DSP FUNCTIONAL TESTER

P-708

PROJECT REPORT
SUBMITTED BY :

AARTHE .S
AARTHI .S
JAMUNA RANGANATHAN
LAVANYA RANI .S

GUIDED BY
Ms. P. KAVITHA B.E.,

SPONSORED BY:
PREMIER POLYTRONICS LTD.
COIMBATORE – 641005

In partial fulfillment of the requirements for the award of the degree of BACHELOR OF ENGINEERING in ELECTRICAL AND ELECTRONICS ENGINEERING OF THE BHARATHIAR UNIVERSITY.

P.-708
K.C.T. LIBRARY Coimbatore 641 006

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641006

CERTIFICATE

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641006

CERTIFICATE

This is to certify that the report entitled "DSP FUNCTIONAL TESTER" has been submitted by

AARTHE. S

AARTHI. S

JAMUNA RANGANATHAN

LAVANYA RANI. S

In partial fulfillment of the requirement for the award of the Degree of BACHELOR OF ENGINEERING IN ELECTRICAL AND ELECTRONICS ENGINEERING of Bharathiar University, Coimbatore During the academic year 2001-2002

P. Dauuls 15/03/02

Guide

V. DURAISAMY M.E., M.I.S.T.E.,
Assistant Professor of Elec. Engg.
Kumaraguru College of Tech.
COIMBATORE - 641 006.

Head of the Department

Certified that the candidates with university register numbers ---------- were examined in project work

Viva - Voce Examination held on ----------

Internal Examiner

External Examiner

PREMIER

CERTIFICATE

This is to certify that the following B.E. [Branch : Electrical & Electronics Engineering] students of **KUMARAGURU COLLEGE OF TECHNOLOGY**, Coimbatore, had undertaken their project **"DSP Functional Tester"**, from April 2001 to February 2002 at our Industry and have successfully completed it.

Ms.S. Aarthi
Ms.S. Aarthe
Ms.Jamuna Ranganathan
Ms.S. Lavanya Rani

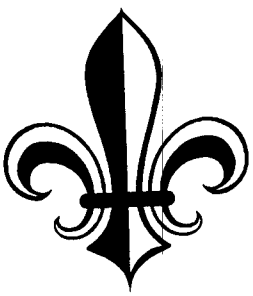
Their performance during that period was found to be good. We wish them all success.

Place : Coimbatore
Date : March 1, 2002

Industry Seal
Premier Polytronics Limited
304, TRICHY ROAD
SINGANALLUR POST
COIMBATORE-641 005.

K K VENKATARAMAN
Vice President - R&D

Important : All information contained herein is confidential and not to be disclosed to parties not specifically noted.
ISO 9001

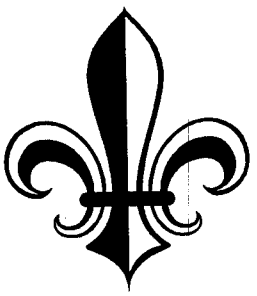


DEDICATED TO OUR PARENTS

WHO SACRIFICED THEIR

TODAY FOR OUR BETTER

TOMMORROW



ACKNOWLEDGEMENTS

ACKNOWLEDGEMENTS

We wish to express our deep sense of gratitude to our internal guide **Ms. P.Kavitha, B.E.**, Lecturer, Department of Electrical and Electronics Engineering for her motivation and guidance.

We thank our former H.O.D **Dr.K.A.Palaniswamy** and our Project Co-ordinator **Mr.V.Duraisamy,M.E** for giving us the opportunity and encouragement to undertake this project.

We wish to express our deep and unfathomable thanks to our Principal **Dr.K.K.Padmanaban B.Sc., Engg, M.Tech, Ph.d.,** and management for their support.

We convey our genuine regards and respect to **Mr. K.K.Venkatraman, B.E., (Hons)** Vice President, R&D PPL, for his untiring efforts and sustained help in accomplishing this project work.

