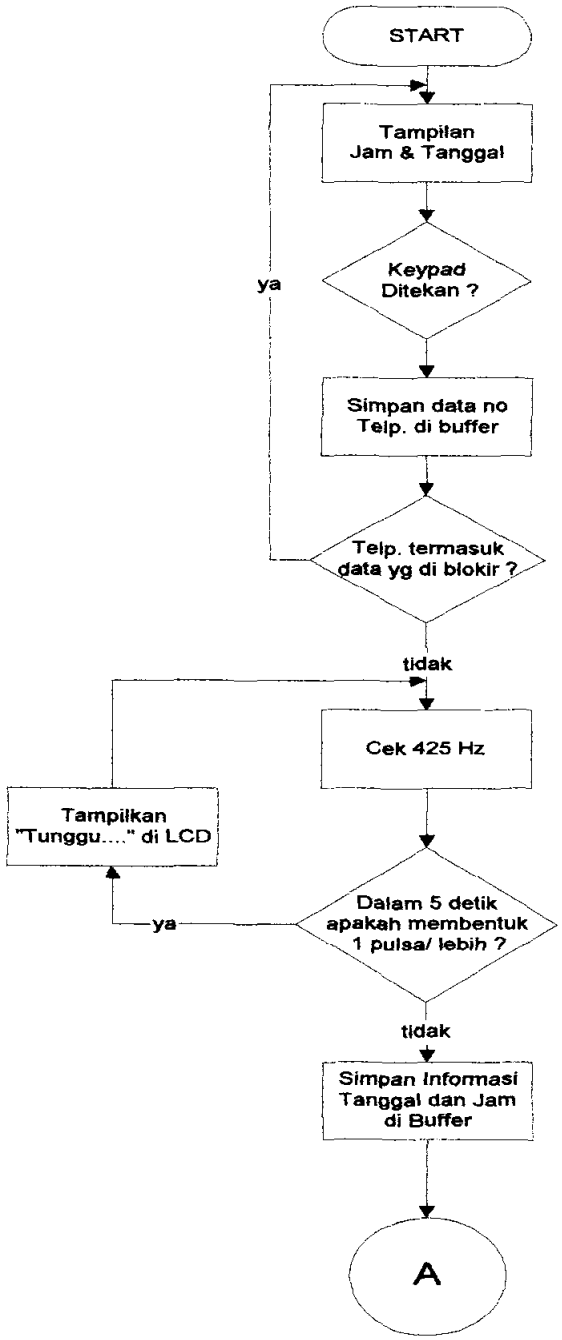
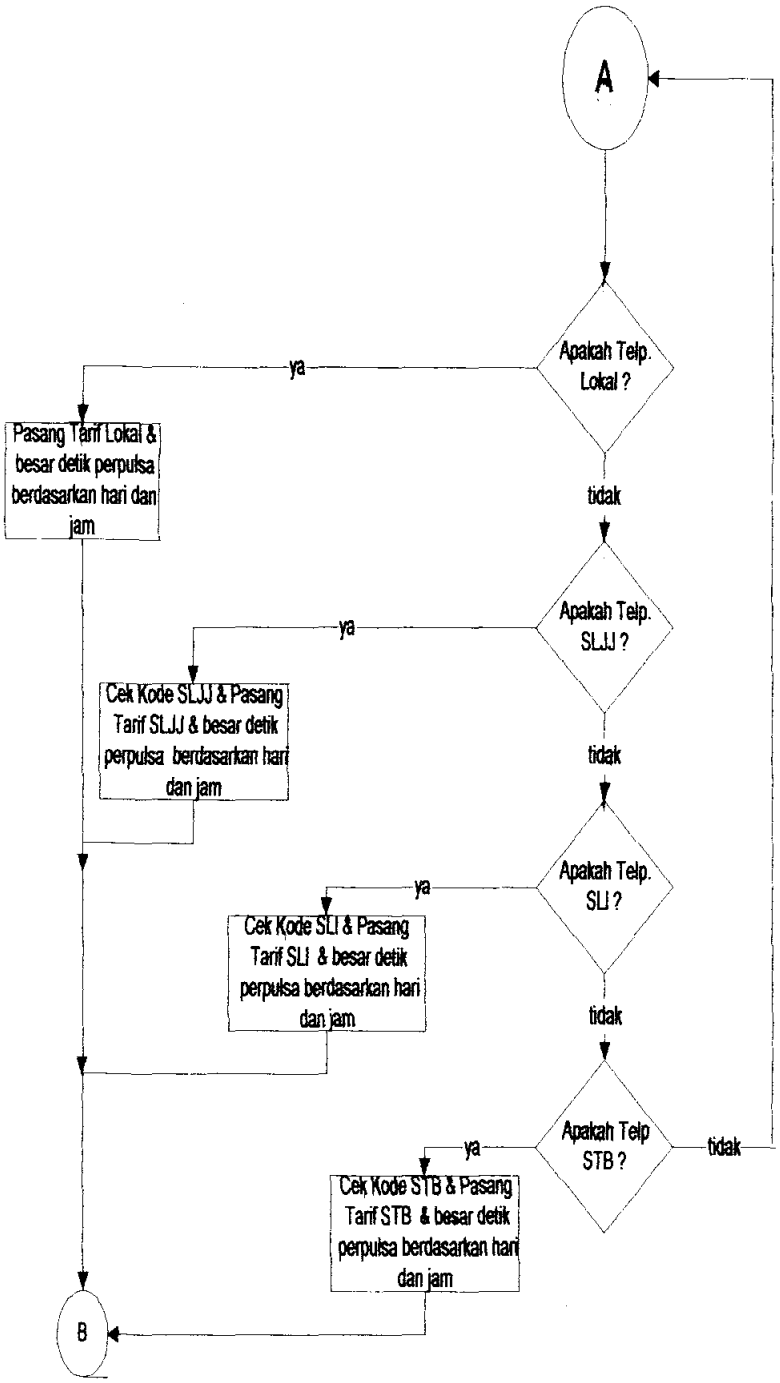
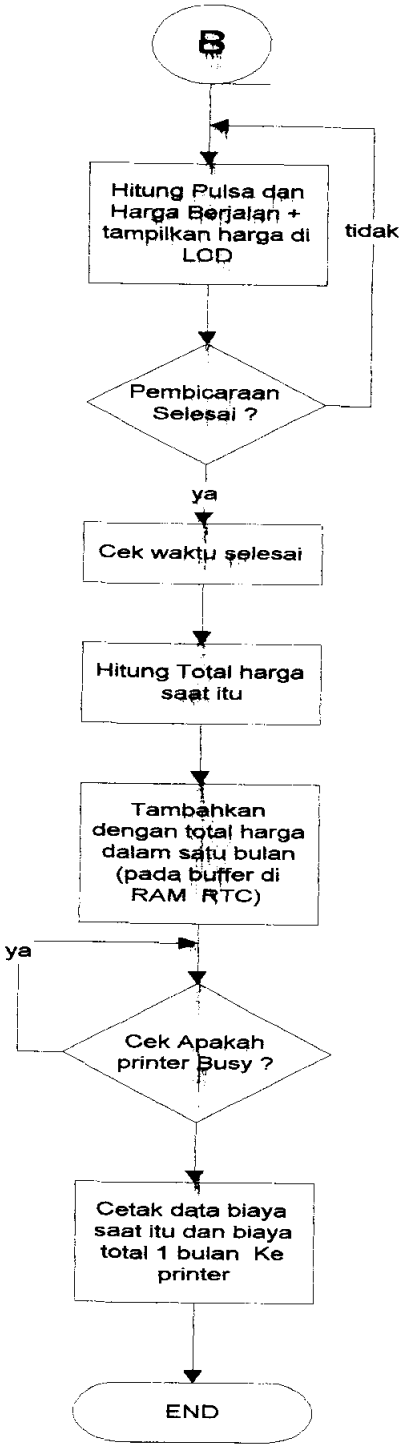
LAMPIRAN

BLOK DIAGRAM SOFTWARE




A
Apakah Telp. Lokal ?
ya
Pasang Tarif Lokal & besar detik perpulsa berdasarkan hari dan jam
tidak
Apakah Telp. SLJJ ?
ya
Cek Kode SLJJ & Pasang Tarif SLJJ & besar detik perpulsa berdasarkan hari dan jam
tidak
Apakah Telp. SLI ?
ya
Cek Kode SLI & Pasang Tarif SLI & besar detik perpulsa berdasarkan hari dan jam
tidak
Apakah Telp STB ?
ya
tidak
Cek Kode STB & Pasang Tarif STB & besar detik perpulsa berdasarkan hari dan jam
B


