I^{2}C Interface to 8051 Microcontroller

Application Staff

Introduction to I^{2}C

The I^{2}C (Inter-Integrated Circuit) bus is a 2-wire serial bus which provides a small networking system for circuits sharing a common bus. The devices on the bus can vary from microcontrollers to LCD drivers to E^{2}PROMs.

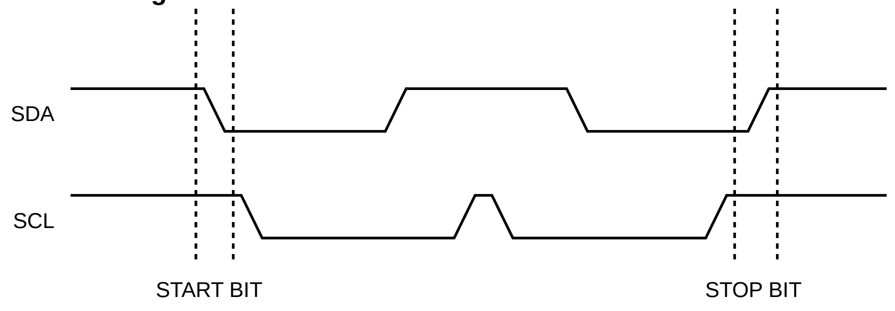
Two bi-directional lines, a serial data (SDA) and a serial clock (SCL) line, transmit data between the devices connected to the bus. Each device has a unique address to differentiate it from the other devices on the bus, and each is configured either as a master or a slave when performing data transfers (see Table 1). A master is the device which initiates a data transfer and generates the clock signals necessary for the transfer. Any device that is addressed is considered a slave. The I^{2}C bus is a multi-master bus, which means that more than one device that is capable of controlling the bus can be connected to it.

Each transmission on the bus begins with the Master sending a Start condition and ends with a Stop condition (see Figure 1). The Master then sends the address of the particular slave device it is requesting. The first four bits of this slave address are fixed as 1010. The next three bits specify a combination of the device address bit(s) and which 2K array of the memory is being addressed (see Figure 2). The last bit of the slave address specifies whether a read or write operation is to be performed. When this bit is a “1”, a read operation is performed, and when it is a “0”, a write operation is performed.

Table 1. Definition of I^{2}C Bus Terminology

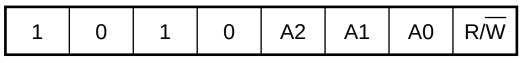
Term	Description
Transmitter	The device which sends the data to the bus.
Receiver	The device which receives the data from the bus.
Master	The device which initiates a transfer, generates clock signals and terminates a transfer.
Slave	The device addressed by a master.
Multi-Master	More than one master can attempt to control the bus at the same time without corrupting the message.
Arbitration	Procedure to ensure that, if more than one master simultaneously tries to control the bus, only one is allowed to do so, and the message is not corrupted.
Synchronization	Procedure to synchronize the clock signals of two or more devices.

Figure 1. START/STOP Timing



5194 FHD F01

Figure 2. Slave Address Bits



1	0	1	0	A2	A1	A0	R/$\overline{W}$

5194 FHD F07

After the Master sends a Start condition, the slave (E^2PROM) monitors the bus and responds with an acknowledge when its address matches the transmitted slave address (see Figure 3). The device then performs a read or write operation depending on the state of the $R/\overline{W}$ bit.

CAT24CXX Interface to 8051 Microcontroller

Catalyst's I^2C family of devices interfaces directly with industry standard microcontrollers such as the Intel MCS-51 family. This family includes 8031/8051 and 8032/8052 (ROMless/ROM) family types.

Catalyst I^2C E^2PROMs are 2-wire interface, nonvolatile memories ranging from 2K bits (CAT24WC02) to 64K bits (CAT24WC64) in density. They adhere to the I^2C protocol which uses 2 lines, a data (SDA) and serial clock (SCL) line for all transmissions, as described above.