It is our privilege and honour to extend our profound gratitude and express our indebtedness to our guide **Mr. V.Karthikeyan,** Juniour Engineer PPL, from whom we received continuous assistance for the successful completion of our project.

We express our sincere thanks to the director, **PREMIER POLYTRONICS LIMITED** for having given necessary facilities to carryout this project.

SYNOPSIS

SYNOPSIS

Accuracy, Reliability and Versatility are important for any electronics product. The objective of the project is to test the DSP card used in the product Classidata which is used for fault classification of yarn. The Classidata accepts measuring head signals process them, analyses them and classifies them with help of a DSP card. The DSP processor used is TMS320C203. This DSP card is the heart of Classidata. To check the functioning of the DSP card, a hardware circuit and software has been designed .The measuring head signals are generated with the help of a voltage divider circuit for testing purposes. These signals are processed by passing through a Multiplexer, Analog to Digital Converter, and the DSP processor present in the DSP card.

The software is written using the assembly language to test DSP card. The program written is loaded into the EPROM and fixed on to the DSP card. The testing of the card starts with the testing of the Dual Port RAM, Multiplexer, followed by the ADC. The test results are indicated using Light Emitting Diodes.

CONTENTS

CONTENTS

INTRODUCTION

CHAPTER - 1

INTRODUCTION TO CLASSIDATA

Classidata is used to detect yarn faults. Measuring head is a sensing unit, which converts yarn mass into corresponding electric signals. These signals are amplified and send to the DSP card, present inside the PC. The software in the DSP card compares the received signal with the mean signal. The DSP software counts the number of occasions, each fault has occurred. These classification results are converted into text files by the linking software and then linked with the front-end software. The front-end software adds the classification results into its database.

1.1 MAJOR MODULES IN CLASSIDATA

*Measuring Head

*Signal Unit

*Interface unit

*PC

*Printer

*DSP card

MEASURING HEAD

Measuring head is the sensing part of the Classidata installation. It converts the yarn variations into corresponding electrical variations.

SIGNAL UNIT

The signal received from the measuring head is of very low amplitude. So it is amplified and the taken for further processing.

INTERFACE UNIT

Interface unit is used to provide supply voltage for signal units, generate 2mm pulse generation, communication with PC for getting settings, alarm activation.

PC

The PC is used for interaction with the user such as to getting the settings, displaying the classification results, storing the results etc.

PRINTER

A Printer is connected to the PC, for taking hard copy printout of various results.

DSP CARD

The DSP card is fixed inside the PC, which processes the signals from the interface unit. The classification is done in DSP card. The DSP card can be divided into two parts namely

*PC section.

*DSP section.

1.2 DSP CARD-PC SECTION

The PC section is used for communication with the PC for transferring the classification results, mean value data, and yarn running status to display the results to the user through the front-end software. This card uses the ISA Bus of the mother board for the above operations.

The DSP card uses its 16 bit address bus and 16 bit data bus for the data transfer. In order to communicate with the PC, the card has its unique address set by the DIP switches. Comparator logic is used to compare the address place in the ISA bus with the address set in the DIP switches. If both get matched, then the communication between the DSP card and PC takes place. The address and data bus is connected to three Dual Port RAM IC's These DPRAM is accessed by both PC as well as the DSP processor by giving appropriate control signals.

The control signals and chip select logic's are arrived using three Programmable Array Logic ICs. The program memory is stored in the 32K X16 bit EPROM and the Dual Port RAMs (each of 2K X 16 bit) are used as data memory.

DSP Card.