B
Hitung Pulsa dan Harga Berjalan + tampilkan harga di LCD
tidak
Pembicaraan Selesai ?
ya
Cek waktu selesai
Hitung Total harga saat itu
Tambahkan dengan total harga dalam satu bulan (pada buffer di RAM RTC)
ya
Cek Apakah printer Busy ?
Cetak data biaya saat itu dan biaya total 1 bulan Ke printer
END

```
r; Skripsi  : Penerapan uC MCS51 Pada Pencatat
Biaya Telepon
; Nama              : Valens Sigit Hascaryo

; NRP               : 5103097064
; Last Update       : Januari, 2001

;-------------------------------------------------------
--
;                PPI PIN & PPI ADDRESS
;-------------------------------------------------------
--
;PA0      = ACK              <--
;PA1      = Busy             <--
;PA2      = Paper End        <--
;PA3      = Select <--
;PA4      = Error  <--
;PB0      = Strobe-->
;PB1      = Auto Feeder  -->
;PB2      = Select In        -->
;PB3      = Init             -->
;PC0-PC7 = Data Bit<->
pa        equ      06000h
pctl      equ      06001h
pdata     equ      06002h
pcw       equ      06003h
cw        equ      10010000b          ;Control
Word / A=in / b=out / c=out

;-------------------------------------------------------
--
;                ARITMATIK ADDRESS
;-------------------------------------------------------
--
SizeX     equ      4                  ;24 bit!
Operand             equ      30h
Pembagi             equ      34h
HasilBagi equ      38h
SisaBagi  equ      3Ch
Pengali             EQU      Pembagi
HasilKali  EQU      HasilBagi

;-------------------------------------------------------
--
;    VARIABEL UNTUK TARIF PERCAKAPAN
BERDASARKAN WAKTU
;-------------------------------------------------------
--


;-------------------------------------------------------
--
;                REAL TIME CLOCK ADDRESS
;-------------------------------------------------------
--
RTC                          EQU      0000H
RTC_SEC                      EQU      RTC
ALARM_SEC                    EQU      RTC
RTC_min                      EQU      RTC
ALARM_MIN                    EQU      RTC
RTC_HOUR                     EQU      RTC
ALARM_HOUR                   EQU      RTC
RTC_DAY                      EQU      RTC
RTC_DATE                     EQU      RTC
RTC_MONTH                    EQU      RTC
RTC_YEAR                     EQU      RTC
REGA                         EQU      RTC
REGB                         EQU      RTC
REGC                         EQU      RTC
REGD                         EQU      RTC
DRREGA                       data     21H
        ;data reg. A
DRREGB                       data     22H
        ;data reg. B
DRREGC                       data     23H
        ;data reg. C
DRREGD                       data     24H
        ;data reg. D
RRAM                         EQU      RTC□
        ;internal RAM 114 bytes

EEPROM              equ      2000h
EEPROM2             equ      4000h
LCD       EQU       8000H    ;LCD Address
LCD0      EQU       LCD      ;LCD CONTROL
OPERATION
LCD1                EQU      LCD      ;LCD
DATA OPERATION
onoffhookbit        00h      ;on-off hook
relcutbit  bit      01h      ;relay cut
det425bit bit       02h      ;detect 425 Hz
digit_notelp        equ      56h
digit_notelp2       equ      2Ah

no_telepon          equ      42h
print_harga         equ      71h
pharga              equ      08h
detik_pulsa         equ      10h
detik_perpulsa      equ      57h
harga_perpulsa      equ      5bh
tambah_satu         equ      5fh
batas               equ      2eh
batascounter        equ      2fh
cek_jam             equ      2dh
cek_jaml  equ       2ch
cek_zona equ        29h
alamatbawah         equ      57h
alamatatas          equ      58h

;-------------------------------------------------------
--
;                89c51 PIN
;-------------------------------------------------------
--
;P1.0 - P1.3          = DTMF Signal input
;det425hzbit P1.4   ; 425Hz Detector
Relcut               BIT P1.5  ; Cut Telephone line
/ 0=off/ 1=on(default)
StD                  bit p1.6   :StD from 8870
/std=1=signal on/std=0=signal of
hookdet              bit p3.2
;P1.7      = ??
;Rx(P3.0) - Tx(P3.1) = Down. RTC & EEPROM /
Data Monitor to PC
;INT0(P3.2) = HOOK /1=off hook(ON)
/0=onhook(OFF)
;INT1(P3.3) = Interupt from RTC (optional)
;T0(P3.4)   = Interupt from 567 / Reverse Polarity
& 16Kc signal
;T1(P3.5)   = Intrupt from RTC Square Wave /
Time Counter
```

```
;-----------------------------------------------------
--
;                Register Function (BANK 0)
;-----------------------------------------------------
--
;R0      = pointer operand
;R1      = pointer pengali
;R2      = sizex
;R3      = sizex * 8
;R4      = delay tunggu
;R5      = pointer no telepon
;R6      = column position at 1st row (LCD)
;R7      = column position at 2nd row (LCD)

;-----------------------------------------------------
------
;                              INISIALISASI LCD
;-----------------------------------------------------
------
DISPCLR EQU 00000001B      ;DISPLAY CLEAR
FUNCSET EQU 00111000B      ;INTERFACE DATA
LENGTH :8 BITS
ENTRMOD EQU 00000110B      ;INCREMENT, NO
DISPLAY SHIFT
DISPON  EQU 00001100B      ;DISPLAY ON,
CURSOR OFF, BLINK OFF
CURSOR  EQU 00001110B      ;DISPLAY ON,
CURSOR ON, BLINK OFF
BLINK   EQU 00001101B      ;DISPLAY ON,
CURSOR OFF, BLINK ON

                ORG   00H
                AJMP  set_awal

                ORG   40H

;-----------------------------------------------------
--
;       Procedure : Multi Byte divider
;-----------------------------------------------------
--
Pembagian:
        MOV     R0,#HasilBagi
        ACALL   HapusNilai
        MOV     R0,#SisaBagi
        ACALL   HapusNilai
;
        MOV     R3,#SizeX*8
LoopPembagian:
        CLR     C
        MOV     R0,#Operand
        ACALL   GeserKiri1X
        MOV     R0,#SisaBagi
        ACALL   GeserKiri1X
;
        MOV     R0,#SisaBagi
        MOV     R1,#Pembagi
        ACALL   Perbandingan
        SisaBagi-Pembagi?
        JC      JanganDikurangi
        SisaBagi<Pembagi, skip!
;
        MOV     R0,#SisaBagi
        MOV     R1,#Pembagi
        ACALL   Pengurangan
        SisaBagi:=SisaBagi-Pembagi
JanganDikurangi:
        CPL     C
        MOV     R0,#HasilBagi
        Simpan hasil
        ACALL   GeserKiri1X
        DJNZ    R3,LoopPembagian
        RET

GeserKiri1X:
        MOV     R2,#SizeX
LeftShift:
        MOV     A,@R0
        RLC     A
        MOV     @R0,A
        INC     R0
        DJNZ    R2,LeftShift
        RET

;-----------------------------------------------------
--
;       Procedure : Multi Byte Multiplier
;-----------------------------------------------------
--
Perkalian:
        MOV     R0,#HasilKali
        ACALL   HapusNilai
;
        MOV     R3,#SizeX*8
LoopPerkalian:
        CLR     C
        MOV     R0,#Pengali□1
        MOV     R2,#SizeX
GeserKanan:
        MOV     A,@R0
        RRC     A
        MOV     @R0,A
        DEC     R0
        DJNZ    R2,GeserKanan

        JNC     JanganDitambah

        MOV     R0,#HasilKali
        MOV     R1,#Operand
        ACALL   Penambahan
JanganDitambah:
        CLR     C
        MOV     R0,#Operand
        ACALL   GeserKiri1X
        DJNZ    R3,LoopPerkalian
        RET

;-----------------------------------------------------
--
;       Procedure : Multi Byte Adder
;-----------------------------------------------------
--
Penambahan:
        CLR     C
        MOV     R2,#SizeX
LoopPenambahan:
        MOV     A,@R0
        ADDC    A,@R1
        MOV     @R0,A
        INC     R0
        INC     R1
        DJNZ    R2,LoopPenambahan
```

```
        RET
;-----------------------------------------------------
--
;       Procedure : Multi Byte Comparator
;-----------------------------------------------------
--
Perbandingan:
        CLR     C
        MOV     R2,#SizeX
LoopPerbandingan:
        MOV     A,@R0
        SUBB    A,@R1
        INC     R1
        INC     R0
        DJNZ    R2,LoopPerbandingan
        RET

;-----------------------------------------------------
--
;       Procedure : Multi Byte Substractor
;-----------------------------------------------------
--
Pengurangan:
        CLR     C
        MOV     R2,#SizeX
LoopPengurangan:
        MOV     A,@R0
        SUBB    A,@R1
        MOV     @R0,A
        INC     R0
        INC     R1
        DJNZ    R2,LoopPengurangan
        RET
;
; ***** Multi Byte Eraser
;
HapusNilai:
        MOV     R2,#SizeX
LoopHapus:
        MOV     @R0,#0
        INC     R0
        DJNZ    R2,LoopHapus
        RET

;-----------------------------------------------------
--
;       Procedure : Multi Byte Copier
;-----------------------------------------------------
--
Copy:
        MOV     R2,#SizeX
LoopCopy:
        MOV     A,@R0
        MOV     @R1,A
        INC     R0
        INC     R1
        DJNZ    R2,LoopCopy
        RET

;-----------------------------------------------------
--
;               Procedure LCD Display
;-----------------------------------------------------
--
POSISI2.1:
                MOV     A,R7
;KOLOM 1
POSISI2:
                ADD     A,#11000000B
;POSISI DI BARIS 2
                SJMP    POSISI.SUB

POSISI1.1:
                MOV     A,R6
;KOLOM 1
POSISI1:
                ADD     A,#10000000B
;POSISI DI BARIS 1

POSISI.SUB:
                DEC     A
;AWALAN POSISI KOLOM DIMULAI DARI 0
                ACALL   CONTROLOUT
;KIRIM SEBAGAI OPERASI KONTROL
                RET

PRINTSTRING2:                   ;CETAK
STRING DI BARIS 2 KOLOM 1
                ACALL POSISI2.1
;BARIS 2 KOLOM 1
                SJMP PRINTSTRING
;CETAK STRING

PRINTSTRING1:                   ;CETAK
STRING DI BARIS 1 KOLOM 1
                ACALL POSISI1.1         ;BARIS
1 KOLOM 1

PRINTSTRING:                    ;CETAK
STRING
                SJMP OUTSTRING
;AMBIL DATA DULU

PRINTSTRINGLOOP:
                ACALL DATAOUT
;KIRIM SEBAGAI OPERASI DATA
                INC DPTR
;POSISI DATA BERIKUTNYA

OUTSTRING:
                CLR A
;POINTER=0
                MOVC A,@A□
;AMBIL DATA BERDASARKAN DPTR
                JNZ PRINTSTRINGLOOP
;APAKAH MASIH ADA DATA BERIKUTNYA
                RET

CONTROLOUT:
                PUSH DPH
                PUSH DPL
                MOV DPTR,#LCD0
;ALAMAT OPERASI CONTROL LCD
                SJMP LCD.OUT            ;KIRIM
KE LCD

DATAOUT:
                PUSH DPH
                PUSH DPL
                MOV DPTR,#LCD1
;ALAMAT OPERASI DATA LCD
```

```
LCD.OUT:                        ;KIRIM KE LCD
                MOVX @DPTR,A
DELAY.LCD:
                MOV A,#250
                DJNZ A,$
                POP DPL
                POP DPH
                RET

DELAY.INIT.LCD:
                MOV R1,#020h
        ;20h
DLY.LCD.LP:
                MOV R2,#0
                DJNZ R2,$
                DJNZ R1,DLY.LCD.LP
                RET

INIT.LCD:
                MOV A,#DISPCLR
;DISPLAY CLEAR
                ACALL CONTROLOUT
                ACALL DELAY.INIT.LCD
                MOV A,#FUNCSET
;FUNCTION SET
                ACALL CONTROLOUT
                MOV A,#DISPON
;DISPLAY ON
                ACALL CONTROLOUT
                MOV A,#ENTRMOD
;ENTRY MODE
                ACALL CONTROLOUT
                RET

DELAY:
                 MOV  r1,#10
DEL:             MOV  r2,#30
              DJNZ  r2,$
                 DJNZ  r1,DEL
                 RET



vsh:      DB ' Valens Sigit H',0
wm:       DB ' Widya Mandala',0
tek9:     DB ' Tekan 9 ......',0
siltek:   DB ' Tekan no. tujuan',0
jam2:     DB 'Time :  ',0
senin:  DB 'Senin  ',0
selasa: DB 'Selasa ',0
rabu:   DB 'Rabu   ',0
kamis:  DB 'Kamis  ',0
jumat:  DB 'Jumat  ',0
sabtu:  DB 'Sabtu  ',0
minggu: DB 'Minggu ',0
wait1:    DB '  Tunggu.....',0

nama            db      'Valens Sigit
Hascaryo - 5103097064',0fh
garis           db      '-----------------------
------',0fh
tanggal         db      'Tanggal    : ',0fh
hari            db      'Hari       : ',0fh
jam             db      'Jam        : ',0fh
tujuan          db      'Tujuan     : ',0fh
jumpul          db      'Jumlah Pulsa : ',0fh
harga           db      'Harga      : Rp.
',0fh


asenin:  DB 'Senin  ',0fh
aselasa: DB 'Selasa ',0fh
arabu:   DB 'Rabu   ',0fh
akamis:  DB 'Kamis  ',0fh
ajumat:  DB 'Jumat  ',0fh
asabtu:  DB 'Sabtu  ',0fh
aminggu: DB 'Minggu ',0fh


;-----------------------------------------------------
---
;               Procedure Read Addres MAP
(RTC)
;-----------------------------------------------------
---
Read_sec mov    dptr,#00h
                movx    a,@dptr
                ret
Read_minmov     dptr,#02h
                movx    a,@dptr
                ret
Read_hour       mov     dptr,#04h
                movx    a,@dptr
                ret

read_day mov    dptr,#06h
                movx    a,@dptr
                ret

read_datemov    dptr,#07h
                movx    a,@dptr
                ret

read_month      mov     dptr,#08h
                movx    a,@dptr
                ret

read_yearmov    dptr,#09h
                movx    a,@dptr
                ret

;-----------------------------------------------------
---
;               Procedure Write Address MAP
(RTC)
;-----------------------------------------------------
---
write_sec mov   dptr,#00h
                movx    @dptr,a
                ret
write_minmov    dptr,#02h
                movx    @dptr,a
                ret
write_hour      mov     dptr,#04h
                movx    @dptr,a
                ret

write_daymov    dptr,#06h
                movx    @dptr,a
                ret
```

```
write_date      mov     dptr,#07h
                movx    @dptr,a
                ret

write_month     mov     dptr,#08h
                movx    @dptr,a
                ret

write_year      mov     dptr,#09h
                movx    @dptr,a
                ret

;-------------------------------------------------------
---
;               Procedure Read Register
A,B,C,D (RTC)
;-------------------------------------------------------
---
Read_Reg_A      mov     dptr,#0Ah
                movx    a,@dptr
                mov     DRRegA,a
                ret
Read_Reg_B      mov     dptr,#0Bh
                movx    a,@dptr
                mov     DRRegB,a
                ret
Read_Reg_C      mov     dptr,#0Ch
                movx    a,@dptr
                mov     DRRegC,a
                ret
Read_Reg_D      mov     dptr,#0Dh
                movx    a,@dptr
                mov     DRRegD,a
                ret
Read_All_Reg    acall   Read_Reg_A
                acall   Read_Reg_B
                acall   Read_Reg_C
                acall   Read_Reg_D
                ret

;-------------------------------------------------------
---
;               Procedure Write Register
A,B,C,D (RTC)
;-------------------------------------------------------
---
Write_Reg_A     mov     dptr,#0Ah
                mov     a,#DRRegA
                movx    @dptr,a
                ret
Write_Reg_B     mov     dptr,#0Bh
                mov     a,#DRRegB
                movx    @dptr,a
                ret
Write_Reg_C     mov     dptr,#0Ch
                mov     a,#DRRegC
                movx    @dptr,a
                ret
Write_Reg_D     mov     dptr,#0Dh
                mov     a,#DRRegD
                movx    @dptr,a
                ret
Write_All_Reg   acall   Write_Reg_A
                acall   Write_Reg_B
                acall   Write_Reg_C
                acall   Write_Reg_D
                ret

printROM:mov    A,#0h
                movc    A,@A□
                cjne    A,#0Fh,print
                Ajmp    printend
print           push    dph
                push    dpl
                push    a
                Acall   ngeprint  ;printit
                pop     a
                pop     dpl
                pop     dph
                inc     dptr
                Ajmp    printROM
printend  ret

ngeprint: mov   dptr,#pdata
                movx    @dptr,a
                nop
                nop
                nop
                nop
;               mov     dptr,#pctl
;               mov     a,#1
;               movx    @dptr,a
;               mov     a,#100
;               djnz    a,$
                mov     dptr,#pctl
                mov     a,#0
                movx    @dptr,a
                mov     a,#100
                djnz    a,$
                mov     dptr,#pctl
                mov     a,#1
                movx    @dptr,a
tunggux1:       mov     dptr,#pa
                movx    a,@dptr
                anl     a,#00000010b
                cjne    a,#00h,tunggux1
                nop
                nop
                ret

initprinter:    mov     A,#27
        ;command mode
                acall   ngeprint
                mov     A,#64
        ;reset printer
                acall   ngeprint
                ret

gantibaris mov  r7,#10
                mov     a,#13
                acall   ngeprint
                mov     a,#10
                acall   ngeprint
                ret


;-------------------------------------------------------
---
;                       Main Program
;-------------------------------------------------------
---
```

```
set_awal:
                mov  sp,#60H
                mov  R6,#01h
                mov  R7,#01h
                mov  20h,#00h
                mov  r4,#0ffh
tunggu_1:   acall delay
                djnz  r4,tunggu_1

                MOV  DRREGB,#00000010B
        ;Set register A&B RTC
                MOV  DRREGA,#00101111B

                acall     write_reg_a
                acall     write_reg_b
                ajmp      mulai_1

;-------------------------------------------------------
--
; Set waktu dan tanggal secara manual pada RTC
dengan mengubah
; register B bit ke 7 dengan '1'
;-------------------------------------------------------
--
                mov       a,#00h
        ;Set waktu RTC
                acall     write_sec
                mov       a,#10H
                acall     write_min
                mov       a,#01H
                acall     write_hour

                mov       a,#03h
        ;set tanggal RTC
                acall     write_day
                mov       a,#15h
                acall     write_date
                mov       a,#01h
                acall     write_month
                mov       a,#02h
                acall     write_year

;-------------------------------------------------------
--
; Penulisan Tanggal dan Waktu pada LCD dengan
pengaturan baris
; dan kolom
;-------------------------------------------------------
--
mulai_1:  mov       r3,#50
tunggu_2:   acall           delay
                djnz    r3,tunggu_2
                acall     init.lcd
                mov       detik_pulsa,#00h

mulai_3:  mov       r6,#01h
        ;Penulisan hari pada display
                acall     posisi1.1
                acall     read_day
                anl       a,#00001111b
                cjne      a,#01h,sen
                mov       dptr,#minggu
                acall     printstring1
                sjmp      ming
sen:            cjne      a,#02h,sel
                mov       dptr,#senin
                acall     printstring1
                sjmp      ming
sel:            cjne      a,#03h,rab
                mov       dptr,#selasa
                acall     printstring1
                sjmp      ming
rab:            cjne      a,#04h,kam
                mov       dptr,#rabu
                acall     printstring1
                sjmp      ming
kam:            cjne      a,#05h,jum
                mov       dptr,#kamis
                acall     printstring1
                sjmp      ming
jum:            cjne      a,#06h,sab
                mov       dptr,#jumat
                acall     printstring1
                sjmp      ming
sab:            cjne      a,#07h,sen
                mov       dptr,#sabtu
                acall     printstring1
ming:           mov       r6,#08h
        ;Penulisan tanggal pada Display
                acall     posisi1.1
                acall     read_date
                swap      a
                anl       a,#00001111b
                add       a,#30h
                acall     dataout

                mov       r6,#09h
                acall     posisi1.1
                acall     read_date
                anl       a,#00001111b
                add       a,#30h
                acall     dataout

                mov       r6,#0Ah
                acall     posisi1.1
                mov       a,#'-'
                acall     dataout

                mov       r6,#0Bh
        ;Penulisan bulan pada Display
                acall     posisi1.1
                acall     read_month
                swap      a
                anl       a,#00001111b
                add       a,#30h
                acall     dataout

                mov       r6,#0Ch
                acall     posisi1.1
                acall     read_month
                anl       a,#00001111b
                add       a,#30h
                acall     dataout

                mov       r6,#0dh
                acall     posisi1.1
                mov       a,#'-'
                acall     dataout
                sjmp      tollop4

tollop3:  ajmp      mulai_3
tollop4:  mov       r6,#0eh
```

```
                acall   posisi1.1
                mov     a,#""
                acall   dataout

                mov     r6,#0fh
        ;Penulisan tahun pada Display
                acall   posisi1.1
                acall   read_year
                swap    a
                anl     a,#00001111b
                add     a,#30h
                acall   dataout

                mov     r6,#10h
                acall   posisi1.1
                acall   read_year
                anl     a,#00001111b
                add     a,#30h
                acall   dataout

                mov     r7,#01h
                mov     dptr,#jam2
                acall   printstring2

                mov     r7,#09h
        ;Penulisan jam pada Display
                acall   posisi2.1
                acall   read_hour
                swap    a
                anl     a,#00001111b
                add     a,#30h
                acall   dataout

                mov     r7,#0ah
                acall   posisi2.1
                acall   read_hour
                anl     a,#00001111b
                add     a,#30h
                acall   dataout

                mov     r7,#0bh
                acall   posisi2.1
                mov     a,#':'
                acall   dataout

                mov     r7,#0ch
        ;Penulisan menit pada Display
                acall   posisi2.1
                acall   read_min
                swap    a
                anl     a,#00001111b
                add     a,#30h
                acall   dataout

                mov     r7,#0dh
                acall   posisi2.1
                acall   read_min
                anl     a,#00001111b
                add     a,#30h
                acall   dataout
                sjmp    tollop2

tollop1:  ajmp    tollop3
tollop2:  mov     r7,#0eh
                acall   posisi2.1
                mov     a,#':'
                acall   dataout
                mov     r7,#0fh
        ;Penulisan detik pada Display
                acall   posisi2.1
                acall   read_sec
                swap    a
                anl     a,#00001111b
                add     a,#30h
                acall   dataout
                mov     r7,#10h
                acall   posisi2.1
                acall   read_sec
                anl     a,#00001111b
                add     a,#30h
                acall   dataout
                jb      hookdet,lomrel2
                ajmp    mulai_3
tollop9:  ajmp    tollop1
;-----------------------------------------------------------
---
; Cek penekanan Keypad Telepon disertai
masukan/interupt berupa
; Cek 425Hz, Cek hook detector, Cek reverse
polarity, Cek 16Kc
; disertai tampilan pada LCD keypad yang ditekan.
; Memasukkan data nomor telepon pada alamat
42h-55h

;-----------------------------------------------------------
---
lomrel2:  mov     a,#dispclr
;Display clear
                acall   controlout
lomrel1:  mov     r6,#01h
                mov     dptr,#siltek
                acall   printstring1

                mov     r0,#no_telepon
;alamat awal penyimpanan no.telp.
delnotel:  mov    @r0,#0ffh
;kosongkan 20 byte buffer no.telp.
                inc     r0
                cjne    r0,#56h,delnotel
                mov     r0,#no_telepon
                mov     digit_notelp,#00h
                mov     digit_notelp2,#00h

                sjmp    tollop22
tollop20: ajmp    tollop9
tollop22:

mulai2:         mov     R7,#01h


start3:         acall   posisi2.1
read88701:      jnb     std,jbstd5
;Tunggu penekanan Keypad
                sjmp    jbstd4
jbstd5:         jnb     hookdet,tollop20
;Cek apakah telepon ditutup ?
                sjmp    read88701

start31:  acall   posisi2.1
read887011:     jnb     std,jbstd51

                sjmp    jbstd4
```

```
jbstd51:  jnb     hookdet,tollop20
          jnb     P1.4,lom425hz1
          ;Cek apakah ada nada 425Hz ?
          sjmp    read887011

jbstd4:   mov     a,p1
          ;Ambil data dari DTMF decoder
          anl     a,#$0f
          mov     @r0,a
          inc     r0
          inc     digit_notelp
          inc     digit_notelp2
          add     a,#30h
          cjne    a,#3Ah,lomk1
          ;Apakah menekan 0 ?
          mov     a,#30h

          sjmp    datout
lomk1:    cjne    a,#3Bh,lomk2
          ;Apakah menekan * ?
          mov     a,#2Ah
          sjmp    datout
lomk2:    cjne    a,#3Ch,datout
          ;Apakah menekan # ?
          mov     a,#23h
datout:   acall   dataout
          inc     R7
          cjne    R6,#16,lomk3

          mov     R6,#01H
bersih1:  mov     A,#dispclr
          acall   controlout

lomk3:
jbstd1:   jb      std,jbstd2
          ;Tunggu keypad dilepas
          sjmp    jbstd3
jbstd2:   jnb     hookdet,tollop20
          ;Cek apakah telepon ditutup ?
;         jnb     P1.4,lom425hz1
          ;Cek apakah ada nada 425Hz ?
          sjmp    jbstd1
jbstd3:   ajmp    start31
tollop6:  ajmp    tollop1
tollop7:

;-----------------------------------------------------------
---
; Detik 425Hz dalam counter 5 detik dengan
melihat apakah dalam
; 5 detik telah membentuk 1 kali dering, jika tidak
maka dianggap
; telepon telah tersambung ke nomor tujuan. Hal
ini mengingat
; dering 425Hz adalah 1 detik 'high' & 4 detik 'low'
;-----------------------------------------------------------
---
lom425hz1:
          mov     a,#dispclr
          acall   controlout
          mov     r6,#01h
          mov     dptr,#wait1
          acall   printstring1

          mov     40h,#0ffh

resdata1: mov     41h,#00h
          ;Set 31h pada 00h

ambildata1:       jb      P1.4,mark425on
          ;Cek tidak ada dering ?
          jnb     P1.4,mark425off
          ;Cek ada dering ?
mark425on:        setb    det425bit
          ;Tandai 02h dengan 1
          sjmp    tes4251
mark425off:       clr     det425bit
          ;Tandai 02h dengan 0
tes4251:
          jnb     hookdet,tollop6
          ;Cek apakah telepon ditutup ?
          acall   read_sec
          anl     a,#00001111b
          cjne    a,40h,lom4251
          ;Cek apakah sudah 1 detik ?
          sjmp    ambildata1
lom4251:  mov     40h,a
          ;Sudah mencapai 1 dering
          PUSH    A
          inc     41h
          mov     a,41h
cek0:     cjne    a,#01h,cek1
          jnb     det425bit,resdata1
          sjmp    terus1
cek1:     cjne    a,#02h,cek2
          jnb     det425bit,resdata1
          sjmp    terus1
cek2:     cjne    a,#03h,cek3
          jnb     det425bit,resdata1
          sjmp    terus1
cek3:     cjne    a,#04h,cek4
          jnb     det425bit,resdata1
          sjmp    terus1
cek4:     cjne    a,#05h,cek5
          jnb     det425bit,resdata1
          sjmp    terus1
cek5:     cjne    a,#06h,cek6
          jnb     det425bit,resdata1
          sjmp    terus1
cek6:     cjne    a,#07h,cek7
cek7:     jb      det425bit,bitlom2
terus1:   pop     a
;         setb    det425bit
          sjmp    ambildata1
tollop60: ajmp    tollop6


;-----------------------------------------------------------
---
; Penampilan Kembali No. Telp. pada baris satu
; Penghitungan durasi pulsa percakapan dan
harga yang ditampilkan
; pada layar LCD berserta lama pembicaraan
;-----------------------------------------------------------
---
bitlom2:  mov     40h,#00h
          mov     a,#dispclr
          acall   controlout
          mov     r6,#1
          mov     r0,#no_telepon
tulisbar1:
```

```
                acall   posisi1.1  ;tulis no.
telp. dari buffer
                mov     a,@r0
                add     a,#30h
                cjne    a,#3Ah,lomk_1
        ;Apakah menekan 0 ?
                mov     a,#30h
                sjmp    lanjut_x1
lomk_1:         cjne    a,#3Bh,lomk_2
        ;Apakah menekan * ?
                mov     a,#2Ah
                sjmp    lanjut_x1
lomk_2:         cjne    a,#3Ch,lanjut_x1
        ;Apakah menekan # ?
                mov     a,#23h
lanjut_x1:inc   r0
                inc     r6
                dec     digit_notelp
                mov     r5,digit_notelp
                acall   dataout
                cjne    r5,#00h,tulisbar1
                acall   posisi1.1
                mov     a,@r0
                add     a,#30h
                cjne    a,#3Ah,lomk_14
        ;Apakah menekan 0 ?
                mov     a,#30h
                sjmp    lanjut_x14
lomk_14: cjne   a,#3Bh,lomk_24
        ;Apakah menekan * ?
                mov     a,#2Ah
                sjmp    lanjut_x14
lomk_24: cjne   a,#3Ch,lanjut_x14  ;Apakah
menekan # ?
                mov     a,#23h
lanjut_x14:     acall   dataout

;-----------------------------------------------------
-------
;               Simpan Waktu Tanda
Percakapan Mulai
;-----------------------------------------------------
-------

                acall   read_hour
                swap    a
                anl     a,#00001111b
                add     a,#30h
                mov     78h,a

                acall   read_hour
                anl     a,#00001111b
                add     a,#30h
                mov     79h,a
                mov     a,#':'
                mov     7Ah,a

                acall   read_min
                swap    a
                anl     a,#00001111b
                add     a,#30h
                mov     7bh,a

                acall   read_min
                anl     a,#00001111b
                add     a,#30h
                mov     7ch,a

                mov     a,#':'
                mov     7dh,a

                acall   read_sec
                swap    a
                anl     a,#00001111b
                add     a,#30h
                mov     7eh,a

                acall   read_sec
                anl     a,#00001111b
                add     a,#30h
                mov     7fh,a

                mov     R0,#detik_perpulsa
                acall   HapusNilai

                mov     r0,#harga_perpulsa
                acall   hapusnilai
                mov     r0,#tambah_satu
                acall   hapusnilai


;-----------------------------------------------------
-------
;               Cek harga berdasarkan jam
;-----------------------------------------------------
-------
                lcall   read_hour
                cjne    a,#00h,cj1
                mov     cek_jam,#04h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj1:            cjne    a,#01h,cj2
                mov     cek_jam,#04h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj2:            cjne    a,#02h,cj3
                mov     cek_jam,#04h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj3:            cjne    a,#03h,cj4
                mov     cek_jam,#04h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj4:            cjne    a,#04h,cj5
                mov     cek_jam,#04h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj5:            cjne    a,#05h,cj6
                mov     cek_jam,#04h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj6:            cjne    a,#06h,cj7
                mov     cek_jam,#04h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj7:            cjne    a,#07h,cj8
                mov     cek_jam,#02h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj8:            cjne    a,#08h,cj9
                mov     cek_jam,#03h
                mov     cek_jaml,#01h
```

```
                ljmp    cekzend
cj9:            cjne    a,#09h,cj10
                mov     cek_jam,#03h
                mov     cek_jaml,#02h
                ljmp    cekzend
cj10:           cjne    a,#10h,cj11
                mov     cek_jam,#03h
                mov     cek_jaml,#02h
                ljmp    cekzend
cj11:           cjne    a,#11h,cj12
                mov     cek_jam,#03h
                mov     cek_jaml,#02h
                ljmp    cekzend
cj12:           cjne    a,#12h,cj13
                mov     cek_jam,#03h
                mov     cek_jaml,#02h
                ljmp    cekzend
cj13:           cjne    a,#13h,cj14
                mov     cek_jam,#03h
                mov     cek_jaml,#02h
                ljmp    cekzend
cj14:           cjne    a,#14h,cj15
                mov     cek_jam,#03h
                mov     cek_jaml,#02h
                ljmp    cekzend
cj15:           cjne    a,#15h,cj16
                mov     cek_jam,#03h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj16:           cjne    a,#16h,cj17
                mov     cek_jam,#03h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj17:           cjne    a,#17h,cj18
                mov     cek_jam,#03h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj18:           cjne    a,#18h,cj19
                mov     cek_jam,#03h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj19:           cjne    a,#19h,cj20
                mov     cek_jam,#02h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj20:           cjne    a,#20h,cj21
                mov     cek_jam,#01h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj21:           cjne    a,#21h,cj22
                mov     cek_jam,#01h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj22:           cjne    a,#22h,cj23
                mov     cek_jam,#01h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj23:           cjne    a,#23h,cj24
                mov     cek_jam,#04h
                mov     cek_jaml,#01h
                ljmp    cekzend
cj24:           mov     cek_jam,#04h
                mov     cek_jaml,#01h

cekzend:
```

```
;-----------------------------------------------------------
-------
;                       Cek harga LOKAL, SJLL, SLI,
STB
;-----------------------------------------------------------
-------
                        mov     r0,#no_telepon
                        mov     a,@r0
                        add     a,#30h

                        cjne    a,#3Ah,lokal1
                ;cek 0
                        inc     r0

                        mov     a,@r0
                        add     a,#30h

                        cjne    a,#38h,ceksljj
                ;cek 08
                        ajmp    cekhp

ceksljj:        cjne    a,#3Ah,taripsljj    ;cek 00
sli:                    mov     harga_perpulsa,#94
taripsli:       inc     r0
                        mov     a,@r0
                        mov     b,#1000
                        mul     ab
                        mov     alamatBAWAH,a
                        mov     alamatATAS,B
terus2i:        inc     r0
                        mov     a,@r0
                        mov     b,#100
                        mul     ab
                        add     a,alamatBAWAH
                        mov     alamatBAWAH,a
                        MOV     A,B
                        addC    A,alamatATAS
                        mov     alamatATAS,A
terus3i                 inc     r0
                        mov     a,@r0
                        mov     b,#10
                        mul     ab
                        add     a,alamatBAWAH
                        mov     alamatBAWAH,a
                        MOV     A,B
                        addC    A,alamatATAS
                        mov     alamatATAS,A
terus4i                 inc     r0
                        mov     a,@r0
                        add     a,alamatBAWAH
                        mov     alamatBAWAH,a
                        MOV     A,B
                        addC    A,alamatATAS
                        mov     alamatATAS,A
                        MOV     A,ALAMATBAWAH
                        MOV     DPL,A
                        MOV     A,ALAMATATAS
                        ADD     A,#$20
                        MOV     DPH,A
                        movx    a,@dptr
ceksljj33x1i:           cjne    a,#03h,ceksljj33xi
                        mov     cek_zona,#03h
                        ljmp    outzonai
ceksljj33xi:            cjne    a,#02h,ceksljj37i
                        mov     cek_zona,#02h
                        ljmp    outzonai
```

```
ceksljj37i:mov   cek_zona,#01h
                 ljmp    outzonai

lokal1:          mov
      harga_perpulsa,#167
                 mov     a,cek_jaml
                 cjne    a,#02h,cekjl1t
                 mov     batas,#78h
                 ljmp    xaxaxa
cekjl1t:  mov    batas,#0b4h
                 ljmp    xaxaxa

cekhp:           mov     harga_perpulsa,#41
                 mov     batas,#6
                 ljmp    xaxaxa

taripsljj:  inc  r0
                 mov     a,@r0
                 mov     b,#1000
                 mul     ab
                 mov     alamatBAWAH,a
                 mov     alamatATAS,B
terus2:          inc     r0
                 mov     a,@r0
                 mov     b,#100
                 mul     ab
                 add     a,alamatBAWAH
                 mov     alamatBAWAH,a
                 MOV     A,B
                 addC    A,alamatATAS
                 mov     alamatATAS,A
terus3:          inc     r0
                 mov     a,@r0
                 mov     b,#10
                 mul     ab
                 add     a,alamatBAWAH
                 mov     alamatBAWAH,a
                 MOV     A,B
                 addC    A,alamatATAS
                 mov     alamatATAS,A
terus4:          inc     r0
                 mov     a,@r0
                 add     a,alamatBAWAH
                 mov     alamatBAWAH,a
                 MOV     A,B
                 addC    A,alamatATAS
                 mov     alamatATAS,A
                 MOV     A,ALAMATBAWAH
                 MOV     DPL,A
                 MOV     A,ALAMATATAS
                 ADD     A,#$20
                 MOV     DPH,A
                 movx    a,@dptr
ceksljj33x1:     cjne    a,#03h,ceksljj33x
                 mov     cek_zona,#03h
                 ljmp    outzona
ceksljj33x:      cjne    a,#02h,ceksljj37
                 mov     cek_zona,#02h
                 ljmp    outzona
ceksljj37: mov   cek_zona,#01h
                 ljmp    outzona

outzona:         mov
      harga_perpulsa,#134
                 mov     a,cek_zona
                 cjne    a,#01h,cekzona2x
                 ljmp    cekzona1
cekzona2x:       cjne    a,#02h,cekzona3
                 ljmp    cekzona2

cekzona3:mov     a,cek_jam
                 cjne    a,#01h,tarsljjx1
                 mov     batas,#8
                 ljmp    xaxaxa
tarsljjx1:  cjne a,#02h,tarsljjx2
                 mov     batas,#4
                 ljmp    xaxaxa
tarsljjx2:  cjne a,#03h,tarsljjx3
                 mov     batas,#3
                 ljmp    xaxaxa
tarsljjx3:  mov  batas,#16
                 ljmp    xaxaxa

cekzona1:mov     a,cek_jam
                 cjne    a,#01h,tarsljjy12
                 mov     batas,#14
                 ljmp    xaxaxa
tarsljjy12:cjne  a,#02h,tarsljjy22
                 mov     batas,#7
                 ljmp    xaxaxa
tarsljjy22:cjne  a,#03h,tarsljjy32
                 mov     batas,#6
                 ljmp    xaxaxa
tarsljjy32:mov   batas,#28
                 ljmp    xaxaxa

cekzona2:mov     a,cek_jam
                 cjne    a,#01h,tarsljjz121
                 mov     batas,#10
                 ljmp    xaxaxa
tarsljjz121:     cjne    a,#02h,tarsljjz221
                 mov     batas,#5
                 ljmp    xaxaxa
tarsljjz221:     cjne    a,#03h,tarsljjz321
                 mov     batas,#4
                 ljmp    xaxaxa
tarsljjz321:     mov     batas,#20
                 ljmp    xaxaxa

outzonai:
xaxaxa:

;-------------------------------------------------------
;          Hitung Harga Berdasarkan Pulsanya
;-------------------------------------------------------

                 mov     tambah_satu,#01h
cek1det1x:
                 mov     r0,#detik_perpulsa
                 mov     r1,#tambah_satu
                 lcall   penambahan

                 mov     R0,#Operand
                 lcall   HapusNilai
          ;noi-kan operand

                 mov     r0,#harga_perpulsa
          ;operand<harga perpulsa
                 mov     r1,#operand
                 lcall   copy
```

```
                mov     R0,#Pengali
                lcall   HapusNilai
        ;nol-kan pengali

                mov     r0,#detik_perpulsa
        ;pengali<detik perpulsa
                mov     r1,#pengali
                lcall   copy
                lcall   Perkalian
        ;hasil kali = oper. x peng.

                mov     r7,#16
                mov     r0,#hasilkali
                mov     r1,#operand
                lcall   copy
terus12:  mov   pembagi,#10
                lcall   pembagian

                lcall   posisi2.1
                mov     a,3ch
                push    a
                mov     A,hasilkali
                cjne    A,#00h,lanjut12
                ajmp    kk
lanjut12: pop   a
                add     a,#30h
                lcall   dataout
                dec     r7
                mov     r0,#hasilbagi
                mov     r1,#operand
                lcall   copy
                sjmp    terus12
kk:             pop     a
                add     a,#30h
                lcall   dataout
                dec     r7
                lcall   posisi2.1
                mov     a,#' '
                lcall   dataout
                dec     r7
                lcall   posisi2.1
                mov     a,#'.'
                lcall   dataout
                dec     r7
                lcall   posisi2.1
                mov     a,#'p'
                lcall   dataout
                dec     r7
                lcall   posisi2.1
                mov     a,#'R'
                lcall   dataout
                inc     detik_pulsa
                mov     batascounter,#00h
        ;batascounter

cek1det1:
                jb      hookdet,colot
        ;Cek apakah telepon ditutup ?
                ljmp    habis1
colot:          lcall   read_sec
        ;
                anl     a,#00001111b
                cjne    a,40h,cek_2
        ;Cek apakah sudah 1 detik ?
                ljmp    cek1det1
cek_2:          mov     40h,a
        ;Sudah mencapai 1 dering
                inc     batascounter
        ;batascounter
                mov     a,batas
                cjne
        a,batascounter,cek1det1
        ;66=batascounter
                ljmp    cek1det1x

habis1:
        ;ngeprint data
;               jnb     hookdet,habis1


;-----------------------------------------------------------
-------
; Mencetak Ke Printer hasil dari laporan
percakapan dengan telepon
;-----------------------------------------------------------
-------

                mov     r4,#250
del1xx:         mov     r5,#250
                djnz    r5,$
                djnz    r4,del1xx

                mov     dptr,#pcw
                mov     a,#cw
                movx    @dptr,a

                lcall   initprinter

                mov     A,#10
        ;line feed
                lcall   ngeprint
                mov     A,#10
        ;line feed
                lcall   ngeprint


                mov     r4,#250
del1x:          mov     r5,#250
                djnz    r5,$
                djnz    r4,del1x

;               mov     r0,#no_telepon
                mov     dptr,#nama
                lcall   printrom
                lcall   gantibaris

                mov     dptr,#garis
                lcall   printrom
                lcall   gantibaris

                mov     dptr,#tanggal
                lcall   printrom
                lcall   read_date
                swap    a
                anl     a,#00001111b
                add     a,#30h
                lcall   ngeprint

                lcall   read_date
                anl     a,#00001111b
```

```
                add     a,#30h
                lcall   ngeprint

                mov     a,#':'
                lcall   ngeprint

                lcall   read_month
                swap    a
                anl     a,#00001111b
                add     a,#30h
                lcall   ngeprint

                lcall   read_month
                anl     a,#00001111b
                add     a,#30h
                lcall   ngeprint

                mov     a,#':'
                lcall   ngeprint

                mov     a,#'"
                lcall   ngeprint

                lcall   read_year
                swap    a
                anl     a,#00001111b
                add     a,#30h
                lcall   ngeprint
                lcall   read_year
                anl     a,#00001111b
                add     a,#30h
                lcall   ngeprint
                lcall   gantibaris

                mov     dptr,#hari
        ;hari
                lcall   printrom
                lcall   read_day
                anl     a,#00001111b
                cjne    a,#01h,asenxx
                mov     dptr,#aminggu
                lcall   printrom
                ljmp    amingxx
asenxx:         cjne    a,#02h,aselxx
                mov     dptr,#asenin
                ljmp    amingxx
aselxx:         cjne    a,#03h,arabxx
                mov     dptr,#aselasa
                ljmp    amingxx
arabxx:         cjne    a,#04h,akamxx
                mov     dptr,#arabu
                ajmp    amingxx
akamxx:         cjne    a,#05h,ajumxx
                mov     dptr,#akamis
                ajmp    amingxx
ajumxx:         cjne    a,#06h,asabxx
                mov     dptr,#ajumat
                ajmp    amingxx
asabxx:         cjne    a,#07h,asenxx
                mov     dptr,#asabtu
amingxx: lcall  printrom
                lcall   gantibaris


                mov     dptr,#jam
        ;jam
                lcall   printrom
                mov     a,78h
                lcall   ngeprint
                mov     a,79h
                lcall   ngeprint
                mov     a,7ah
                lcall   ngeprint
                mov     a,7bh
                lcall   ngeprint
                mov     a,7ch
                lcall   ngeprint
                mov     a,7dh
                lcall   ngeprint
                mov     a,7eh
                lcall   ngeprint
                mov     a,7fh
                lcall   ngeprint
                lcall   gantibaris

                mov     dptr,#tujuan
                lcall   printrom
                mov     r0,#no_telepon

tulisbar1p:     mov     a,@r0
        ;tulis no. telp. dari buffer
                add     a,#30h
                cjne    a,#3Ah,lomk_1p
        ;Apakah menekan 0 ?
                mov     a,#30h
                sjmp    lanjut_x1p
lomk_1p: cjne   a,#3Bh,lomk_2p
        ;Apakah menekan * ?
                mov     a,#2Ah
                sjmp    lanjut_x1p
lomk_2p: cjne   a,#3Ch,lanjut_x1p  ;Apakah
menekan # ?
                mov     a,#23h
lanjut_x1p:     inc     r0
                dec     digit_notelp2
                mov     r5,digit_notelp2
                lcall   ngeprint
                cjne    r5,#01h,tulisbar1p
                mov     a,@r0
                add     a,#30h
                cjne    a,#3Ah,lomk_14p
        ;Apakah menekan 0 ?
                mov     a,#30h
                sjmp    lanjut_x14p
lomk_14p:       cjne    a,#3Bh,lomk_24p
        ;Apakah menekan * ?
                mov     a,#2Ah
                sjmp    lanjut_x14p
lomk_24p:       cjne    a,#3Ch,lanjut_x14p
        ;Apakah menekan # ?
                mov     a,#23h
lanjut_x14p:    lcall   ngeprint
                lcall   gantibaris


                mov     dptr,#jumpul
                lcall   printrom
                mov     a,detik_perpulsa
                mov     b,#100
                div     ab
                add     a,#$30
                lcall   ngeprint
```

```
                mov     a,b
                mov     b,#10
                div     ab
                add     a,#$30
                lcall   ngeprint
                mov      a,b
                add     a,#$30      ;ascii
satuan disimpan di Acc
                lcall   ngeprint
                lcall   gantibaris

                mov     pharga,#71h
                mov     dptr,#harga
                lcall   printrom
                mov     r0,#harga_perpulsa
                mov     r1,#operand
                lcall   copy
                mov     R0,#Pengali
                lcall   HapusNilai
                ;nol-kan pengali
                mov     r0,#detik_perpulsa
                mov     r1,#pengali
                lcall   copy
                lcall   Perkalian

                mov     r0,#hasilkali
                mov     r1,#operand
                lcall   copy
pterus12: mov   pembagi,#10
                lcall   pembagian
                mov     a,3ch
                push    a
                mov     A,hasilkali
                cjne    A,#00h,planjut12
                ajmp    pkk
planjut12:pop   a
                add     a,#30h
                push    a
                mov     r0,pharga
                mov     @r0,a
                inc     pharga
                pop     a
;               lcall   ngeprint
                mov     r0,#hasilbagi
                mov     r1,#operand
                lcall   copy
                sjmp    pterus12
pkk:            pop     a
                add     a,#30h
                mov     r0,pharga
                mov     @r0,a
pulangp1:mov    r0,pharga
                mov     a,@r0
                lcall   ngeprint
                dec     pharga
                mov     r1,pharga
                cjne    r1,#70h,pulangp1
                lcall   gantibaris
                mov     A,#10
        ;line feed
                lcall   ngeprint

habis4:
                jnb     hookdet,habis4
```

```
Name     : SLJJ.lst
Function : SLJJ Identifier
Location : Surabaya
Made by  : Valens Sigit Hacaryo
Date     : XX-XX-01
Max Area : 900 Kode Area
begin
031      Surabaya          0     ;0 berarti
lokal
0986     MANOKWARI         3
0984     NABIRE            3
0983     SERUI             3
0981     BIAK              3
0980     RANSIKI           3
0979     FREEPORT          3
0975     TANAHMERAH        3
0974     BADE              3
0971     MERAUKE           3
0969     WAMENA            3
0967     ABEPURA           3
0966     SARMI             3
0961     MARAU             3
0957     KAIMANA           3
0956     FAKFAK            3
0955     BINTUNI           3
0952     TEMINABUAN        3
0951     SORONG            3
0929     MANGOLE           3
0928     LAIWUI            3
0927     LABUHA            3
0925     WEDA              3
0924     TOBELO            3
0923     MOROTAI           3
0922     JAILOLO           3
0921     SOASIU            3
0918     LARAT             3
0917     DOBO              3
0916     TUAL              3
0915     BULA              3
0914     MASOHI            3
0913     NAMLEA            3
0912     SAPARUA           3
0911     AMBON             3
0910     NEIRA             3
0902     AGATS             3
0901     KUALAKENCANA      3
0779     TANJUNGBATU       3
0778     BATAM CORE        3
0777     TANJUNGBALAIKAR   3
0776     DABOSINGKEP       3
0773     RANAI             3
0772     TEREMPA           3
0771     KIJANG            3
0769     AIRMOLEK          3
0768     TEMBILAHAN        3
0767     BAGANSIAPIAPI     3
0766     BENGKALIS         3
0765     BAGANBATU         3
0764     SIAKSRIINDRAPUR   3
0763     SELATPANJANG      3
0762     BANGKINANG        3
0761     PBR-ARENGKA       3
0760     TALUKKUANTAN      3
0759     MUARASIBERUT      3
0757     BALAISALASA       3
0756     PAINAN            3
0755     ALAHANPANJANG     3
0754     SAWAHLUNTO        3
0753     BKT-SIMPANGEMPA   3
0752     BATUSANGKAR       3
0751     BANDARBUAT        3
0748     SUNGAIPENUH       3
0747     MUARABUNGO        3
0746     BANGKO            3
0745     SAROLANGUNJAMBI   3
0744     MUARATEBO         3
0743     MUARABULIAN       3
0742     KUALATUNGKAL      3
0741     JAMBI             3
0739     BINTUHAN          3
0738     MUARAAMAN         3
0737521  ARGAMAKMUR        3
0737     IPUH              3
0736     BENGKULU          3
0735     BATURAJA          3
0734     MUARAENIM         3
0733     LUBUKLINGGAU      3
0732     CURUP             3
0731     LAHAT             3
0730     PAGARALAM         3
0729     PRINGSEWU         3
0728     KRUI              3
0727     KALIANDA          3
0726     MENGGALA          3
0725     BANDARJAYA        3
0724     BUKITKEMUNING     3
0722     KOTAAGUNG         3
0721     BANDARLAMPUNG     3
0719     MANGGAR           3
0718     KOBAPALEMBANG     3
0717     PANGKALPINANG     3
0716     MUNTOK            3
0714     SEKAYU            3
0713     PENDOPOTALANGUB   3
0712     KAYUAGUNG         3
0711     BETUNG            3
0702     TEBINGTINGGISUM   3
0659     BLANGPIDIE        3
0658     SINGKIL           3
0657     BAKONGAN          3
0656     TAPAKTUAN         3
0655     JEURAM            3
0654     CALANG            3
0653     BEUREUNUN         3
0652     SABANG            3
0651     BANDAACEH         3
0650     SINABANG          3
0646     IDI               3
0645     BLANGLANCANG      3
0644     BIREUEN           3
0643     TAKENGON          3
0642     BLANGKEJERAN      3
0641     KUALASIMPANG      3
0639     GUNUNGSITOLI      3
0636     KOTANOPAN         3
0635     GUNUNGTUA         3
0634     BATANGTORU        3
0633     DOLOKSANGGUL      3
0632     BALIGE            3
0631     SIBOLGA           3
0630     TELUKDALAM        3
0629     KUTACANE          3
0628     BANDARBARU S.     3
0627     SIDIKALANG        3
0626     PANGURURAN        3
0625     AMBARITA          3
0624     AEKKANOPAN        3
0623     KISARAN           3
0622     MEDANG•           3
0621     SUNGAIRAMPAH      3
0620     PANGKALANBRANDA   3
061896   TANJUNGPURA       3
```