The CAT24WC02 E^2PROM has an 8 byte page write buffer and a write protect pin for inadvertent write protection. The CAT24WC04, CAT24WC08 and CAT24WC16 devices have 16 byte page write buffers. Up to eight CAT24WC02 devices, four CAT24WC04 devices, two CAT24WC08 devices and one CAT24WC16 device may be connected to an I^2C bus and addressed independently. The CAT24WC32/64 has a 32-byte Page Write buffer and up to eight devices may be connected to an I^2C bus and addressed independently. Unique addressing is accomplished through hard-wiring address pins A0, A1 and A2 on each device. An example program follows that demonstrates simple byte write and byte read routines as well as page mode and sequential read routines using an 8051 microcontroller. Figure 4 shows a simple hardware interface.

Figure 3. ACKNOWLEDGE Timing

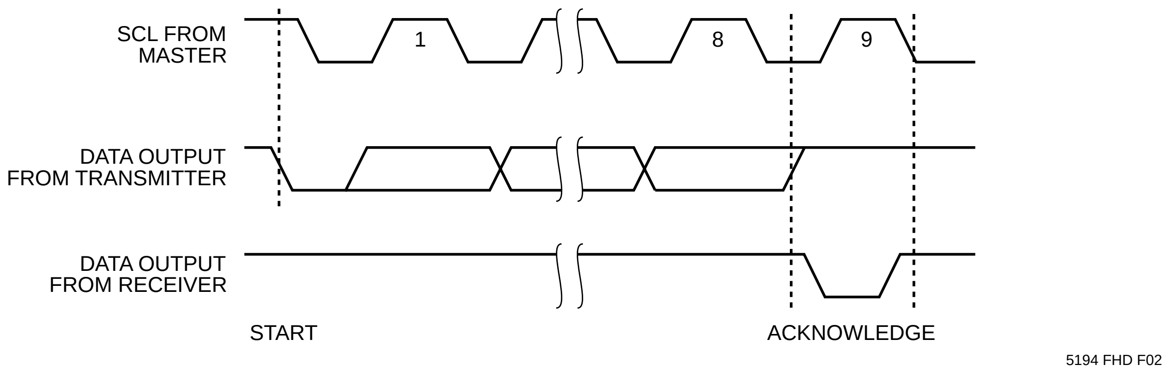
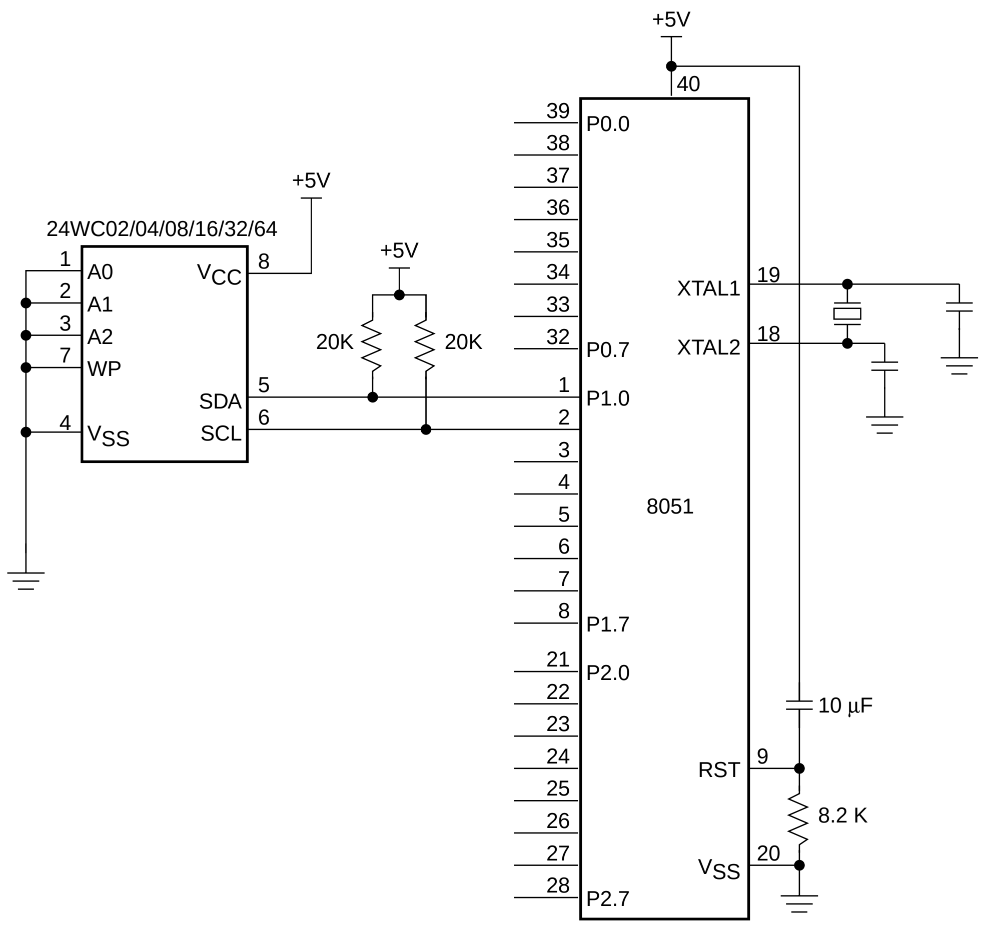