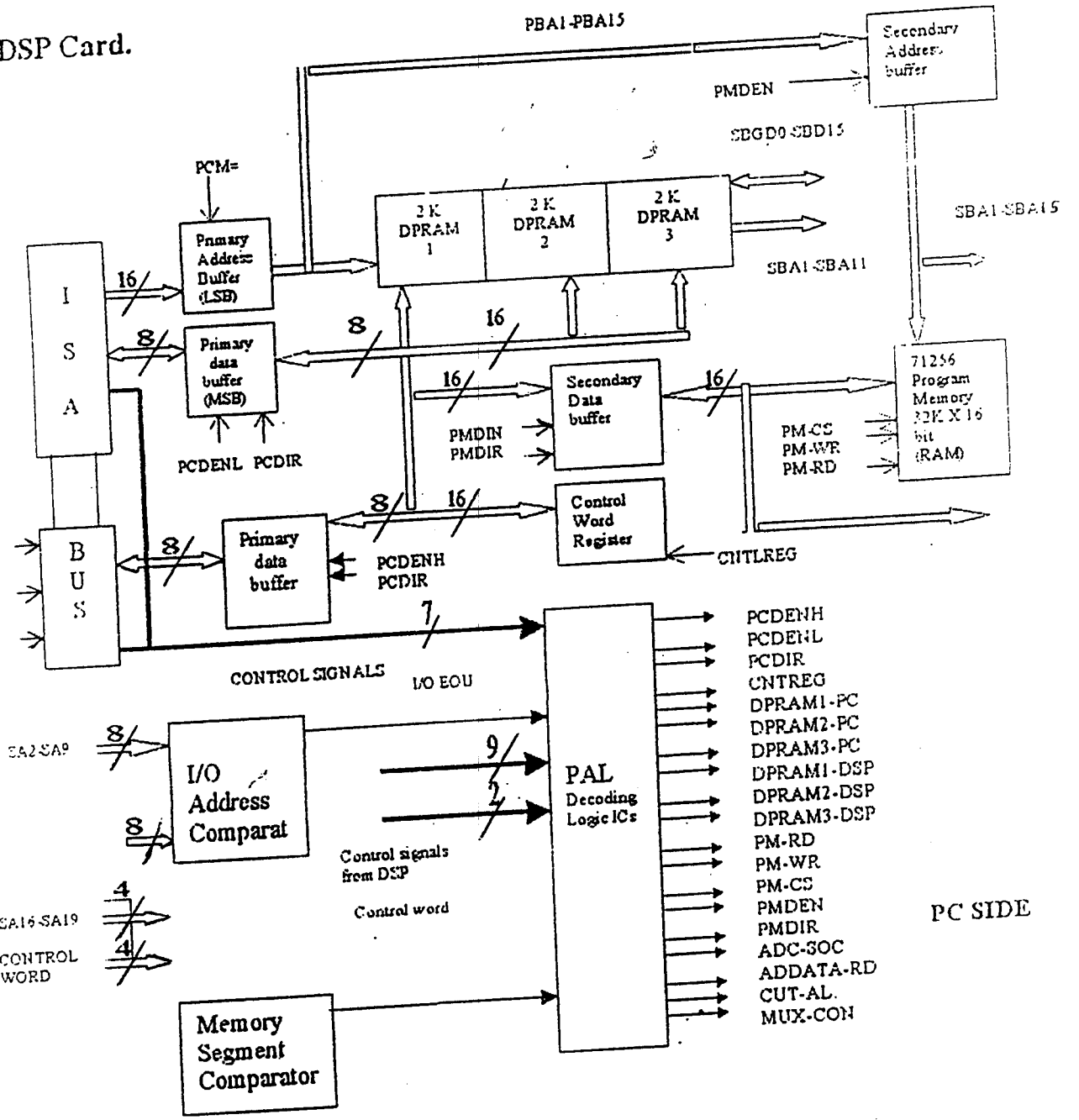