061893	KUALA	3
061882	BINJAI	3
06184	MDN-CINTADAMAI	3
061836	MDN-TUNTUNGAN	3
06182	MDN-PADANGBULAN	3
061798	GALANG	3
061797	PANTAICERMIN	3
061795	LUBUKPAKAM	3
061794	TANJUNGMORAWA	3
06178	MDN-SIMPANGLIMU	3
06173	MDN-SUKARAMAI	3
061703	MDN-DELITUA	3
061	BELAWAN	3
0568	NANGAPINOH	3
0567	PUTUSSIBAU	3
0565	SINTANG	3
0564	BALAIKARANGAN	3
0563	NGABANG	3
056267	SUNGAIDURI,PTK	3
056263	SINGKAWANG	3
0562391	SAMBAS	3
0562371	TEBAS	3
0562	BENGKAYANG	3
056181	PTK-SIANTAN	3
056172	PTK-SEI RAYA	3
05617	PONTIANAK CENTR	3
056169	MEMPAWAH	3
056165	SUNGAIPINYUH	3
0556	NUNUKAN	3
0554	TANJUNGREDEB	3
0553	MALINAU	3
0552	TANJUNGSELOR	3
0551	TARAKAN	3
0549	SANGATTA	3
0548	BONTANG	3
0545	MELAK	3
0543	TANAHGROGOT	3
0542__4_	BALIKPAPAN CENT	3
05423	BALIKPAPAN CENT	3
0542	BPP-CENTRUM	3
05417	SAMARINDA	3
054166	TENGGARONG	3
0541	LOABAKUNG	3
0538	KUALAPEMBUANG	2
0537	KUALAKURUN	3
0536	KASUNGAN	3
0535	SUKADANAKAL	3
0534	KETAPANG	3
0533	NANGATAYAP	3
0532	KUMAI	3
0531	SAMPIT	3
0528	PURUKCAHU	3
0527	AMUNTAI	3
0526	TAMIANGLAYANG	3
0525	BUNTOK	3
0522	AMPAH	3
05192	MUARATEWEH	3
0518	KOTABARUPLAUT	3
0517	BARABAI	3
0516	TANJUNG	3
0515	BATULICIN	3
0514	PALANGKARAYA	3
0513	KUALAKAPUAS	3
0512	PLEIHARI	3
051179	MARABAHAN	3
051177	BANJARBARU	3
051172	MARTAPURA	3
051170	LANDASANULIN	3
05114	BANJARMASIN 1A	3
05113	BANJARMASIN 1B	3
0511	BJM-ULIN	3
0485	SENGKANG	3
0484	WATANSOPENG	3
0482	SINJAI	3
0481	WATAMPONE	3
0473	MASAMBA	3
0471	PALOPO	3
0465	KOLONEDALE	3
0464	AMPANA	3
0463	BUNTA	3
0462	BANGGAI	3
0461	LUWUK	3
0458	TENTENA	3
0457	DONGGALA	3
0453	TOLITOLI	3
0452	POSO	3
0451	PALU	3
0450	PARIGI	3
0443	MARISA	3
0438	BITUNG	3
0435	GORONTALO	3
0434	KOTAMOBAGU	3
0432	TAHUNA	3
0431	KLEAK/BAHU	3
0430	AMURANG	3
0428	POLEWALI	3
0427	BARRU	3
0426	MAMUJU	3
0423	MAKALE	3
0422	MAJENE	3
0421	PANGKAJENE SID	3
0420	ENREKANG	3
0419	JENEPONTO	3
0418	TAKALAR (TKL)	3
0417	MALINO	3
0414	SALAYAR	3
0413	BANTAENG	3
0411	ANTANG	3
0410	PANGKAJENEKEP	3
0408	UNAUNA	3
0405	KOLAKA	3
0404	WANCI	3
0403	RAHA	3
0402	BAUBAU	3
0401	KENDARI	3
0399	BAUCAU	3
0398	ERMERA	3
0396	LOSPALOS	3
0395	MANATUTO	3
0394	MALIANA	3
0390	AILIU	3
0389	ATAMBUA	3
0388	KEFAMENANU	3
0387	WAIKABUBAK	3
0386	KALABAHI	3
0385	LABUHANBAJO	3
0384	BAJAWA	3
0383	LARANTUKA	3
0382	MAUMERE	3
0381	ENDE	3
0380	BAA	3
0379	SUAI-TIMTIM	3
0378	PANTEMAKASAR	3
0377	VIQUEQUE	3
0376	SELONG	3
0374	BIMA	3
0373	DOMPU	3
0372	ALAS	3
0371	SUMBAWABESAR	3
0370	GERUNG	2
0368	BATURITI	2
0366	BANGLI	2