Figure 4. I²C Interface to 8051 Microcontroller



```
<< ASM51 >> CROSS ASSEMBLER VER.2.5M    ASSEMBLE LIST DATE:           PAGE:   1

 LOC. OBJECT            LINE    STATEMENT                          I2C_8051.ASM

                          1
;***********************************************************************
                          2    ; THE FOLLOWING CODE SHOWS AN INTERFACE BETWEEN AN 8051
MICROCONTROLLER
                          3    ; AND CATALYST'S I2C FAMILY OF EEPROMS.
                          4    ;
                          5    ; IT DEMONSTRATES A BYTE WRITE/BYTE READ ROUTINE AND A PAGE MODE
                          6    ; WRITE/SEQUENTIAL READ ROUTINE.  IT USES TWO LINES FROM PORT 1
                          7    ; (P1.0 AND P1.1) OF THE 8051 TO COMMUNICATE WITH THE CAT24CXX.
                          8    ;
                          9    ; THIS PROGRAM WILL WORK WITH THE CAT24WC02/04/08/16 DEVICES.
NOTE:
                         10    ; THE 24WC02 HAS AN 8 BYTE PAGE BUFFER AND 24WC04/08/16 HAVE A
                         11    ; 16-BYTE PAGE
BUFFER.;*********************************************************************
                         12
    0090                 13    SCL         BIT    P1.0                    ;SCL BIT IS PORT 1,
BIT 0
    0091                 14    SDA         BIT    P1.1                    ;SDA BIT IS PORT 1,
BIT 1
    0005                 15    SLV_ADDR    EQU    0101B                   ;FIXED SLAVE ADDRESS
BITS
    REG                        16    DATAOUT     EQU    R5                      ;DATA READ
FROM DEVICE
    0085                 17    ACK_READ    EQU    10000101B               ;READ FOR ACK POLLING
                         18
 ——                      19                DSEG
 0030                    20                ORG    0030H
                         21
 0030                    22    PAGE_DATA:  DS     1
 0031                    23    BLK_ADDR:   DS     1
 0032                    24    BYTE_ADDR:  DS     1
 0033                    25    BYTE_DATA:  DS     1
                         26
 0040                    27                ORG    40H
 0040                    28    STACK:      DS     31
                         29
 ——                      30                CSEG
 0040                    31                ORG    0040H
 0040 02 01 00                 32                LJMP   BEGIN
                         33
 0100                    34                ORG    0100H
 0100 75 81 40                 35    BEGIN:      MOV    SP,#STACK               ;INITIALIZE
STACK POINTER
                         36
 0103 75 31 00                 37                MOV    BLK_ADDR,#000B          ;INITIALIZE 2K
BLOCK
 0106 75 33 55                 38                MOV    BYTE_DATA,#55H          ;BYTE DATA
 0109 75 32 00                 39                MOV    BYTE_ADDR,#00H          ;BYTE ADDRESS
 010C 75 30 AA                 40                MOV    PAGE_DATA,#0AAH         ;PAGE DATA
                         41
 010F 31 45 [0145]       42                ACALL  PAGE_WR                 ;CALL PAGE WRITE
ROUTINE
 0111 51 1D [021D]       43                ACALL  SEQ_RD                  ;CALL SEQ. READ
ROUTINE
 0113 31 1A [011A]       44                ACALL  BYTE_WR                 ;CALL BYTE WRITE
ROUTINE
 0115 31 D6 [01D6]       45                ACALL  SELECT_RD               ;CALL BYTE READ
```

```
ROUTINE
 0117 02 01 17              46    DONE:       LJMP   DONE                     ;LOOP UNTIL
RESET OCCURS
                           47
                           48    ;******************************************
                           49
                           50    ;************** BYTE WRITE *****************
                           51
 011A 31 95 [0195]         52    BYTE_WR:     ACALL  START_BIT                ;SEND START BIT
 011C 74 05                53                 MOV    A,#SLV_ADDR              ;FIRST 4 SLAVE AD-
DRESS