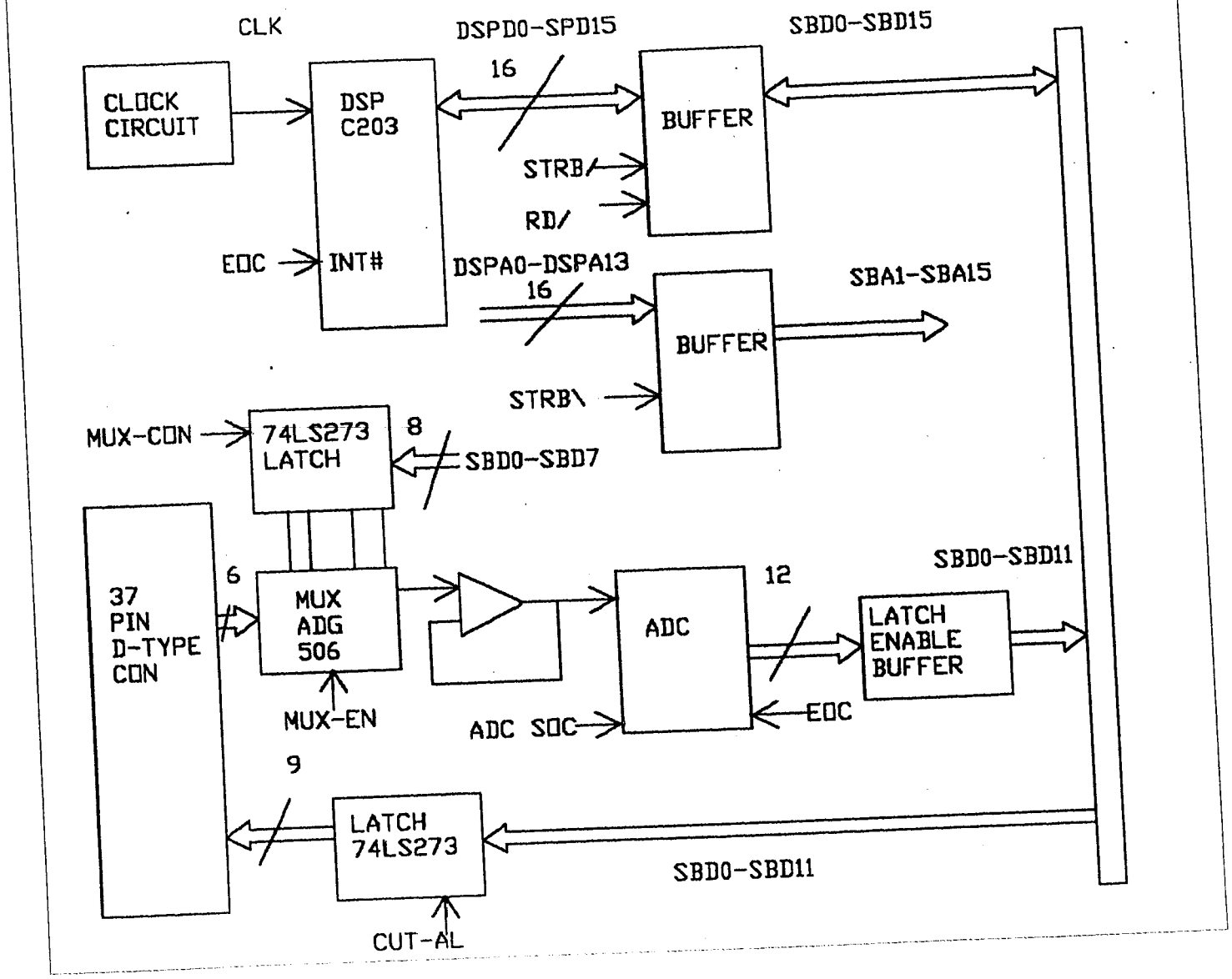