0335581	SUKAPURA	1
0335511	TONGAS	1
03354	PROBOLINGGO	1
033532	PROBOLINGGO	1
033488	LUMAJANG	1
033461	SENDURO	1
033459	PRONOJIWO	1
033457	PASIRIAN	1
033452	TEMPEH	1
033444	KLAKAH	1
033439	YOSOWILANGUN	1
033432	JATIROTO	1
0334	PASIRIAN	1
033397	KALIBARU	2
033381	GLENMORE	2
03338	GENTENG	2
033371	PESANGGARAN	2
033363	ROGOJAMPI	2
033361	WONGSOREJO	2
0333510	KETAPANG	2
03335	MUNCAR	2
03334	BANYUWANGI	2
033339	BENCULUK	2
03332	BANYUWANGI	2
03329	BESUKI	1
033256	PRAJEKAN	2
033241	BONDOWOSO	2
033232	SUKOSARI	2
033231	BONDOWOSO	2
033191	KALISAT	2
033175	JENGGAWAH	1
033171	RAMBIPUJI	1
033161	SEMPOLAN	2
03316	SUKOWONO	2
0331592	KALISAT	2
0331540	ARJASA	1
0331521	SEMPOLAN	2
03314	JEMBER_I	1
03312	JEMBER_I	1
032881	PERAGAAN	2
03286	SUMENEP	2
0328511	BATANGBATANG	2
0328311	AMBUNTEN	2
032781	GAYAM	2
032761	MASALEMBU	2
032751	SAPEKAN	2
032731	ARJASA	2
032541	TAMBAK	2
032521	SANGKAPURA	2
032451	WARU	1
03243	PAMEKASAN	1
03242	PAMEKASAN	1
03238	KETAPANG	1
032378	OMBEN	1
03232	SAMPANG	1
03226	BRONDONG	1
03224	BABAD	1
032239	SUKODADI	1
032231	LAMONGAN	1
03222	LAMONGAN	1
03218	JOMBANG	1
032171	NGOROJOMBANG	1
032169	PACET	1
032161	NGOROINDUSTRI	1
03216	PLOSO	1
032159	MOJOSARI	1
032151	DLANGGU	1
032149	MOJOAGUNG	1
032136	MLIRIP	1
03213	MOJOKERTO_B	1
03212	MOJOKERTO_B	1
031897	KRIAN	1
031896	GEDANGAN	1
03189	SIDOARJO	1
031885	TULANGAN	1
031883	SUKODONO	1
03187	JAGIR	1
03186	TROPODO	1
03185	WARU	1
03184	RUNGKUT	1
03182	INJOKO	1
031799	BALUNGPANGGANG	1
0317912	KEDAMEAN	1
03178	SEPANJANG	1
03176	KARANGPILANG	1
031759	BAMBE_B	1
03175	LAKARSANTRI	1
031749	KALIANAK_B	1
031748	KALIANAK_B	1
03174	KANDANGAN_B	1
0317	TANDES_A	1
03159	MANYAR	1
03150	GUBENG	1
0315	MERGOYOSO_D	1
031399	CERME_A	1
031398	GRESIK	1
031397	GRESIK	1
031395	PONGANGAN_B	1
031394	SEDAYU_B	1
031390	DUDUKSAMPEYAN	1
03138	KENJERAN_A	1
03137	KAPASAN_B	1
03132	PERAK	1
031309	BANGKALAN	1
0313086	TRAGAH	1
0313081	TANAHMERAH	1
0313080	GALIS	1
0313079	SEPULU	1
0313071	TANJUNGBUMI	1
031305	AROSBAYA	1
031304	BLEGA	1
0313036	KOKOP	1
0313031	GEGER	1
0313	KEBALEN	1
03129	ISDN / FAX	1
0312	KEBALEN_B	1
03113	PAGER	1
0311	INFO	1
0298	AMBARAWA	1
0297	KARIMUNJAWA	2
0296	BLORA	1
0295	JUWANA	1
0294	KENDAL	2
0293	MAGELANG	2
0292	GUBUG	1
0291	DEMAK	1
0289	BUMIAYU	2
0287	GOMBONG	2
0286	BANJARNEGARA	2
0285	BATANG	2
0284	PEMALANG	2
0283	ADIWERNA	2
0282	CILACAP-A.YANI	2
0281	BANYUMAS	2
0280	MAJENANG	1
0276	BOYOLALI	1
0275	KUTOARJO	2
0274	BANTUL	1
0273	BATURETNO	1
0272	DELANGGU	1
0271	KARANGANYAR SKA	1
0269	PAMEUNGPEUK	2

0365	GILIMANUK	2
0363	AMLAPURA	2
0362	LOVINA	2
0361	BENOA	2
035879	PRAMBON	3
03587	WARUJAYENG	3
03586	GONDANG	1
03585	KERTOSONO	3
0358325	NGANJUK	3
03578	PACITAN	1
0357481	WATUGEDE	1
0357421	SUDIMORO	1
035741	LOROG	1
035737	NAWANGAN	1
035735	JERUK	1
0357331	BANDAR	1
0357311	TEGALOMBO	1
03568	RENGEL	1
035671	MERAKURAK	1
035661	KEREK	1
035655	JATIROGO	1
0356491	TANJUNGAWARAWAR	1
035641	BANCAR	1
035632	TUBAN	1
03558	DURENAN	1
035579	TRENGGALEK	1
0355591	KALIDAWIR	1
0355551	PRIGI	1
035553	CAMPURDARAT	1
035539	NGUNUT	1
035532	TULUNGAGUNG	1
03538	BOJONEGORO	1
0353591	NGROHO	1
0353571	TAMBAKREJO	1
0353551	PURWOSARI	1
0353531	KASIMAN	1
0353511	KALITIDU	1
0353451	BUBULAN	1
0353431	NGAMBON	1
0353411	NGASEM	1
0353391	SUGIHWARAS	1
0353371	TEMAYANG	1
0353351	KEDUNGADEM	1
035333	SUMBERREJO	1
0353311	KEPUHBARU	1
03528	PONOROGO	1
0352791	SAMPUNG	1
035275	SUMOROTO	1
035261	PONOROGO	1
0352591	NGEBEL	1
035257	PULUNG	1
0352531	JENANGAN	1
035248	PONOROGO	1
0352391	NGRAYON	1
0352371	SLAHUNG	1
035231	SAMBIT	1
035189	MAGETAN	1
035188	SARANGAN	1
0351871	PARANG	1
035186	MAOSPATI	1
035173	JOGOROGO	1
03517	NGAWI	1
035166	KARANGJATI	1
0351656	BRINGIN	1
0351651	PANGKUR	1
0351611	KWARAKAN	1
03515	MADIUN_A	1
035146	MADIUN_A	1
035145	MADIUN_A	1
035143	GORANGGARENG	1
035138	CARUBAN	1
035136	UTERAN	1
0351331	SAWAHAN	1
0351321	KARE	1
0351311	GEMARANG	1
03512	MADIUN_A	1
0351	SARANGAN	1
034385	GEMPOL	1
03438	PRIGEN	1
03437	BANGIL	1
034365	BEJI	1
034363	PANDAAN	1
034361	PURWOSARI	1
0343571	TOSARI	1
034349	NONGKOJAJAR	1
034348	GRATI	1
0343441	GONDANGWETAN	1
034341	PASURUAN	1
034320	PASURUAN	1
03428	BLITAR	1
034269	WLINGI	1
0342562	PANATARAN	1
03425	SRENGAT	3
03424	LODOYO	1
0342351	BINANGUN	1
03423	KESAMBEN	1
0342	PANATARAN	1
034189	DAMPIT	1
0341881	DONOMULYO	1
0341871	SUMBERMANJING	1
034187	GONDANGLEGI	1
0341851	AMPELGADING	1
0341841	BANTUR	1
0341833	BULULAWANG	1
034182	TUREN_A	1
034180	GADANG	1
034179	PAKIS	1
034178	TUMPANG	1
034175	BURING	1
034171	SAWOJAJAR	1
034159	BATU	1
0341521	NGANTANG	1
034152	PUJON	1
03415	KLOJEN	1
034146	KARANGPLOSO	1
034145	SINGOSARI	1
034142	LAWANG	1
03414	BLIMBING	1
034139	KEPANJEN	1
034138	SUMBERPUCUNG	1
034137	GUNUNGKAWI	1
034136	MALANG	1
034135	MALANG	1
034134	MALANG	1
034132	MALANG	1
0341311	PAGAK	1
03412	MALANG	1
0341123	MALANG Info PLN	1
03386	SITUBONDO	2
033845	ASEMBAGUS	2
0338390	MLANDINGAN	1
033828	ASEMBAGUS	2
033688	AMBULU	1
033672	PUGER	1
03366	BALUNG	1
033644	TANGGUL	1
03363	KENCONG	1
0336	PUGER	1
03358	KRAKSAAN	1
033577	PAITON	1
033568	LECES	1
033561	GENDING	1