 011E 7F 04                54                 MOV    R7,#4H                   ;BITS
 0120 31 89 [0189]         55                 ACALL  SHFTO
 0122 E5 31                56                 MOV    A,BLK_ADDR               ;2K BLOCK ADDRESS
 0124 7F 03                57                 MOV    R7,#3H
 0126 31 89 [0189]         58                 ACALL  SHFTO
<< ASM51 >> CROSS ASSEMBLER VER.2.5M    ASSEMBLE LIST DATE:           PAGE:   2

 LOC. OBJECT              LINE    STATEMENT                      I2C_8051.ASM

 0128 74 00                59                 MOV    A,#00H                   ;R/W BIT SET TO 0 FOR
 012A 7F 01                60                 MOV    R7,#1H                   ;WRITE
 012C 31 89 [0189]         61                 ACALL  SHFTO
 012E 31 AA [01AA]         62                 ACALL  SLAVE_ACK
                           63
 0130 E5 32                64                 MOV    A,BYTE_ADDR              ;BYTE ADDRESS
 0132 7F 08                65                 MOV    R7,#8H
 0134 31 89 [0189]         66                 ACALL  SHFTO
 0136 31 AA [01AA]         67                 ACALL  SLAVE_ACK
 0138 E5 33                68                 MOV    A,BYTE_DATA              ;BYTE DATA
 013A 7F 08                69                 MOV    R7,#8H
 013C 31 89 [0189]         70                 ACALL  SHFTO
 013E 31 AA [01AA]         71                 ACALL  SLAVE_ACK
 0140 31 A1 [01A1]         72                 ACALL  STOP_BIT                 ;STOP BIT
 0142 31 74 [0174]         73                 ACALL  ACK_POL                  ;CALL ACK POLLING,
WAIT
 0144 22                   74                 RET                             ;FOR END OF WRITE
CYCLE
                           75    ;******************************************
                           76
                           77    ;************** PAGE WRITE *****************
                           78
 0145 31 95 [0195]         79    PAGE_WR:     ACALL  START_BIT                ;SEND START BIT
 0147 74 05                80                 MOV    A,#SLV_ADDR              ;FIRST 4 SLAVE AD-
DRESS
 0149 7F 04                81                 MOV    R7,#4H                   ;BITS
 014B 31 89 [0189]         82                 ACALL  SHFTO
 014D E5 31                83                 MOV    A,BLK_ADDR               ;2K BLOCK ADDRESS
 014F 7F 03                84                 MOV    R7,#3H
 0151 31 89 [0189]         85                 ACALL  SHFTO
 0153 74 00                86                 MOV    A,#00H                   ;R/W BIT SET TO 0 FOR
 0155 7F 01                87                 MOV    R7,#1H                   ;WRITE
 0157 31 89 [0189]         88                 ACALL  SHFTO
 0159 31 AA [01AA]         89                 ACALL  SLAVE_ACK
 015B E5 32                90                 MOV    A,BYTE_ADDR              ;BYTE ADDRESS
 015D 7F 08                91                 MOV    R7,#8H
 015F 31 89 [0189]         92                 ACALL  SHFTO
 0161 31 AA [01AA]         93                 ACALL  SLAVE_ACK
 0163 7C 0F                94                 MOV    R4,#0FH
                           95    NEXT_DATA:                                   ;WRITE 16 BYTES TO
 0165 E5 30                96                 MOV    A,PAGE_DATA              ;EEPROM
 0167 7F 08                97                 MOV    R7,#8H
 0169 31 89 [0189]    98                ACALL  SHFTO
 016B 31 AA [01AA]    99                ACALL  SLAVE_ACK
 016D DC F6 [0165]   100                DJNZ   R4,NEXT_DATA
 016F 31 A1 [01A1]   101                ACALL  STOP_BIT
 0171 31 74 [0174]   102                ACALL  ACK_POL                  ;CALL ACK
POLLING,WAIT
 0173 22             103                RET                             ;FOR END OF WRITE
CYCLE
                     104    ;*******************************************
                     105
                     106    ;************** ACK_POL ********************
                     107
 0174 7B 40          108    ACK_POL:    MOV    R3,#40H                  ;# OF TIMES TO POLL