Fig. 1

1.3 DSP CARD-DSP SECTION

The Digital Signal Processor used in this board is TMS320C203.The address and data bus of the DSP processor are buffered and connected to the Program memory and DPRAM.The control signal and chip select logic's for this section is also derived from the same PAL ICs.

The signals from the interface unit are connected with the 37 pin D- the multiplexer IC (ADG 506) and the output of the multiplexer is given to an analog to digital converter IC (SP774BK) through an unity gain amplifier. The control signals for the multiplexer are derived by the DSP processor and it is given through the latch. The output of the ADC is connected to the DSP data bus through a Latch enable buffer. The buffer sends the data to the DSP processor. The processor compares the received signal value with the mean signal value .The cut signals for the measuring heads are derived by the processor and it is given to the 37 pin connector through a latch enable buffer IC (74LS273).



DSP CARD- DSP SECTION

Fig · 2

1.4 FUNCTIONS OF DSP CARD IN CLASSIDATA:

- Getting analog signal related to yarn mass from interface unit and converts into digital form.
- Received signal is classified into 23 classes according to mass and length of yarn fault.
- Calculating mean value during mean value process.
- Sends cut signal to the signal unit through interface unit for yarn faults set as cut mode by user.
- Sending signal to activate the electrical and textile alarms in the interface unit.
- Provide yarn-running status to Front End Software.

DIGITAL SIGNAL PROCESSOR

CHAPTER – 2

DIGITAL SIGNAL PROCESSOR

2.1 WHAT IS DSP?

Digital Signal Processing is a method of processing real world signals (represented by a sequence of numbers) using mathematical techniques to perform transformations or extract information.

2.2 DIGITAL SIGNAL PROCESSOR

A digital signal processor (DSP) is a type of microprocessor - one that is incredibly fast and powerful. A DSP is unique because it processes data in real time. This real-time capability makes a DSP perfect for applications that cannot tolerate any delays. Here are just some of the advantages of designing with DSPs over other microprocessors:

- Single-cycle multiply-accumulate operations.
- Real-time performance, simulation, and emulation.
- Flexibility.
- Reliability.
- Increased system performance.
- Reduced system cost.

2.3 TMS320 FAMILY

The TMS320 family consists of fixed-point, floating-point, and multiprocessor digital signal processors (DSPs). TMS320 DSPs have an architecture designed specifically for real-time signal processing.

2.3.1 TMS320C20x GENERATION

Texas Instruments uses static CMOS integrated-circuit technology to fabricate the TMS320C20x DSPs. The architectural design of the 'C20x is based on that of the 'C5x. The operational flexibility and speed of the 'C20x and 'C5x are a result of an advanced, modified Harvard architecture (which has separate buses for program and data memory), a multilevel pipeline, on-chip peripherals, on-chip memory, and a highly specialized instruction set. The 'C20x per-forms up to 40 MIPS (million instructions per second).

The 'C20x generation offers the following benefits:

- Enhanced TMS320 architectural design for increased performance and versatility.
- Modular architectural design for fast development of additional spin-off devices.
- Advanced IC processing technology for increased performance.
- Fast and easy performance upgrades for 'C1x and 'C2x source code, which is upward compatible with 'C20x source code.
- Enhanced instruction set for faster algorithms and for optimized high-level language operation.
- New static design techniques for minimizing power consumption.

2.3.2 KEY FEATURES OF THE TMS320C20x

Key features on the various 'C20x devices are:

SPEED:

- 50-, 35-, or 25-ns execution time of a single-cycle instruction.
- 20, 28.5, or 40 MIPS.

CODE COMPATIBILITY:

- Code compatibility with other TMS320 fixed-point devices.
- Source-code compatible with all 'C1x and 'C2x devices.
- Upward compatible with the 'C5x devices.

MEMORY:

- 224K words of addressable memory space (64K words of program space, 64K words of data space, 64K words of I/O space, and 32K words of global space).
- 544 words of dual-access on-chip RAM (288 words for data and 256 words for program/data).

CPU:

- 32-bit arithmetic logic unit (CALU).
- 32-bit accumulator.
- 16-bit × 16-bit parallel multiplier with 32-bit product capability.
- Three scaling shifters.
- Eight 16-bit auxiliary registers with a dedicated arithmetic unit for indirect addressing of data memory.