0268	PALABUHANRATU	3
0267	KARANGLIGAR	3
0266	BOJONGLOPANG	3
0265	BANJARSARI	2
0264	BOJONG	3
0263	CIANJUR	3
0262	CIBATU	2
0261	SUMEDANG	2
0255	SINDANGLAYA	3
0254	BAROS	3
025380	LABUAN	3
025350	MENES	3
025340	SAKETI	3
025320	PANDEGLANG	3
025250	MALINGPING	3
025240	BAYAH	3
025220	RANGKASBITUNG	3
0251	CIAPUS	3
024	SEMARANG	1
023474	HAURGEULIS	2
023450	LOSARANG	2
023442	BALONGAN	2
023435	JATIBARANG	2
023427	INDRAMAYU	2
0234	ARJAWINANGUN	2
023388	JATIWANGI	2
023366	KADIPATEN	2
023331	CIKIJING	2
02332	MAJALENGKA	2
0233	RAJAGALUH	2
023261	CILIMUS	2
0232	KUNINGAN	2
0231	CBN-KANCI	2
02277	RANCAEKEK	2
02273	BD-TURANGGA	2
022203	BD-HEGARMANAH	3
022201	BD-GEGERKALONG	3
022	BANDUNG	3
02189	BKS-CIKARANG	3
02188	BEKASI	3
0218776	GANDARIA	3
021877	CISALAK	3
021873	CIBUBUR	3
02187	GANDARIA	3
021866	KLENDER	3
021864	PONDOK KELAPA	3
021840	PASAR REBO	3
02184	KRANGGAN	3
02183	TEBET	3
02180	CAWANG	3
02177	DEPOK	3
021759	CIPETE-2	3
021755	CINERE	3
021748	PONDOK AREN-2	3
021736	BINTARO	3
02172	TANAH KUSIR	3
02171	KEMANG	3
02168	ANCOL	3
02165	KEMAYORAN	3
02164	ANCOL	3
02163	CIDENG	3
02160	MANGGADUA	3
021588	KAPUK	3
021584	MERUYA	3
021583	KEDOYA	3
02157	SEMANGGI-1	3
02156	SLIPI	3
021559	CENGKARENG	3
021555	TEGAL ALUR	3
021550	CENGKARENG	3
02154	CENGKARENG	3
02153	PALMERAH	3
02148	PULO GEBANG	3
02147	RAWAMANGUN	3
021468	PENGGILINGAN	3
02145	KELAPA GADING	3
02144	CILINCING	3
02142	CEMPAKA PUTIH	3
02138	KPUS-GAMBIR	3
02131	CIKINI	3
0212754	PASARMINGGU-2	3
0212753	KALIBATA-2	3
021250	SEMANGGI-2	3
021	_JAKARTA	3
end		

```
Name     : SLI.lst
Function : Identifier
Location : Surabaya
Made by  : Valens Sigit Hascaryo
Date     : XX-XX-01
Max Area : 300 kode area
begin
;Zona-I
880      Bangladesh       1
975      Bhutan           1
673      Brunei           1
95       Burma            1
855      Cambodia         1
856      Laos             1
853      Macao            1
976      Mongolia         1
84       Vietnam          1
;Zona-II
93       Afghanistan      2
213      Algeria          2
244      Angola           2
1264     Anguilla         2
1268     Antigua-Barbuda  2
599      Antilles         2
54       Argentina        2
20       Egypt            2
7        Tatarstan        2
;Zona-III
357      Cyprus           3
1671     Guam             3
972      Israel           3
962      Jordan           3
961      Lebanon          3
;Zona-IV
1907     Alaska           4
355      Albania          4
376      Andora           4
6721     Antartic Base    4
351      Azores & Madeira 4
;Zona-V
61       Australia        5
1        USA              5
1204     Canada           5
1306     Canada           5
1403     Canada           5
1416     Canada           5
1418     Canada           5
1506     Canada           5
1514     Canada           5
1519     Canada           5
1604     Canada           5
1613     Canada           5
1705     Canada           5
1709     Canada           5
1819     Canada           5
1902     Canada           5
1808     Hawaii           5
852      Hongkong         5
82       South Korea      5
64       New Zealand      5
886      Taiwan           5
;Zona-VI
971      Arab Emirat      6
91       India            6
81       Japan            6
966      Saudi Arabia     6
44       United Kingdom   6
;Zona-VII
43       Austria          7
32       Belgium          7
86       China            7
45       Denmark          7
33       France           7
49       Germany          7
39       Italy            7
31       Netherlands      7
47       Norway           7
34       Spain            7
46       Sweden           7
41       Switzerland      7
;Zona-VIII sampai XI (INMARSAT)
8711     Inmarsat A       11
8713     Inmarsat B       10
8715     Inmarsat Aero    11
8716     Inmarsat M       10
8717     Inmarsat Mini-M  9
8718     Inmarsat A       11
8721     Inmarsat A       11
8723     Inmarsat B       10
8725     Inmarsat Aero    11
8726     Inmarsat M       10
8727     Inmarsat Mini-M  9
8731     Inmarsat A       11
8733     Inmarsat B       8
8735     Inmarsat Aero    11
8736     Inmarsat M       8
8737     Inmarsat Mini-M  8
8738     Inmarsat A       11
8741     Inmarsat A       11
8743     Inmarsat B       10
8745     Inmarsat Aero    11
8746     Inmarsat M       10
8747     Inmarsat Mini-M  9
8748     Inmarsat A       11
end
```

```
Name     : DBase\STB.lst
Function : STB Identifier
Location : Surabaya
Made by  : Valens Sigit Hascaryo
Date     : XX-XX-01
Max Area : 250 Kode Area
begin
081      GSM-900          1
0816542  Mentari-Sby      1
0811321  Halo-Sby         1
082      AMPS             2
083      GSM-1800         3
0815     IM3              3
end
```

ions Sigit Nascaryo - 5103097064

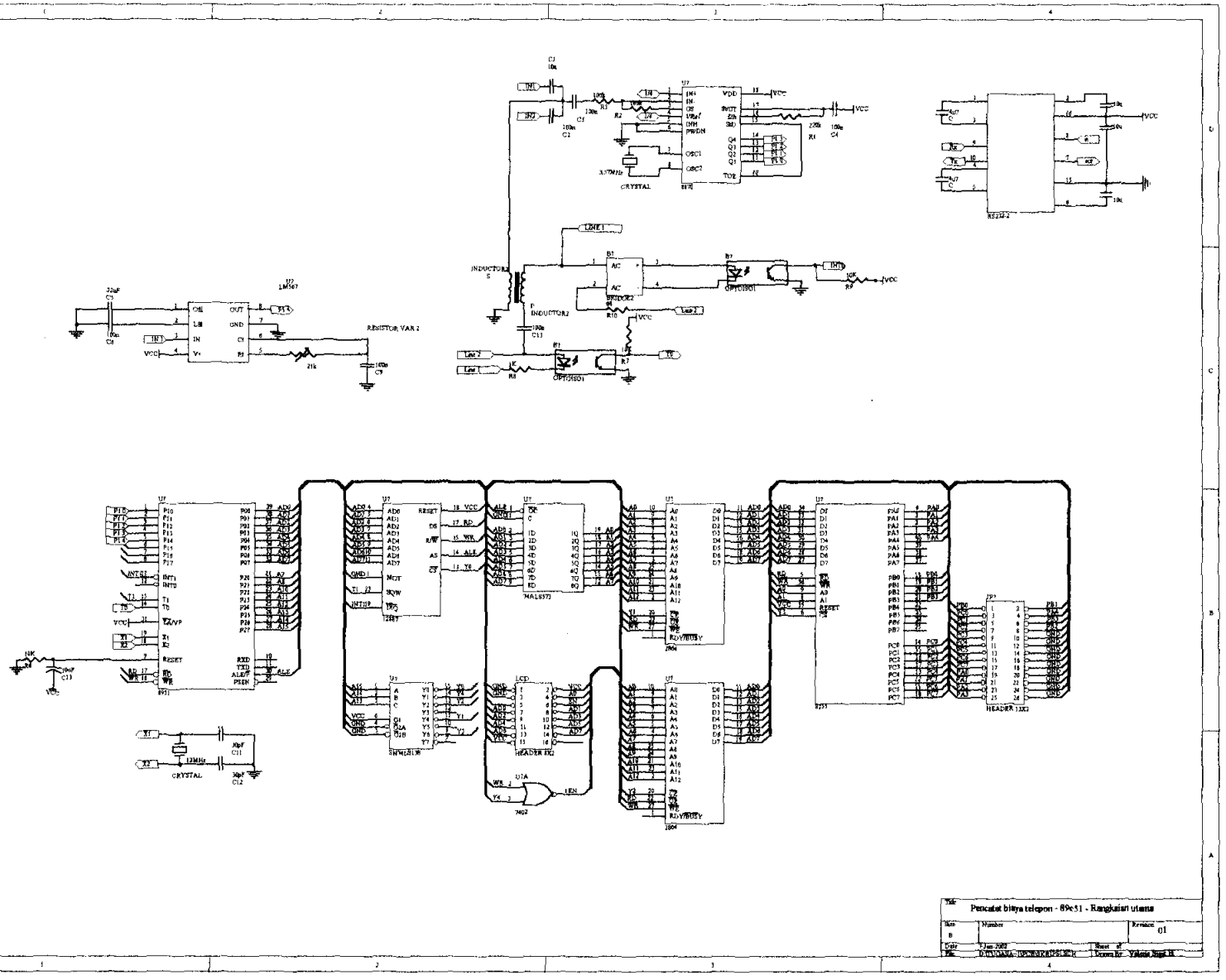
anggal : 04:01:'02
ari : Jumat
am : 07:05:36
ujuan : 3621179
umlah Pulsa : 015
arga : Rp. 2505

ions Sigit Nascaryo - 5103097064

anggal : 04:01:'02
ari : Jumat
am : 08:01:41
ujuan : 081550333881
umlah Pulsa : 006
arga : Rp. 246

ions Sigit Nascaryo - 5103097064

anggal : 04:01:'02
ari : Jumat
am : 08:24:15
ujuan : 0274562237
umlah Pulsa : 009
arga : Rp. 1206



Valeria Sigit H

atures

ompatible with MCS-51™ Products
K Bytes of In-System Reprogrammable Flash Memory
– Endurance: 1,000 Write/Erase Cycles
ully Static Operation: 0 Hz to 24 MHz
hree-Level Program Memory Lock
28 x 8-Bit Internal RAM
2 Programmable I/O Lines
wo 16-Bit Timer/Counters
ix Interrupt Sources
rogrammable Serial Channel
ow Power Idle and Power Down Modes

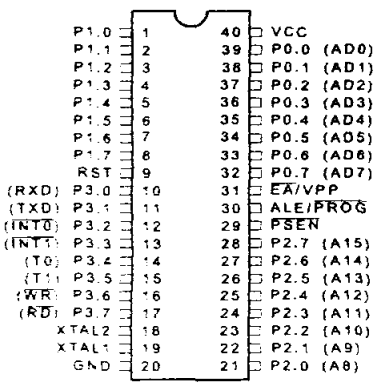
escription

e AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K es of Flash Programmable and Erasable Read Only Memory (PEROM). The vice is manufactured using Atmel's high density nonvolatile memory technology l is compatible with the industry standard MCS-51™ instruction set and pinout. The chip Flash allows the program memory to be reprogrammed in-system or by a con- itional nonvolatile memory programmer. By combining a versatile 8-bit CPU with sh on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which vides a highly flexible and cost effective solution to many embedded control appli- ions.

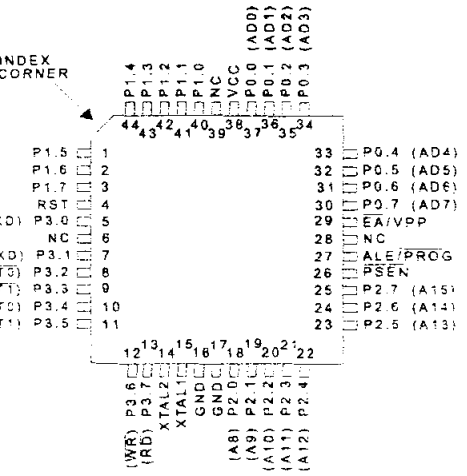
(continued)

n Configurations

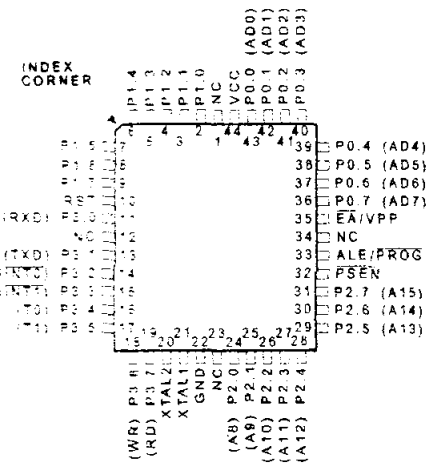
PDIP



PQFP/TQFP



PLCC

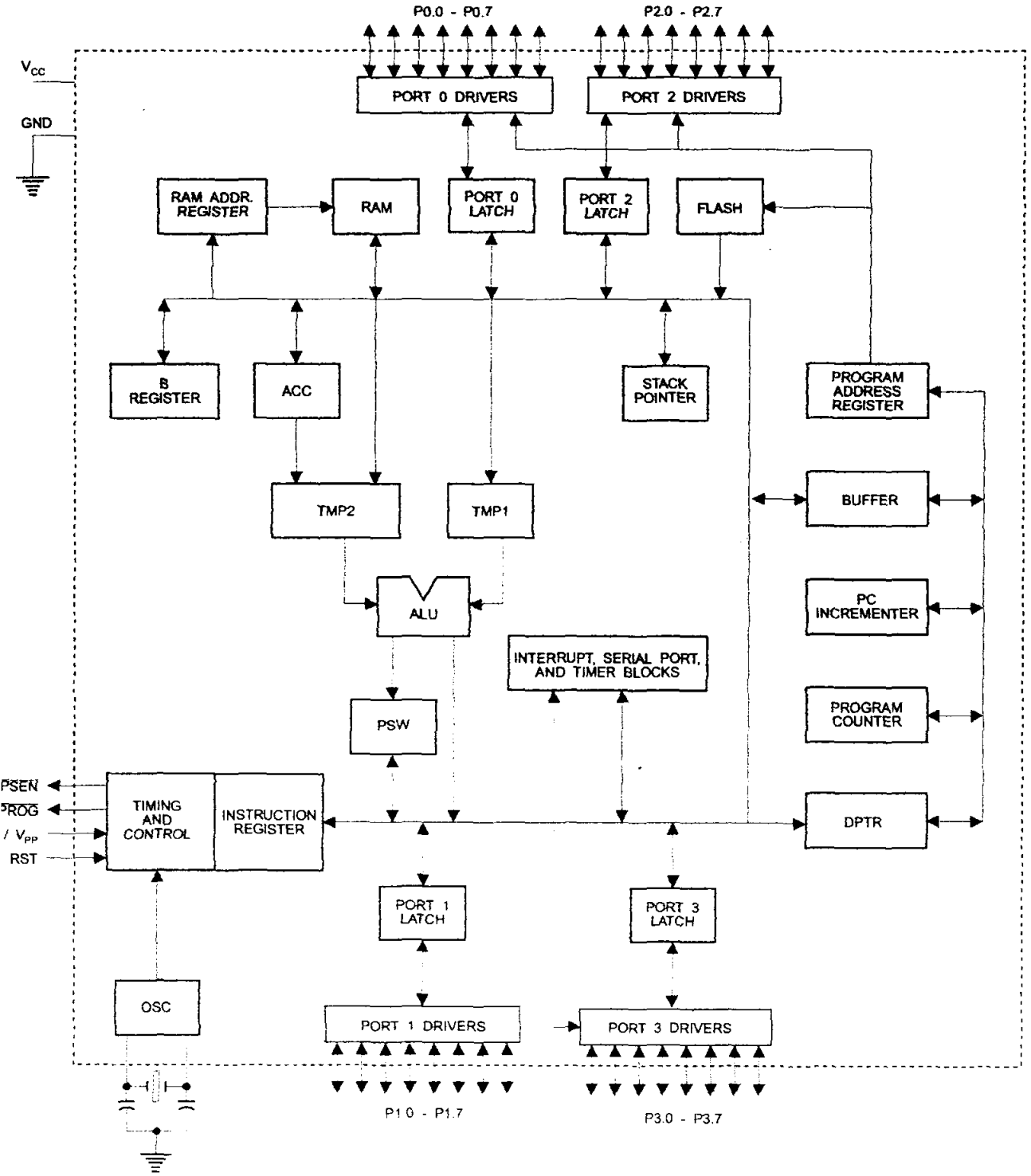


8-Bit Microcontroller with 4K Bytes Flash

AT89C51

0265F-A–12/97

k Diagram


P0.0 - P0.7
P2.0 - P2.7
V_{CC}
GND
PORT 0 DRIVERS
PORT 2 DRIVERS
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
FLASH
B REGISTER
ACC
STACK POINTER
PROGRAM ADDRESS REGISTER
TMP2
TMP1
BUFFER
ALU
PC INCREMENTER
INTERRUPT, SERIAL PORT, AND TIMER BLOCKS
PSW
PROGRAM COUNTER
PSEN
PROG
/ V_{PP}
RST
TIMING AND CONTROL
INSTRUCTION REGISTER
DPTR
PORT 1 LATCH
PORT 3 LATCH
OSC
PORT 1 DRIVERS
PORT 3 DRIVERS
P1.0 - P1.7
P3.0 - P3.7