 0176 DB 02 [017A]   109    ACK_LOOP:   DJNZ   R3,DONE_YET              ;DEVICE
 0178 80 0C [0186]   110                SJMP   DN_ACKPOL
 017A 31 95 [0195]   111    DONE_YET:   ACALL  START_BIT                ;SEND START BIT
 017C 74 85          112                MOV    A,#ACK_READ              ;SEND READ
 017E 7F 08          113                MOV    R7,#8H
 0180 31 89 [0189]   114                ACALL  SHFTO
 0182 31 AA [01AA]   115                ACALL  SLAVE_ACK                ;SEND ACKNOWLEDGE
 0184 40 F0 [0176]   116                JC     ACK_LOOP                 ;LOOP IF NO ACK RCVD,
<< ASM51 >> CROSS ASSEMBLER VER.2.5M    ASSEMBLE LIST DATE:            PAGE:   3

 LOC. OBJECT            LINE    STATEMENT                       I2C_8051.ASM

                     117                                            ;JUMP IF ACK RCVD
 0186 31 A1 [01A1]   118    DN_ACKPOL:  ACALL  STOP_BIT                 ;SEND STOP BEFORE
RETURN
 0188 22             119                RET
                     120    ;*******************************************
                     121
                     122    ;************** SHFTO **********************
                     123
 0189 C2 90          124    SHFTO:      CLR    SCL
 018B C2 90          125    NXTSHF:     CLR    SCL
 018D 13             126                RRC    A                        ;ROTATE DATA INTO
CARRY
 018E 92 91          127                MOV    SDA,C                    ;SEND CARRY TO SDA
 0190 D2 90          128                SETB   SCL
 0192 DF F7 [018B]   129                DJNZ   R7,NXTSHF
 0194 22             130                RET
                     131    ;*******************************************
                     132
                     133    ;************** START BIT ******************
                     134
 0195 D2 90          135    START_BIT:  SETB   SCL                      ;START BIT
 0197 00             136                NOP
 0198 D2 91          137                SETB   SDA
 019A 00             138                NOP
 019B C2 91          139                CLR    SDA
 019D 00             140                NOP
 019E C2 90          141                CLR    SCL
 01A0 22             142                RET
                     143    ;*******************************************
                     144
                     145    ;************** STOP BIT *******************
                     146
 01A1 C2 91          147    STOP_BIT:   CLR    SDA                      ;STOP BIT
 01A3 00            148              NOP
 01A4 D2 90         149              SETB   SCL
 01A6 00            150              NOP
 01A7 D2 91         151              SETB   SDA
 01A9 22            152              RET
                    153     ;*******************************************
                    154
                    155     ;************** SLAVE ACKNOWLEDGE **********
                    156
 01AA 00            157     SLAVE_ACK:  NOP
 01AB 00            158              NOP
 01AC C2 90         159              CLR    SCL                      ;SLAVE ACKNOWLEDGE
BIT
 01AE 00            160              NOP
 01AF D2 91         161              SETB   SDA
 01B1 00            162              NOP
 01B2 00            163              NOP
 01B3 D2 90         164              SETB   SCL
 01B5 00            165              NOP
 01B6 00            166              NOP
 01B7 00            167              NOP
 01B8 A2 91         168              MOV    C,SDA                    ;READ STATE OF SDA,
 01BA C2 90         169              CLR    SCL                      ;SAVE TO CARRY
 01BC 22            170              RET
<< ASM51 >> CROSS ASSEMBLER VER.2.5M    ASSEMBLE LIST DATE:           PAGE:   4

 LOC. OBJECT            LINE    STATEMENT                      I2C_8051.ASM

                    171     ;*******************************************
                    172
                    173     ;************** MASTER ACKNOWLEDGE *********