PROGRAM CONTROL:

- 4-level pipeline operation.
- 8-level hardware stack.
- User-maskable interrupt lines.

INSTRUCTION SET:

- Single-instruction repeat operation.
- Single-cycle multiply/accumulate instructions.
- Memory block move instructions for better program/data management.
- Indexed-addressing capability.
- Bit-reversed indexed-addressing capability for radix-2 FFTs.

ON-CHIP PERIPHERALS:

- Software-programmable timer.
- Software-programmable wait-state generator for program, data, and I/O memory spaces.
- Oscillator and phase-locked loop (PLL) to implement clock options.
- ×1, ×2, ×4, and ÷2 (only ×2 and ÷2 available on 'C209).
- CLK register for turning the CLKOUT1 pin on and off.
- Synchronous serial port (not available on 'C209).
- Asynchronous serial port (not available on 'C209).
- On-chip scanning-logic circuitry for emulation and testing purposes.

POWER:

- 5- or 3.3-V static CMOS technology.
- Power-down mode to reduce power consumption

HARDWARE UNIT

CHAPTER – 3

HARDWARE UNIT

3.1 HARDWARE UNIT

The hardware unit is used to generate the measuring head signals. A slot is provided on the topside of the unit for fixing the DSP card, which is to be tested. A wide range of resistors were selected to give a wide range voltages .The 37 pin connector is the interface between the hardware unit and the DSP card. The channels S1-S12 in the connector are used as the input channels. The channels CUT1-CUT12 are used as the output channels and are connected to Light Emitting Diodes to indicate the test results. The channels AL_TEX and AL_ELE are used for alarm activation. The INT2-EXT is connected to a push button, which acts as the START button for the testing process.

PIN NUMBER	SIGNAL NUMBER	R1(ohms)	R2(ohms)	VOLTAGE(volts) Theortical value
2	S1	10	10	2.5
4	S2	10	12	2.27
6	S3	10	16.3	1.901
8	S4	10	19.95	1.66
10	S5	10	26.97	1.35
12	S6	10	30	1.25
21	S7	10	33.6	1.146
23	S8	10	37.3	1.057
25	S9	10	50.2	0.83
27	S10	10	89	0.505
29	S11	10	110.8	0.4166
31	S12	10	150.1	0.3125