3. Programming the Flash

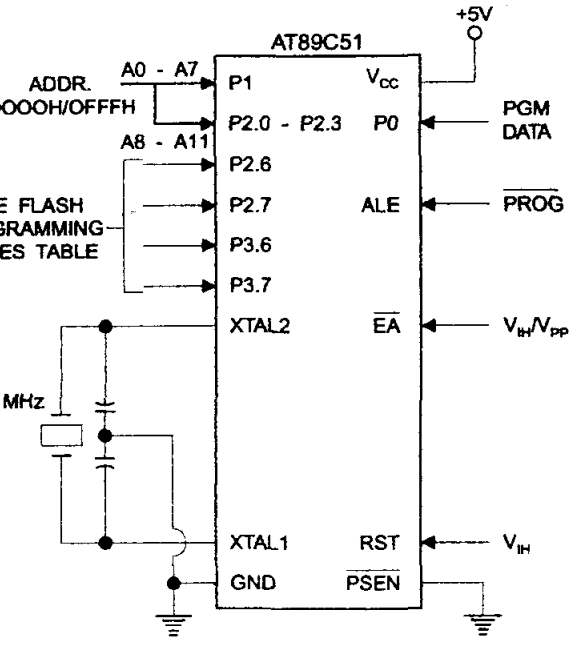
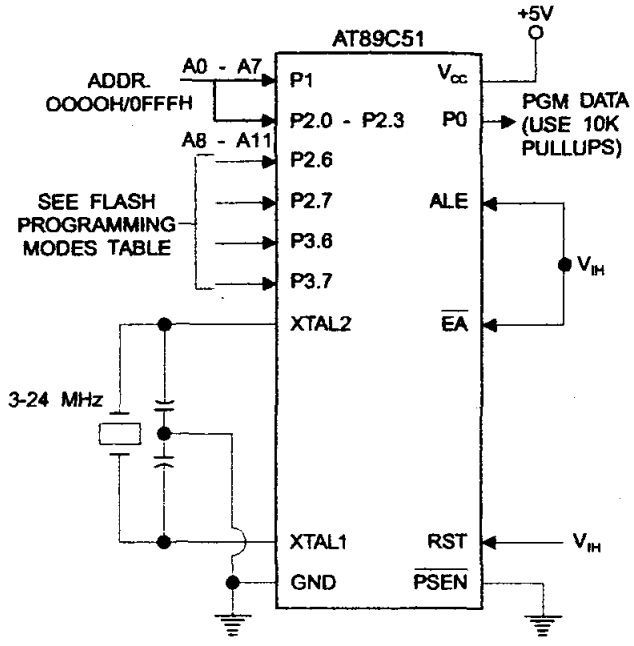


Figure 4. Verifying the Flash



h Programming and Verification Characteristics

°C to 70°C, V_{CC} = 5.0 ± 10%

ol	Parameter	Min	Max	Units
)	Programming Enable Voltage	11.5	12.5	V
	Programming Enable Current		1.0	mA
L	Oscillator Frequency	3	24	MHz
	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
(1)	V_{PP} Hold After $\overline{PROG}$	10		µs
	$\overline{PROG}$ Width	1	110	µs
	Address to Data Valid		$48t_{CLCL}$	
	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
	Byte Write Cycle Time		2.0	ms

1. Only used in 12-volt programming mode.

atures

- Fast Read Access Time - 120 ns
- Fast Byte Write - 200 µs or 1 ms
- Self-Timed Byte Write Cycle
  - Internal Address and Data Latches
  - Internal Control Timer
  - Automatic Clear Before Write
- Direct Microprocessor Control
  - READY/BUSY Open Drain Output
  - DATA Polling
- Low Power
  - 30 mA Active Current
  - 100 µA CMOS Standby Current
- High Reliability
  - Endurance: 10^4 or 10^5 Cycles
  - Data Retention: 10 Years
- 5V ± 10% Supply
- CMOS and TTL Compatible Inputs and Outputs
- JEDEC Approved Byte-Wide Pinout
- Commercial and Industrial Temperature Ranges

64K (8K x 8) CMOS E^2PROM

escription

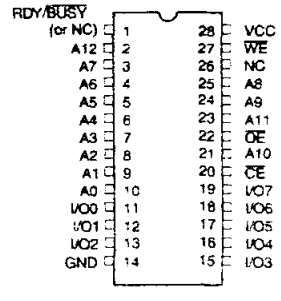
e AT28C64 is a low-power, high-performance 8,192 words by 8 bit nonvolatile ctrically Erasable and Programmable Read Only Memory with popular, easy to e features. The device is manufactured with Atmel's reliable nonvolatile technol-y.

(continued)

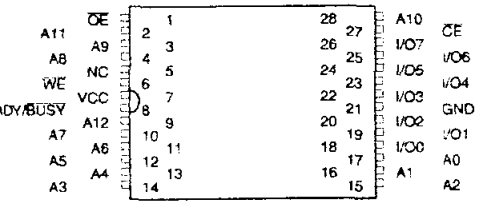
n Configurations

Pin Name	Function
A0 - A12	Addresses
CE	Chip Enable
OE	Output Enable
WE	Write Enable
I/O0 - I/O7	Data Inputs/Outputs
RDY/BUSY	Ready/Busy Output
NC	No Connect
DC	Don't Connect

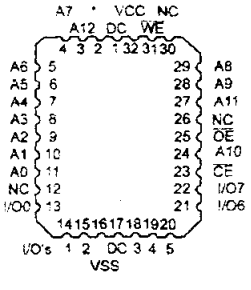
PDIP, SOIC
Top View



TSOP
Top View



LCC, PLCC
Top View



* = RDY/BUSY (or NC)

Note: PLCC package pins 1 and 17 are DON'T CONNECT.

0001G

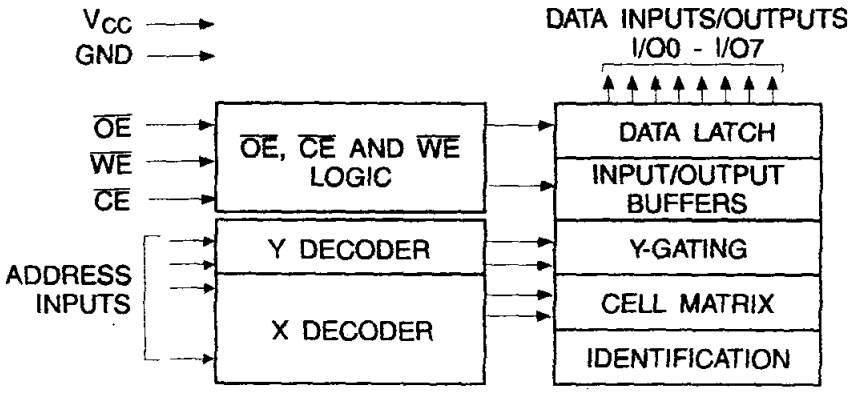
escription (Continued)

ne AT28C64 is accessed like a Static RAM for the read write cycles without the need for external components. uring a byte write, the address and data are latched in- rnally, freeing the microprocessor address and data bus r other operations. Following the initiation of a write cy- e, the device will go to a busy state and automatically ear and write the latched data using an internal control mer. The device includes two methods for detecting the nd of a write cycle, level detection of RDY/BUSY (unless in 1 is N.C.) and DATA POLLING of I/O_7. Once the end f a write cycle has been detected, a new access for a ead or write can begin.

The CMOS technology offers fast access times of 120 ns at low power dissipation. When the chip is deselected the standby current is less than 100 µA.

Atmel's 28C64 has additional features to ensure high quality and manufacturability. The device utilizes error correction internally for extended endurance and for improved data retention characteristics. An extra 32-bytes of E^2PROM are available for device identification or tracking.

Block Diagram



Absolute Maximum Ratings*

Temperature Under Bias	-55°C to +125°C
Storage Temperature	-65°C to +150°C
All Input Voltages (including NC Pins) with Respect to Ground	-0.6V to +6.25V
All Output Voltages with Respect to Ground	-0.6V to V_{CC} + 0.6V
Voltage on $\overline{OE}$ and A9 with Respect to Ground	-0.6V to +13.5V

*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.

C and AC Operating Range

		AT28C64-12	AT28C64-15	AT28C64-20	AT28C64-25
Operating Temperature (Case)	Com.	0°C - 70°C	0°C - 70°C	0°C - 70°C	0°C - 70°C
	Ind.	-40°C - 85°C	-40°C - 85°C	-40°C - 85°C	-40°C - 85°C
V_{CC} Power Supply		5V ± 10%	5V ± 10%	5V ± 10%	5V ± 10%

Operating Modes

Mode	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	I/O
Read	V_{IL}	V_{IL}	V_{IH}	D_{OUT}
Write (2)	V_{IL}	V_{IH}	V_{IL}	D_{IN}
Standby/Write Inhibit	V_{IH}	X (1)	X	High Z
Write Inhibit	X	X	V_{IH}	
Write Inhibit	X	V_{IL}	X	
Output Disable	X	V_{IH}	X	High Z
Chip Erase	V_{IL}	V_H (3)	V_{IL}	High Z

Notes: 1. X can be V_{IL} or V_{IH}.
2. Refer to AC Programming Waveforms.
3. V_H = 12.0V ± 0.5V.

DC Characteristics

Symbol	Parameter	Condition		Min	Max	Units
I_{LI}	Input Load Current	V_{IN} = 0V to V_{CC} + 1V			10	µA
I_{LO}	Output Leakage Current	$V_{I/O}$ = 0V to V_{CC}			10	µA
I_{SB1}	V_{CC} Standby Current CMOS	$\overline{CE}$ = V_{CC} - 0.3V to V_{CC} + 1.0V			100	µA
I_{SB2}	V_{CC} Standby Current TTL	$\overline{CE}$ = 2.0V to V_{CC} + 1.0V	Com.		2	mA
			Ind.		3	mA
I_{CC}	V_{CC} Active Current AC	f = 5 MHz; I_{OUT} = 0 mA $\overline{CE}$ = V_{IL}	Com.		30	mA
			Ind.		45	mA
V_{IL}	Input Low Voltage				0.8	V
V_{IH}	Input High Voltage			2.0		V
V_{OL}	Output Low Voltage	I_{OL} = 2.1 mA = 4.0 mA for RDY/$\overline{BUSY}$			.45	V
V_{OH}	Output High Voltage	I_{OH} = -400 µA		2.4		V

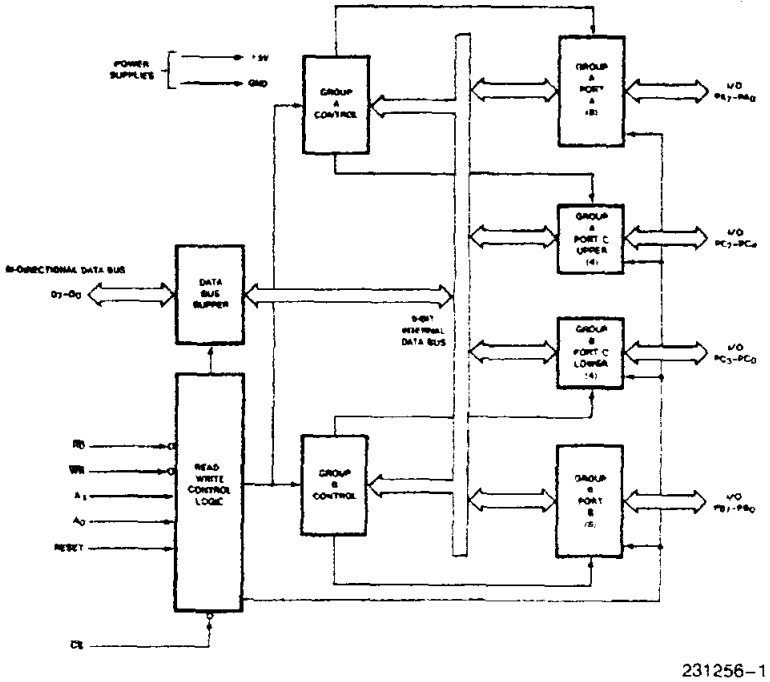
82C55A
CHMOS PROGRAMMABLE PERIPHERAL INTERFACE

- **Compatible with all Intel and Most Other Microprocessors**
- **High Speed, "Zero Wait State" Operation with 8 MHz 8086/88 and 80186/188**
- **24 Programmable I/O Pins**
- **Low Power CHMOS**
- **Completely TTL Compatible**
- **Control Word Read-Back Capability**
- **Direct Bit Set/Reset Capability**
- **2.5 mA DC Drive Capability on all I/O Port Outputs**
- **Available in 40-Pin DIP and 44-Pin PLCC**
- **Available in EXPRESS**
  - **Standard Temperature Range**
  - **Extended Temperature Range**

The Intel 82C55A is a high-performance, CHMOS version of the industry standard 8255A general purpose programmable I/O device which is designed for use with all Intel and most other microprocessors. It provides 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The 82C55A is pin compatible with the NMOS 8255A and 8255A-5.

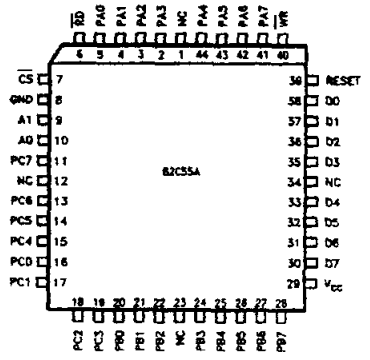
In MODE 0, each group of 12 I/O pins may be programmed in sets of 4 and 8 to be inputs or outputs. In MODE 1, each group may be programmed to have 8 lines of input or output. 3 of the remaining 4 pins are used for handshaking and interrupt control signals. MODE 2 is a strobed bi-directional bus configuration.

The 82C55A is fabricated on Intel's advanced CHMOS III technology which provides low power consumption with performance equal to or greater than the equivalent NMOS product. The 82C55A is available in 40-pin DIP and 44-pin plastic leaded chip carrier (PLCC) packages.

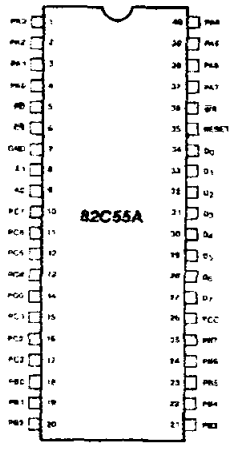


231256-1

Figure 1. 82C55A Block Diagram



231256-31



231256-2

Figure 2. 82C55A Pinout

Diagrams are for pin reference only. Package sizes are not to scale.

Table 1. Pin Description

Symbol	Pin Number Dip	Pin Number PLCC	Type	Name and Function
PA_{3-0}	1–4	2–5	I/O	**PORT A, PINS 0–3:** Lower nibble of an 8-bit data output latch/ buffer and an 8-bit data input latch.
$\overline{RD}$	5	6	I	**READ CONTROL:** *This input is low during CPU read operations.*
$\overline{CS}$	6	7	I	**CHIP SELECT:** A low on this input enables the 82C55A to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and WR are ignored otherwise.
GND	7	8		**System Ground**
A_{1-0}	8–9	9–10	I	**ADDRESS:** These input signals, in conjunction $\overline{RD}$ and $\overline{WR}$, control the selection of one of the three ports or the control word registers.

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input Operation (Read)
0	0	0	1	0	Port A - Data Bus
0	1	0	1	0	Port B - Data Bus
1	0	0	1	0	Port C - Data Bus
1	1	0	1	0	Control Word - Data Bus
					Output Operation (Write)
0	0	1	0	0	Data Bus - Port A
0	1	1	0	0	Data Bus - Port B
1	0	1	0	0	Data Bus - Port C
1	1	1	0	0	Data Bus - Control
					Disable Function
X	X	X	X	1	Data Bus - 3 - State
X	X	1	1	0	Data Bus - 3 - State

PC_{7-4}	10–13	11,13–15	I/O	**PORT C, PINS 4–7:** Upper nibble of an 8-bit data output latch/ buffer and an 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.
PC_{0-3}	14–17	16–19	I/O	**PORT C, PINS 0–3:** Lower nibble of Port C.
PB_{0-7}	18–25	20–22, 24–28	I/O	**PORT B, PINS 0–7:** An 8-bit data output latch/buffer and an 8-bit data input buffer.
V_{CC}	26	29		**SYSTEM POWER:** + 5V Power Supply.
D_{7-0}	27–34	30–33, 35–38	I/O	**DATA BUS:** Bi-directional, tri-state data bus lines, connected to system data bus.
RESET	35	39	I	**RESET:** A high on this input clears the control register and all ports are set to the input mode.
$\overline{WR}$	36	40	I	**WRITE CONTROL:** This input is low during CPU write operations.
PA_{7-4}	37–40	41–44	I/O	**PORT A, PINS 4–7:** Upper nibble of an 8-bit data output latch/ buffer and an 8-bit data input latch.
NC		1, 12, 23, 34		*No Connect*

82C55A OPERATIONAL DESCRIPTION

Mode Selection

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

Mode 0 — Basic input/output
Mode 1 — Strobed Input/output
Mode 2 — Bi-directional Bus

When the reset input goes "high" all ports will be set to the input mode with all 24 port lines held at a logic "one" level by the internal bus hold devices (see Figure 4 Note). After the reset is removed the 82C55A can remain in the input mode with no additional initialization required. This eliminates the need for pullup or pulldown devices in "all CMOS" designs. During the execution of the system program, any of the other modes may be selected by using a single output instruction. This allows a single 82C55A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

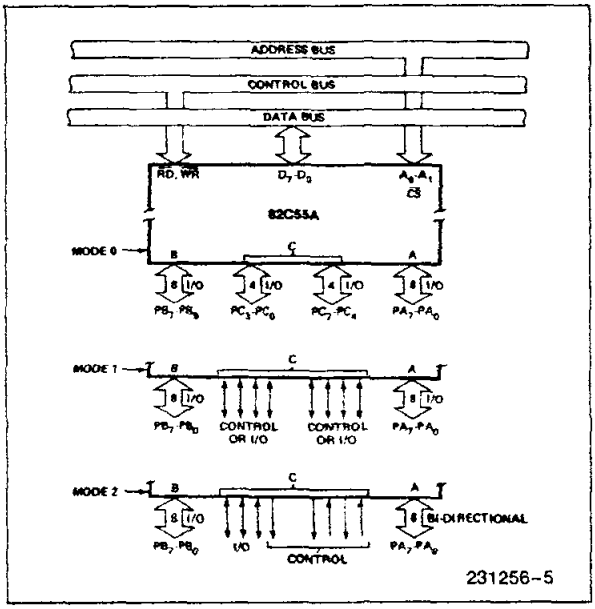


Figure 5. Basic Mode Definitions and Bus Interface

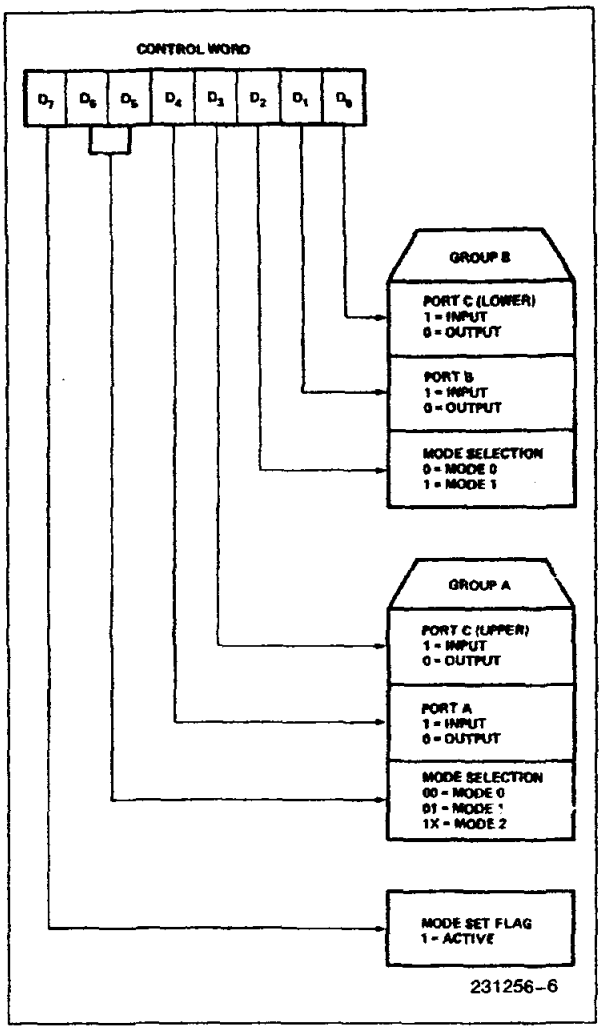


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 82C55A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