                    174
                    175     MSTR_ACK:
 01BD C2 90         176              CLR    SCL                      ;MASTER ACKNOWLEDGE
BIT
 01BF 00            177              NOP
 01C0 C2 91         178              CLR    SDA
 01C2 00            179              NOP
 01C3 00            180              NOP
 01C4 D2 90         181              SETB   SCL
 01C6 00            182              NOP
 01C7 C2 90         183              CLR    SCL
 01C9 00            184              NOP
 01CA D2 91         185              SETB   SDA
 01CC 22            186              RET
                    187     ;*******************************************
                    188
                    189     ;************** NO ACKNOWLEDGE *************
                    190
 01CD D2 91         191     NO_ACK:     SETB   SDA                      ;NO ACKNOWLEDGE
 01CF 00            192              NOP
 01D0 D2 90         193              SETB   SCL
 01D2 00            194              NOP
 01D3 C2 90         195              CLR    SCL
 01D5 22            196              RET
                    197     ;*******************************************
                    198
                    199     ;************** SELECTIVE READ *************
```

```
                        200
                        201    SELECT_RD:
 01D6 31 95 [0195]      202                ACALL  START_BIT                ;START BIT
                        203
 01D8 74 05             204                MOV    A,#SLV_ADDR              ;DUMMY WRITE TO FIRST
 01DA 7F 04             205                MOV    R7,#4H                   ;2K BLOCK
 01DC 31 89 [0189]      206                ACALL  SHFTO
 01DE E5 31             207                MOV    A,BLK_ADDR               ;2K BLOCK ADDRESS
 01E0 7F 03             208                MOV    R7,#3H
 01E2 31 89 [0189]      209                ACALL  SHFTO
 01E4 74 00             210                MOV    A,#00H                   ;R/W BIT SET TO 0
 01E6 7F 01             211                MOV    R7,#1H                   ;FOR WRITE
 01E8 31 89 [0189]      212                ACALL  SHFTO
 01EA 31 AA [01AA]      213                ACALL  SLAVE_ACK                ;SEND ACKNOWLEDG
                        214
 01EC E5 32             215                MOV    A,BYTE_ADDR              ;ADDRESS TO READ
 01EE 7F 08             216                MOV    R7,#8H
 01F0 31 89 [0189]      217                ACALL  SHFTO
 01F2 31 AA [01AA]      218                ACALL  SLAVE_ACK
                        219
 01F4 31 95 [0195]      220                ACALL  START_BIT                ;NEW START BIT
                        221
 01F6 74 05             222                MOV    A,#SLV_ADDR
 01F8 7F 04             223                MOV    R7,#4H
<< ASM51 >> CROSS ASSEMBLER VER.2.5M    ASSEMBLE LIST DATE:           PAGE:   5

 LOC. OBJECT            LINE    STATEMENT                       I2C_8051.ASM

 01FA 31 89 [0189]      224                ACALL  SHFTO
 01FC E5 31             225                MOV    A,BLK_ADDR               ;2K BLOCK TO READ
 01FE 7F 03             226                MOV    R7,#3H
 0200 31 89 [0189]      227                ACALL  SHFTO
 0202 74 01             228                MOV    A,#1H                    ;R/W BIT SET TO 1
 0204 7F 01             229                MOV    R7,#1H                   ;FOR READ
 0206 31 89 [0189]      230                ACALL  SHFTO
 0208 31 AA [01AA]      231                ACALL  SLAVE_ACK
                        232
 020A 7F 08             233                MOV    R7,#8H
 020C D2 90             234    CLOCK8:     SETB   SCL                      ;CLOCK IN DATA
 020E 00                235                NOP
 020F A2 91             236                MOV    C,SDA
 0211 C2 90             237                CLR    SCL
 0213 ED                238                MOV    A,DATAOUT
 0214 33                239                RLC    A                        ;ROTATE NEXT BIT