HARDWARE UNIT

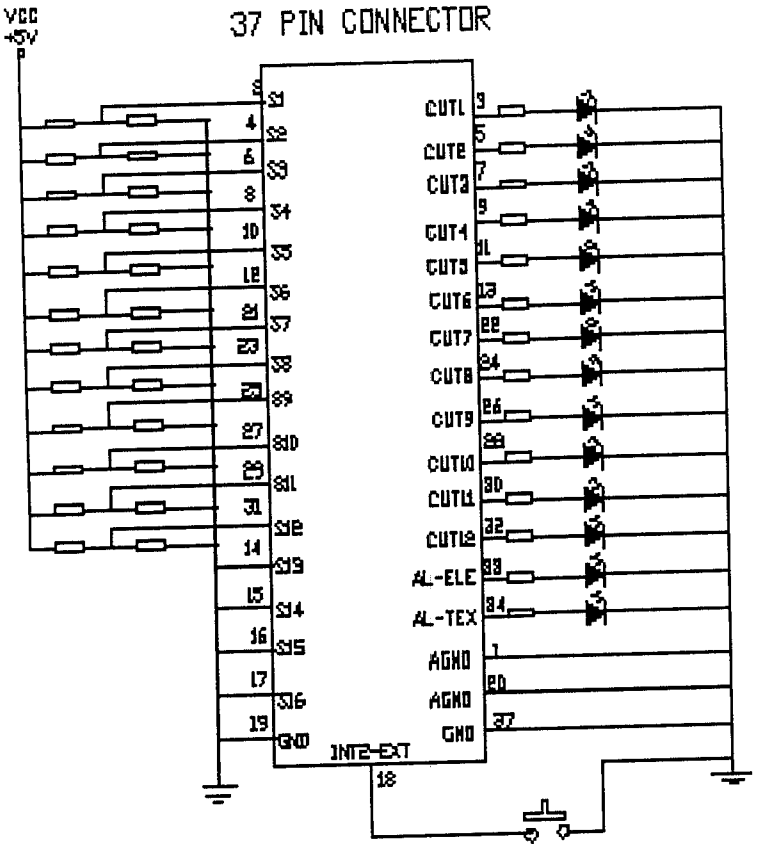


Fig 3

3.2 POWER SUPPLY

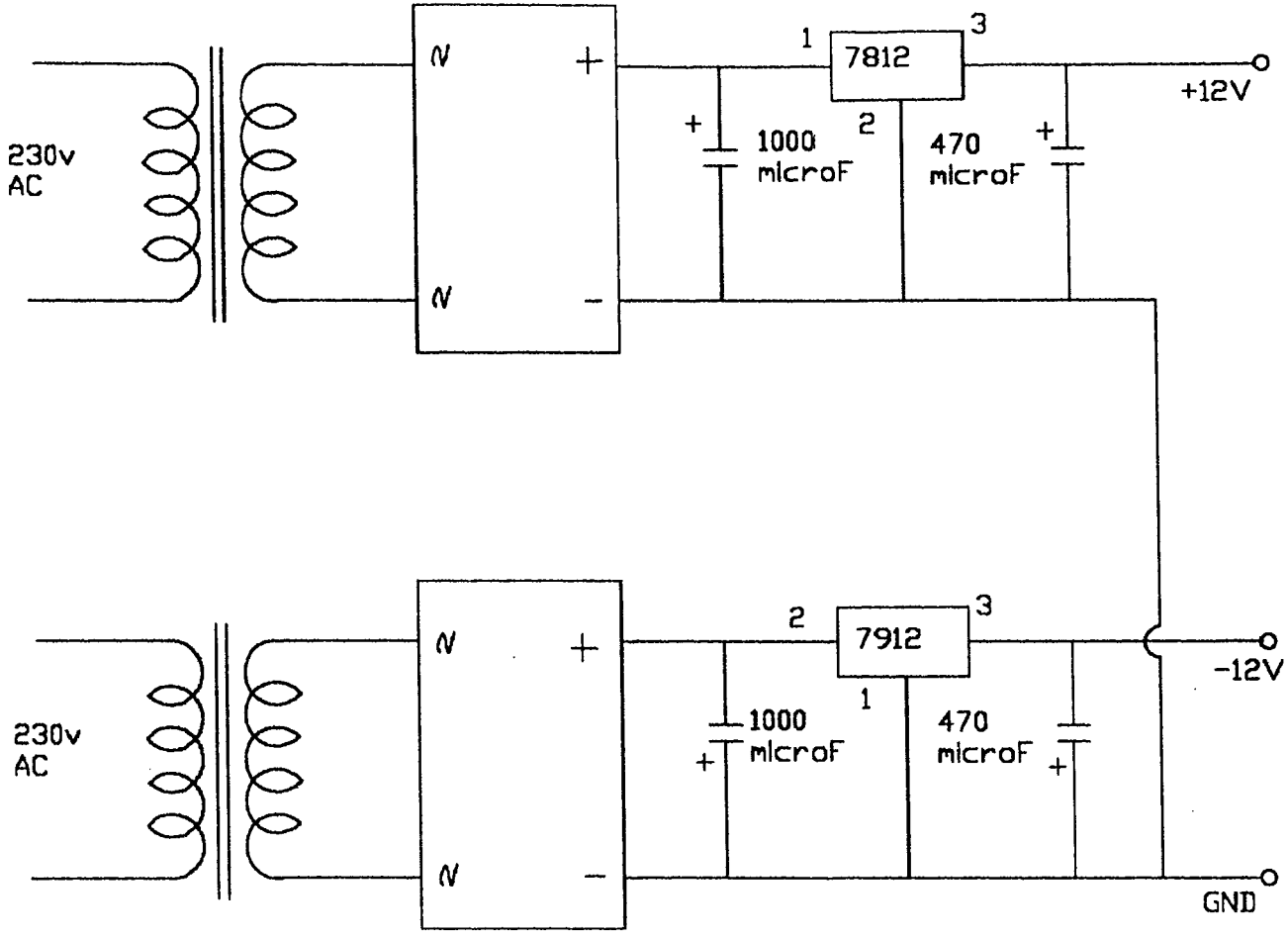
The power supply circuit is used to provide a regulated free +5V, +12V and –12V supply for the entire circuit. It consists of a step down transformer, bridge rectifier, regulating IC's and capacitor filters.