DALLAS SEMICONDUCTOR

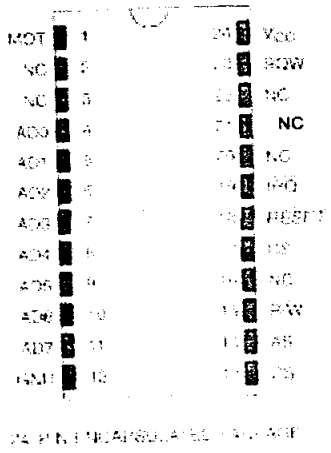
DS12887
Real Time Clock

www.dalsemi.com

FEATURES

- Drop–in replacement for IBM AT computer clock/calendar
- Pin-compatible with the MC146818B and DS1287
- Totally nonvolatile with over 10 years of operation in the absence of power
- Self–contained subsystem includes lithium, quartz, and support circuitry
- Counts seconds, minutes, hours, days, day of the week, date, month, and year with leap year compensation valid up to 2100
- Binary or BCD representation of time, calendar, and alarm
- 12– or 24–hour clock with AM and PM in 12–hour mode
- Daylight Savings Time option
- Selectable between Motorola and Intel bus timing
- Multiplex bus for pin efficiency
- Interfaced with software as 128 RAM locations
  - 14 bytes of clock and control registers
  - 114 bytes of general purpose RAM
- Programmable square wave output signal
- Bus–compatible interrupt signals ($\overline{IRQ}$)
- Three interrupts are separately software–maskable and testable
  - Time–of–day alarm once/second to once/day
  - Periodic rates from 122 ms to 500 ms
  - End of clock update cycle

PIN ASSIGNMENT



24 PIN ENCAPSULATED PACKAGE

PIN DESCRIPTION

AD0–AD7	– Multiplexed Address/Data Bus
NC	– No Connection
MOT	– Bus Type Selection
$\overline{CS}$	– Chip Select
AS	– Address Strobe
R/$\overline{W}$	– Read/Write Input
DS	– Data Strobe
$\overline{RESET}$	– Reset Input
$\overline{IRQ}$	– Interrupt Request Output
SQW	– Square Wave Output
V_{CC}	– +5 Volt Supply
GND	– Ground

DESCRIPTION

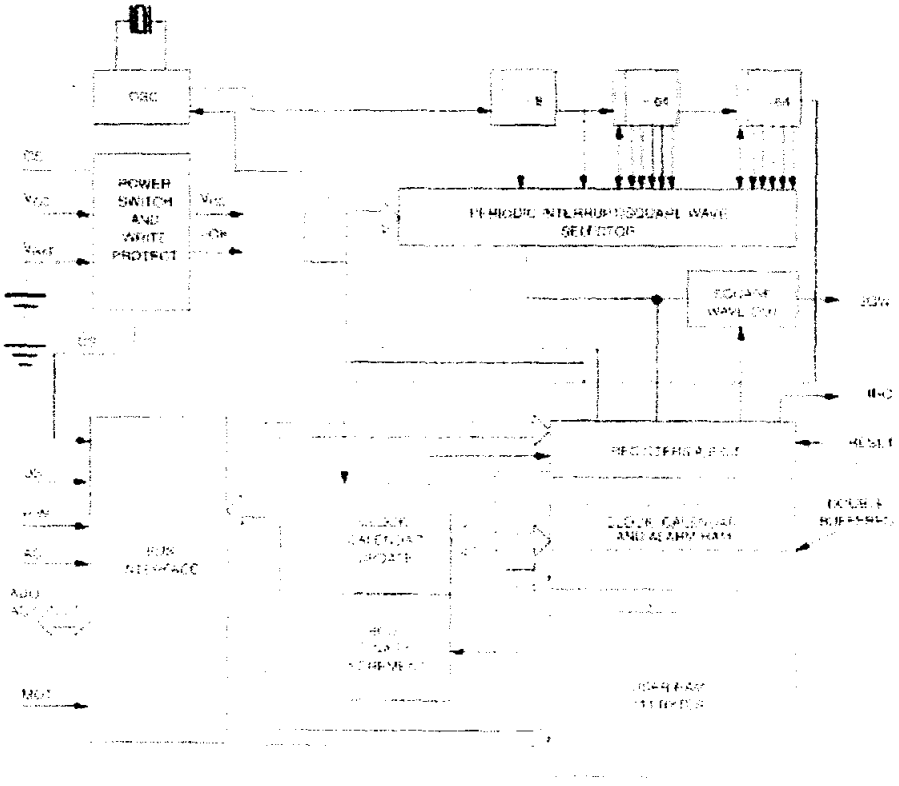
The DS12887 Real Time Clock plus RAM is designed to be a direct replacement for the DS1287. The DS12887 is identical in form, fit, and function to the DS1287, and has an additional 64 bytes of general purpose RAM. Access to this additional RAM space is determined by the logic level presented on AD6 during the address portion of an access cycle. A lithium energy source, quartz crystal, and write–protection circuitry are contained within a 24–pin dual in-line package. As such, the DS12887 is a complete subsystem replacing 16 components in a typical application. The functions include a nonvolatile time–of–day clock, an alarm, a one-hundred–year calendar, programmable interrupt, square wave

generator, and 114 bytes of nonvolatile static RAM. The real time clock is distinctive in that time–of–day and memory are maintained even in the absence of power.

OPERATION

The block diagram in Figure 1 shows the pin connections with the major internal functions of the DS12887. The following paragraphs describe the function of each pin.

BLOCK DIAGRAM DS12887 Figure 1



POWER–DOWN/POWER–UP CONSIDERATIONS

The Real Time Clock function will continue to operate and all of the RAM, time, calendar, and alarm memory locations remain nonvolatile regardless of the level of the V_{CC} input. When V_{CC} is applied to the DS12887 and reaches a level of greater than 4.25 volts, the device becomes accessible after 200 ms, provided that the oscillator is running and the oscillator countdown chain is not in reset (see Register A). This time period allows the system to stabilize after power is applied. When V_{CC} falls below 4.25 volts, the chip select input is internally forced to an inactive level regardless of the value of $\overline{CS}$ at the input pin. The DS12887 is, therefore, write–protected. When the DS12887 is in a write–protected state, all inputs are ignored and all outputs are in a high impedance state. When V_{CC} falls below a level of approximately 3 volts, the external V_{CC} supply is switched off and an internal lithium energy source supplies power to the Real Time Clock and the RAM memory.

SIGNAL DESCRIPTIONS

GND, V_{CC} – DC power is provided to the device on these pins. V_{CC} is the +5 volt input. When 5 volts are applied within normal limits, the device is fully accessible and data can be written and read. When V_{CC} is below 4.25 volts typical, reads and writes are inhibited. However, the timekeeping function continues

unaffected by the lower input voltage. As V_{CC} falls below 3 volts typical, the RAM and timekeeper are switched over to an internal lithium energy source. The timekeeping function maintains an accuracy of ±1 minute per month at 25°C regardless of the voltage input on the V_{CC} pin.

MOT (Mode Select) – The MOT pin offers the flexibility to choose between two bus types. When connected to V_{CC}, Motorola bus timing is selected. When connected to GND or left disconnected, Intel bus timing is selected. The pin has an internal pulldown resistance of approximately 20 kΩ.

SQW (Square Wave Output) – The SQW pin can output a signal from one of 13 taps provided by the 15 internal divider stages of the Real Time Clock. The frequency of the SQW pin can be changed by programming Register A as shown in Table 1. The SQW signal can be turned on and off using the SQWE bit in Register B. The SQW signal is not available when V_{CC} is less than 4.25 volts, typically.

PERIODIC INTERRUPT RATE AND SQUARE WAVE OUTPUT FREQUENCY Table 1

SELECT BITS REGISTER A				t_{PI} PERIODIC INTERRUPT RATE	SQW OUTPUT FREQUENCY
RS3	RS2	RS1	RS0		
0	0	0	0	None	None
0	0	0	1	3.90625 ms	256 Hz
0	0	1	0	7.8125 ms	128 Hz
0	0	1	1	122.070 µs	8.192 kHz
0	1	0	0	244.141 µs	4.096 kHz
0	1	0	1	488.281 µs	2.048 kHz
0	1	1	0	976.5625 µs	1.024 kHz
0	1	1	1	1.953125 ms	512 Hz
1	0	0	0	3.90625 ms	256 Hz
1	0	0	1	7.8125 ms	128 Hz
1	0	1	0	15.625 ms	64 Hz
1	0	1	1	31.25 ms	32 Hz
1	1	0	0	62.5 ms	16 Hz
1	1	0	1	125 ms	8 Hz
1	1	1	0	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz

AD0–AD7 (Multiplexed Bidirectional Address/Data Bus) – Multiplexed buses save pins because address information and data information time-share the same signal paths. The addresses are present during the first portion of the bus cycle and the same pins and signal paths are used for data in the second portion of the cycle. Address/data multiplexing does not slow the access time of the DS12887 since the bus change from address to data occurs during the internal RAM access time. Addresses must be valid prior to the falling edge of AS/ ALE, at which time the DS12887 latches the address from AD0 to AD6. Valid write data must be present and held stable during the latter portion of the DS or $\overline{WR}$ pulses. In a read cycle the DS12887 outputs 8 bits of data during the latter portion of the DS or $\overline{RD}$ pulses. The read cycle is terminated and the bus returns to a high impedance state as DS transitions low in the case of Motorola timing or as $\overline{RD}$ transitions high in the case of Intel timing.

AS (Address Strobe Input) – A positive-going address strobe pulse serves to demultiplex the bus. The falling edge of AS/ALE causes the address to be latched within the DS12887. The next rising edge that

ccurs on the AS bus will clear the address regardless of whether $\overline{CS}$ is asserted. Access commands ıould be sent in pairs.

S (Data Strobe or Read Input) – The DS/$\overline{RD}$ pin has two modes of operation depending on the level f the MOT pin. When the MOT pin is connected to V_{CC}, Motorola bus timing is selected. In this mode S is a positive pulse during the latter portion of the bus cycle and is called Data Strobe. During read ycles, DS signifies the time that the DS12887 is to drive the bidirectional bus. In write cycles the trailing lge of DS causes the DS12887 to latch the written data. When the MOT pin is connected to GND, Intel ıs timing is selected. In this mode the DS pin is called Read ($\overline{RD}$). $\overline{RD}$ identifies the time period when e DS12887 drives the bus with read data. The $\overline{RD}$ signal is the same definition as the Output Enable $\overline{OE}$) signal on a typical memory.

/$\overline{W}$ (Read/Write Input) – The R/$\overline{W}$ pin also has two modes of operation. When the MOT pin is ɔnnected to V_{CC} for Motorola timing, R/$\overline{W}$ is at a level which indicates whether the current cycle is a ad or write. A read cycle is indicated with a high level on R/$\overline{W}$ while DS is high. A write cycle is ıdicated when R/$\overline{W}$ is low during DS.

/hen the MOT pin is connected to GND for Intel timing, the R/$\overline{W}$ signal is an active low signal called /R. In this mode the R/$\overline{W}$ pin has the same meaning as the Write Enable signal ($\overline{WE}$) on generic RAMs.

$\overline{S}$ (Chip Select Input) – The Chip Select signal must be asserted low for a bus cycle in the DS12887 to e accessed. $\overline{CS}$ must be kept in the active state during DS and AS for Motorola timing and during $\overline{RD}$ nd $\overline{WR}$ for Intel timing. Bus cycles which take place without asserting $\overline{CS}$ will latch addresses but no ccess will occur. When V_{CC} is below 4.25 volts, the DS12887 internally inhibits access cycles by ternally disabling the $\overline{CS}$ input. This action protects both the real time clock data and RAM data during ɔwer outages.

$\overline{RQ}$ (Interrupt Request Output) – The $\overline{IRQ}$ pin is an active low output of the DS12887 that can be sed as an interrupt input to a processor. The $\overline{IRQ}$ output remains low as long as the status bit causing the terrupt is present and the corresponding interrupt–enable bit is set. To clear the $\overline{IRQ}$ pin the processor rogram normally reads the C register. The $\overline{RESET}$ pin also clears pending interrupts.

/hen no interrupt conditions are present, the $\overline{IRQ}$ level is in the high impedance state. Multiple terrupting devices can be connected to an $\overline{IRQ}$ bus. The $\overline{IRQ}$ bus is an open drain output and requires an xternal pullup resistor.

$\overline{ESET}$ (Reset Input) – The $\overline{RESET}$ pin has no effect on the clock, calendar, or RAM. On power–up the $\overline{ESET}$ pin can be held low for a time in order to allow the power supply to stabilize. The amount of time at $\overline{RESET}$ is held low is dependent on the application. However, if $\overline{RESET}$ is used on power–up, the me $\overline{RESET}$ is low should exceed 200 ms to make sure that the internal timer that controls the DS12887 ı power-up has timed out. When $\overline{RESET}$ is low and V_{CC} is above 4.25 volts, the following occurs:

Periodic Interrupt Enable (PEI) bit is cleared to 0.
Alarm Interrupt Enable (AIE) bit is cleared to 0.
Update Ended Interrupt Flag (UF) bit is cleared to 0.
Interrupt Request Status Flag (IRQF) bit is cleared to 0.
Periodic Interrupt Flag (PF) bit is cleared to 0.
The device is not accessible until $\overline{\text{RESET}}$ is returned high.
Alarm Interrupt Flag (AF) bit is cleared to 0.
$\overline{\text{IRQ}}$ pin is in the high impedance state.
Square Wave Output Enable ($\overline{\text{SQWE}}$) bit is cleared to 0.
Update Ended Interrupt Enable (UIE) is cleared to 0.

a typical application $\overline{\text{RESET}}$ can be connected to V_{CC}. This connection will allow the DS12887 to go in
d out of power fail without affecting any of the control registers.

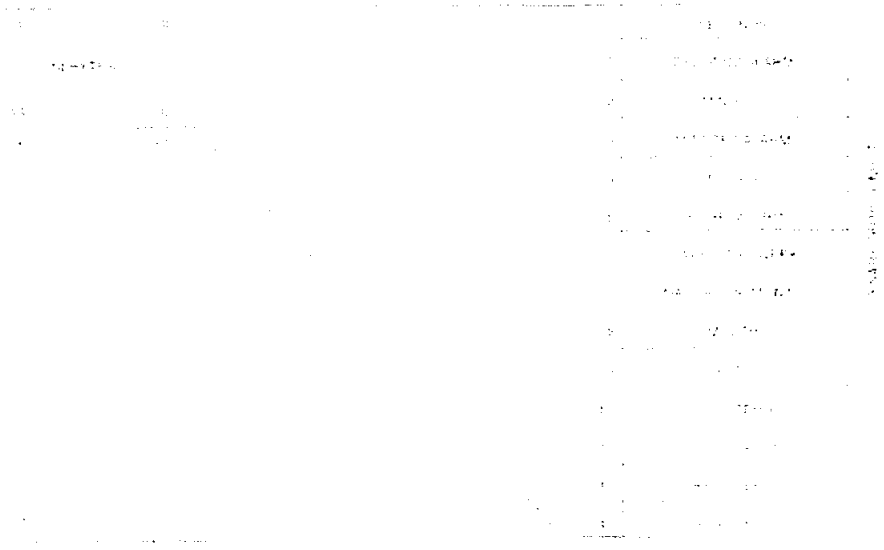
DDRESS MAP

e address map of the DS12887 is shown in Figure 2. The address map consists of 114 bytes of user
AM, 10 bytes of RAM that contain the RTC time, calendar, and alarm data, and 4 bytes which are used
control and status. All 128 bytes can be directly written or read except for the following:

Registers C and D are read–only.
Bit 7 of Register A is read–only.
The high order bit of the seconds byte is read–only.

e contents of four registers (A,B,C, and D) are described in the "Registers" section.

DDRESS MAP DS12887 Figure 2



IME, CALENDAR AND ALARM LOCATIONS

he time and calendar information is obtained by reading the appropriate memory bytes. The time, ılendar, and alarm are set or initialized by writing the appropriate RAM bytes. The contents of the 10 me, calendar, and alarm bytes can be either Binary or Binary–Coded Decimal (BCD) format. Before riting the internal time, calendar, and alarm registers, the SET bit in Register B should be written to a gic 1 to prevent updates from occurring while access is being attempted. In addition to writing the 10 me, calendar, and alarm registers in a selected format (binary or BCD), the data mode bit (DM) of egister B must be set to the appropriate logic level. All 10 time, calendar, and alarm bytes must use the me data mode. The set bit in Register B should be cleared after the data mode bit has been written to low the real time clock to update the time and calendar bytes. Once initialized, the real time clock akes all updates in the selected mode. The data mode cannot be changed without reinitializing the 10 ta bytes. Table 2 shows the binary and BCD formats of the 10 time, calendar, and alarm locations. The 4–12 bit cannot be changed without reinitializing the hour locations. When the 12–hour format is lected, the high order bit of the hours byte represents PM when it is a logic 1. The time, calendar, and arm bytes are always accessible because they are double buffered. Once per second the 10 bytes are lvanced by 1 second and checked for an alarm condition. If a read of the time and calendar data occurs uring an update, a problem exists where seconds, minutes, hours, etc. may not correlate. The probability reading incorrect time and calendar data is low. Several methods of avoiding any possible incorrect me and calendar reads are covered later in this text.

he three alarm bytes can be used in two ways. First, when the alarm time is written in the appropriate ours, minutes, and seconds alarm locations, the alarm interrupt is initiated at the specified time each day the alarm enable bit is high. The second use condition is to insert a "don't care" state in one or more of e three alarm bytes. The "don't care" code is any hexadecimal value from C0 to FF. The two most gnificant bits of each byte set the "don't care" condition when at logic 1. An alarm will be generated ch hour when the "don't care" bits are set in the hours byte. Similarly, an alarm is generated every inute with "don't care" codes in the hours and minute alarm bytes. The "don't care" codes in all three arm bytes create an interrupt every second.