 0215 FD                240                MOV    DATAOUT,A                ;SAVE ROTATED DATA
 0216 DF F4 [020C]      241                DJNZ   R7,CLOCK8                ;READ 8 BITS OF DATA
 0218 31 CD [01CD]      242                ACALL  NO_ACK
 021A 31 A1 [01A1]      243                ACALL  STOP_BIT
 021C 22                244                RET
                        245    ;*******************************************
                        246
                        247    ;************** SEQUENTIAL READ ************ (Refer to Note 1)
                        248
                        249    SEQ_RD:
 021D 31 95 [0195]      250                ACALL  START_BIT                ;START BIT
                        251
 021F 74 05             252                MOV    A,#SLV_ADDR              ;DUMMY WRITE TO FIRST
 0221 7F 04             253                MOV    R7,#4H                   ;2K BLOCK
 0223 31 89 [0189]      254                ACALL  SHFTO
 0225 E5 31             255                MOV    A,BLK_ADDR               ;2K BLOCK ADDRESS
 0227 7F 03             256                MOV    R7,#3H
 0229 31 89 [0189]   257                ACALL  SHFTO
 022B 74 00          258                MOV    A,#00H                   ;R/W BIT SET TO 0
 022D 7F 01          259                MOV    R7,#1H                   ;FOR WRITE
 022F 31 89 [0189]   260                ACALL  SHFTO
 0231 31 AA [01AA]   261                ACALL  SLAVE_ACK
                     262
 0233 E5 32          263                MOV    A,BYTE_ADDR              ;ADDRESS TO READ
 0235 7F 08          264                MOV    R7,#8H
 0237 31 89 [0189]   265                ACALL  SHFTO
 0239 31 AA [01AA]   266                ACALL  SLAVE_ACK
                     267
 023B 31 95 [0195]   268                ACALL  START_BIT                ;NEW START BIT
                     269
 023D 74 05          270                MOV    A,#SLV_ADDR
 023F 7F 04          271                MOV    R7,#4H
 0241 31 89 [0189]   272                ACALL  SHFTO
 0243 E5 31          273                MOV    A,BLK_ADDR               ;2K BLOCK TO READ
 0245 7F 03          274                MOV    R7,#3H
 0247 31 89 [0189]   275                ACALL  SHFTO
 0249 74 01          276                MOV    A,#1H                    ;R/W BIT SET TO 1
 024B 7F 01          277                MOV    R7,#1H                   ;FOR READ
 024D 31 89 [0189]   278                ACALL  SHFTO
 024F 31 AA [01AA]   279                ACALL  SLAVE_ACK
                     280
<< ASM51 >> CROSS ASSEMBLER VER.2.5M    ASSEMBLE LIST DATE:           PAGE:   6

 LOC. OBJECT            LINE    STATEMENT                      I2C_8051.ASM

 0251 7E 0F          281                MOV    R6,#0FH
 0253 7F 08          282    NXT_BYTE:   MOV    R7,#8H
 0255 D2 90          283    ONE_BYTE:   SETB   SCL                      ;READ 16 BYTES OF
DATA
 0257 00             284                NOP
 0258 C2 90          285                CLR    SCL
 025A 00             286                NOP
 025B DF F8 [0255]   287                DJNZ   R7,ONE_BYTE
 025D 31 BD [01BD]   288                ACALL  MSTR_ACK                 ;ACKNOWLEDGE
 025F DE F2 [0253]   289                DJNZ   R6,NXT_BYTE
                     290
 0261 7F 08          291                MOV    R7,#8H
 0263 D2 90          292    LST_BYTE:   SETB   SCL
 0265 00             293                NOP                             ;READ LAST BYTE
 0266 C2 90          294                CLR    SCL
 0268 00             295                NOP
 0269 DF F8 [0263]   296                DJNZ   R7,LST_BYTE
 026B 31 CD [01CD]   297                ACALL  NO_ACK                   ;NO ACKNOWLEDGE
 026D 31 A1 [01A1]   298                ACALL  STOP_BIT                 ;STOP BIT
 026F 22             299                RET
                     300    ;*****************************************
                     301    END



ASSEMBLY END , ERRORS:0
LAST CODE ADDRESS:026F
```