IME, CALENDAR AND ALARM DATA MODES Table 2

ADDRESS OCATION	FUNCTION	DECIMAL RANGE	RANGE	
			BINARY DATA MODE	BCD DATA MODE
0	Seconds	0-59	00-3B	00-59
1	Seconds Alarm	0-59	00-3B	00-59
2	Minutes	0-59	00-3B	00-59
3	Minutes Alarm	0-59	00-3B	00-59
4	Hours-12-hr Mode	1-12	01-0C AM, 81-8C PM	01-12AM, 81-92PM
	Hours-24-hr Mode	0-23	00-17	00-23
5	Hours Alarm-12-hr	1-12	01-0C AM, 81-8C PM	01-12AM, 81-92PM
	Hours Alarm-24-hr	0-23	00-17	00-23
6	Day of the Week Sunday = 1	1-7	01-07	01-07
7	Date of the Month	1-31	01-1F	01-31
8	Month	1-12	01-0C	01-12
9	Year	0-99	00-63	00-99

ISO²-CMOS MT8870D/MT8870D-1
Integrated DTMF Receiver

Features
- Complete DTMF Receiver
- Low power consumption
- Internal gain setting amplifier
- Adjustable guard time
- Central office quality
- Power-down mode
- Inhibit mode
- Backward compatible with MT8870C/MT8870C-1

Applications
- Receiver system for British Telecom (BT) or CEPT Spec (MT8870D-1)
- Paging systems
- Repeater systems/mobile radio
- Credit card systems
- Remote control
- Personal computers
- Telephone answering machine

ISSUE 5 March 1997

Ordering Information

MT8870DE/DE-1	18 Pin Plastic DIP
MT8870DS/DS-1	18 Pin SOIC
MT8870DN/DN-1	20 Pin SSOP

-40 °C to +85 °C

Description

The MT8870D/MT8870D-1 is a complete DTMF receiver integrating both the bandsplit filter and digital decoder functions. The filter section uses switched capacitor techniques for high and low group filters; the decoder uses digital counting techniques to detect and decode all 16 DTMF tone-pairs into a 4-bit code. External component count is minimized by on chip provision of a differential input amplifier, clock oscillator and latched three-state bus interface.

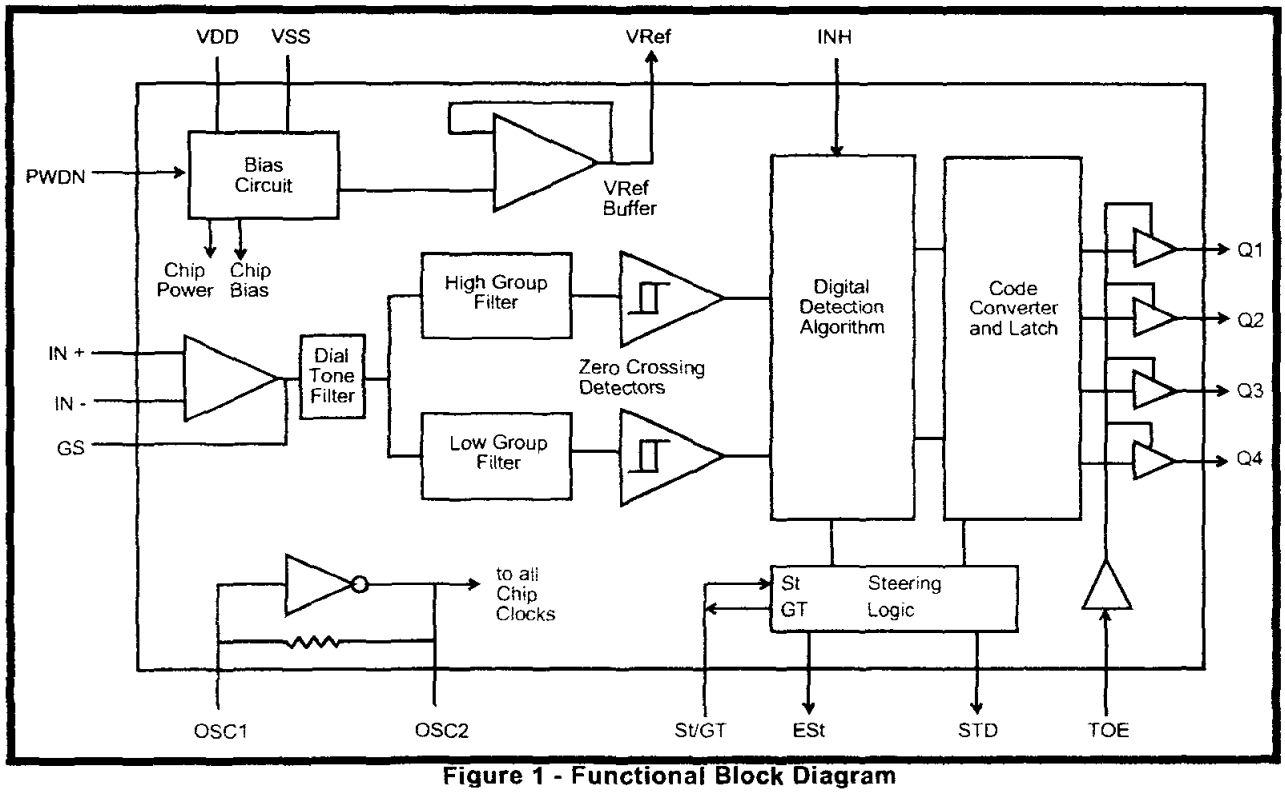


Figure 1 - Functional Block Diagram

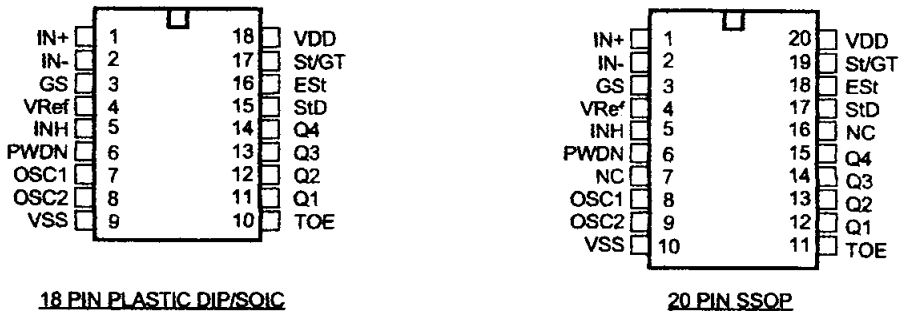


Figure 2 - Pin Connections

Pin Description

Pin #		Name	Description
18	20		
1	1	IN+	**Non-Inverting Op-Amp (Input).**
2	2	IN-	**Inverting Op-Amp (Input).**
3	3	GS	**Gain Select.** Gives access to output of front end differential amplifier for connection of feedback resistor.
4	4	V_{Ref}	**Reference Voltage (Output).** Nominally $V_{DD}/2$ is used to bias inputs at mid-rail (see Fig. 6 and Fig. 10).
5	5	INH	**Inhibit (Input).** Logic high inhibits the detection of tones representing characters A, B, C and D. This pin input is internally pulled down.
6	6	PWDN	**Power Down (Input).** Active high. Powers down the device and inhibits the oscillator. This pin input is internally pulled down.
7	8	OSC1	**Clock (Input).**
8	9	OSC2	**Clock (Output).** A 3.579545 MHz crystal connected between pins OSC1 and OSC2 completes the internal oscillator circuit.
9	10	V_{SS}	**Ground (Input).** 0V typical.
10	11	TOE	**Three State Output Enable (Input).** Logic high enables the outputs Q1-Q4. This pin is pulled up internally.
11-14	12-15	Q1-Q4	**Three State Data (Output).** When enabled by TOE, provide the code corresponding to the last valid tone-pair received (see Table 1). When TOE is logic low, the data outputs are high impedance.
15	17	StD	**Delayed Steering (Output).** Presents a logic high when a received tone-pair has been registered and the output latch updated; returns to logic low when the voltage on St/GT falls below V_{TSt}.
16	18	ESt	**Early Steering (Output).** Presents a logic high once the digital algorithm has detected a valid tone pair (signal condition). Any momentary loss of signal condition will cause ESt to return to a logic low.
17	19	St/GT	**Steering Input/Guard time (Output) Bidirectional.** A voltage greater than V_{TSt} detected at St causes the device to register the detected tone pair and update the output latch. A voltage less than V_{TSt} frees the device to accept a new tone pair. The GT output acts to reset the external steering time-constant; its state is a function of ESt and the voltage on St.
18	20	V_{DD}	**Positive power supply (Input).** +5V typical.
	7, 16	NC	No Connection.

Power-down and Inhibit Mode

A logic high applied to pin 6 (PWDN) will power down the device to minimize the power consumption in a standby mode. It stops the oscillator and the functions of the filters.

Inhibit mode is enabled by a logic high input to the pin 5 (INH). It inhibits the detection of tones representing characters A, B, C, and D. The output code will remain the same as the previous detected code (see Table 1).

Differential Input Configuration

The input arrangement of the MT8870D/MT8870D-1 provides a differential-input operational amplifier as well as a bias source (V_{Ref}) which is used to bias the inputs at mid-rail. Provision is made for connection of a feedback resistor to the op-amp output (GS) for adjustment of gain. In a single-ended configuration, the input pins are connected as shown in Figure 10 with the op-amp connected for unity gain and V_{Ref} biasing the input at $^1/_2V_{DD}$. Figure 6 shows the differential configuration, which permits the adjustment of gain with the feedback resistor R_5.

Crystal Oscillator

The internal clock circuit is completed with the addition of an external 3.579545 MHz crystal and is normally connected as shown in Figure 10 (Single-Ended Input Configuration). However, it is possible to configure several MT8870D/MT8870D-1 devices employing only a single oscillator crystal. The oscillator output of the first device in the chain is coupled through a 30 pF capacitor to the oscillator input (OSC1) of the next device. Subsequent devices are connected in a similar fashion. Refer to Figure 7 for details. The problems associated with unbalanced loading are not a concern with the arrangement shown, i.e., precision balancing capacitors are not required.

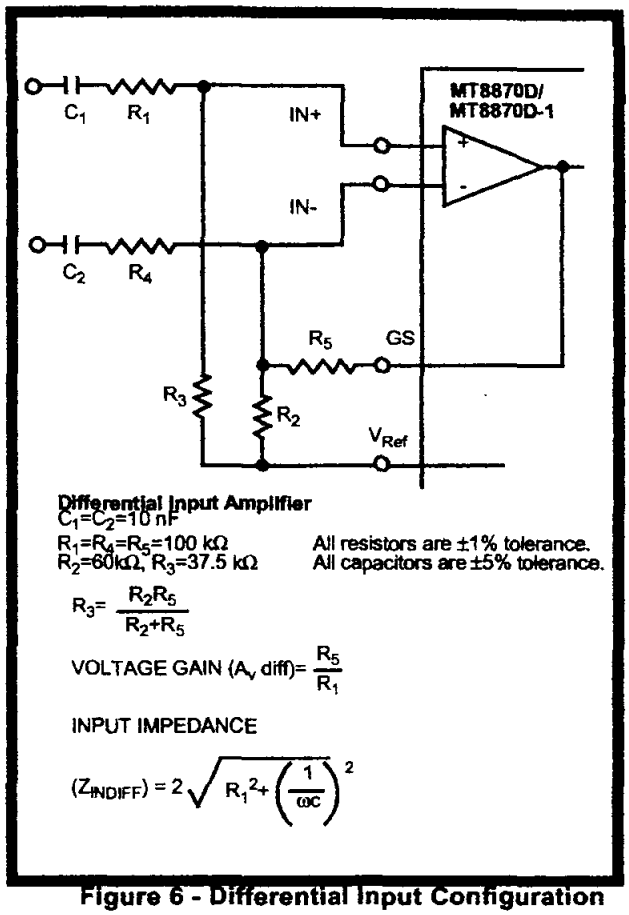


Figure 6 - Differential Input Configuration

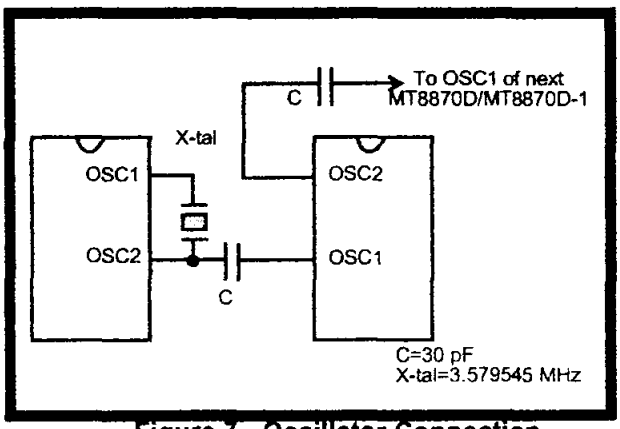


Figure 7 - Oscillator Connection

Parameter	Unit	Resonator
R1	Ohms	10.752
L1	mH	.432
C1	pF	4.984
C0	pF	37.915
Qm	-	896.37
Δf	%	±0.2%

Table 2. Recommended Resonator Specifications

Note: Qm=quality factor of RLC model, i.e., $1/2\Pi fR1C1$.

lications

EIVER SYSTEM FOR BRITISH TELECOM
: POR 1151

circuit shown in Fig. 9 illustrates the use of 870D-1 device in a typical receiver system. BT defines the input signals less than -34 dBm as on-operate level. This condition can be attained hoosing a suitable values of R_1 and R_2 to de 3 dB attenuation, such that -34 dBm input il will correspond to -37 dBm at the gain setting iS of MT8870D-1. As shown in the diagram, the onent values of R_3 and C_2 are the guard time rements when the total component tolerance is For better performance, it is recommended to he non-symmetric guard time circuit in Fig. 8.

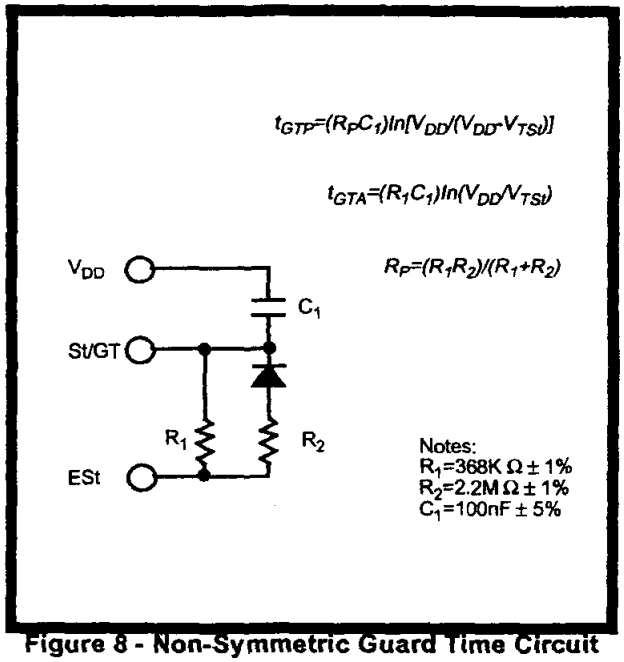
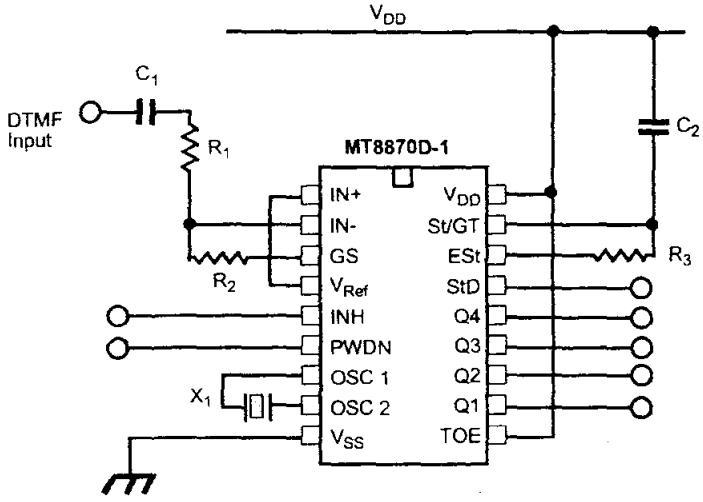


Figure 8 - Non-Symmetric Guard Time Circuit



NOTES:
R_1 = 102KΩ ± 1%
R_2 = 71.5KΩ ± 1%
R_3 = 390KΩ ±1 %
C_1,C_2 = 100 nF ± 5%
X_1 = 3.579545 MHz ± 0.1%
V_{DD} = 5.0V ± 5%

Figure 9 - Single-Ended Input Configuration for BT or CEPT Spec

BIODATA

Nama	: VALENS SIGIT HASCARYO
Nrp	: 5103097064
NIRM	: 97.7.003.31073.38737
Alamat	: KALIJUDAN 158, SURABAYA
Tempat/Tanggal ahir	: SURABAYA,/26 JULI 1978
Agama	: KATOLIK

Riwayat Pendidikan:

- S.D.K. Yohanes Gabriel, Surabaya, Tahun 1985-1991
- S.M.P.K. St. Stanislaus I, Surabaya, Tahun 1991-1994
- S.M.U.K. St. Louis I, Surabaya, Tahun 1994-1997
- Mahasiswa Universitas Katolik Widya Mandala Fakultas Teknik Jurusan Teknik Elektro, Surabaya, Angkatan 1997.

Selama kuliah aktif sebagai :

- Anggota Himpunan Mahasiswa Jurusan Fakultas Teknik Sie Seminar periode 1998 – 1999
- Sekertaris Himpunan Mahasiswa Jurusan Fakultas Teknik periode 1999 – 2000
- Bendahara Training Kader periode 1999 – 2000
- Asisten Praktikum Pengukuran Besaran Listrik