Note 1: For "Sequential Read" subroutine, see Errata "I2C EEPROM Interface to 8051 Microcontroller - Sequential Read," or contact technical support service at factory.

I²C EEPROM INTERFACE TO 8051 MICROCONTROLLER – SEQUENTIAL READ

(Errata for Application Note “I2C Interface to 8051 Microcontroller”)

The following subroutine shows an implementation of “Sequential Read” for I2C EEPROM device interfaced to 8051 microcontroller.

The routine assumes that the address pointer for EEPROM has already been initialized. The code implements sending of “Start” condition, “Slave address” and “Sequential Reading” of data from EEPROM. The total number of bytes to be read from EEPROM is provided in R6. The data read from EEPROM is stored in a Buffer Memory starting at address indicated by R0.

The routine presented below uses other subroutines defined in Catalyst application note “I2C Interface to 8051 Microcontroller” (START_BIT, SLAVE_ACK, MSTR_ACK, NO_ACK, STOP_BIT, SHFTO).

```
Seq_Read:   MOV   R0,#Buf_Mem       ; R0 = Buffer Memory Address to store data
            MOV   R6,#0FH           ; R6 = (Nr of bytes to be read) – 1

            ACALL START_BIT         ; START Condition
            ACALL SLV_ADD_R         ; Send Slave Address (7 bits) + R/W (=1) bit
            ACALL SLAVE_ACK         ; Generate Slave ACK

NXT_BYTE:   ACALL In_Byte           ; Read one Byte from EEPROM
            MOV   @R0,A             ; Store Data in Buf_Mem
            INC   R0
            ACALL MSTR_ACK          ; Generate Master ACK
            DJNZ  R6, NXT_BYTE      ; Loop for next byte read

LAST_BYTE:  ACALL In_Byte           ; Read last byte
            ACALL NO_ACK            ; No ACK
            ACALL STOP_BIT          ; Stop Condition
            RET



In_Byte:                            ; one byte read from EEPROM into Acc
                                    ; generate clock & serial read 8 bits from SDA
line
            MOV   R7,#8H
Clock8:     SETB  SCL
            NOP
            MOV   C,SDA
            CLR   SCL
            RLC   A                 ; enter read bit from SDA in A
            NOP
            DJNZ  R7,Clock8         ; read next bit
            RET

;**********************************************************************************
```

```
; Attention to the timing requirements for Clock (SCL) t_low and t_high period
according to data sheet
; NOP instructions can be inserted in the case of faster microcontroller clock
cycle
;*********************************************************************************
SLV_ADD_R:                          ; Send 7 bit Slave Address & R/W =1
            MOV   A,#SLV_ADDR       ; Send 4 bits  Slave Addr “1010”
            MOV   R7,#4H
            ACALL SHFTO             ; Shift-out 4 bits
            MOV   A,BLK_ADDR        ;Send the next 3 bits (X for CAT24WC16 / 128)
            MOV   R7,#3H
            ACALL SHFTO
            MOV   A,#1H             ; R/W bit = 1
            MOV   R7,#1H
            ACALL SHFTO
            RET
```

CATALYST

Catalyst Semiconductor, Inc.
Corporate Headquarters
1250 Borregas Avenue
Sunnyvale, CA 94089
Phone: 408.542.1000
Fax: 408.542.1200
www.catsemi.com

Publication #:	6001
Revison:	A
Issue date:	05/07/01
Type:	Final