The transformer steps down the supply voltage from 230 V A.C to 9V and 15V. The output from the transformer is rectified using a bridge rectifier and is fed to the regulating IC's 7805, 7812, 7912.

The capacitors are used before and after the regulating IC's in order to reduce the ripples and oscillations. The diodes are employed to provide a fly back path for the discharge of excess capacitance available at the output terminals of the regulating IC's.

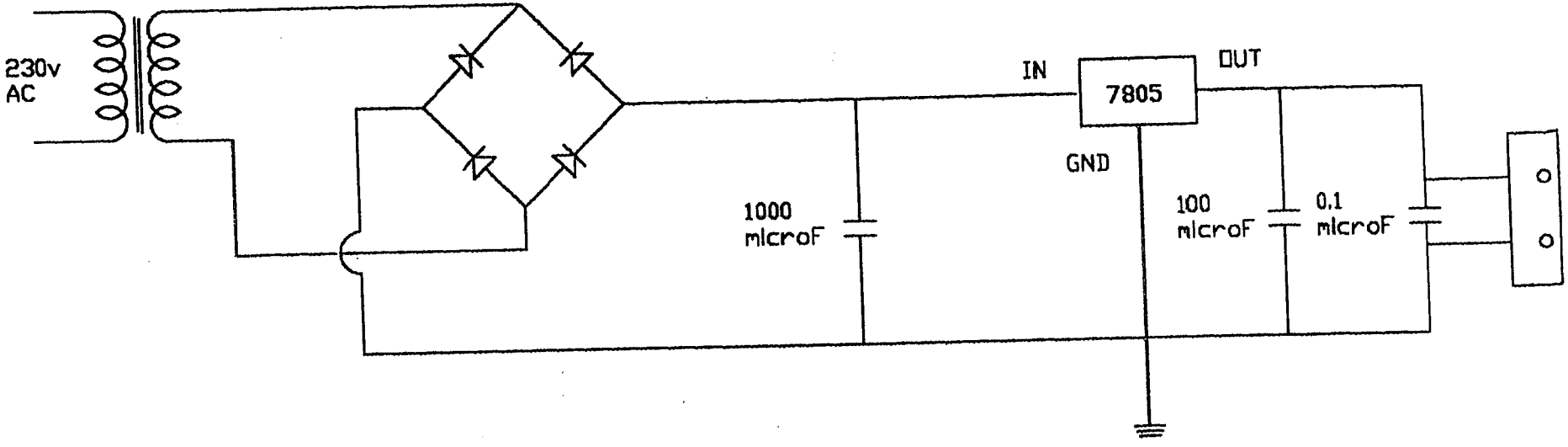
Then the output from the power supply circuit is a constant and regulating DC voltage of +5V, -12V and +12V .

P-708
LIBRARY
641 006



POWER SUPPLY CIRCUIT

Fig. 4



Power Supply Circuit

Fig. 5

TESTING PROCESS PROCEDURE

Step 1: Insert the DSP card to be tested into the slot on the test unit.

Step 2: Switch on the power supply.

Step 3: Push the start button.

Step 4: First 6 LEDs and the last 2 alarm LEDs **BLINK 5 TIMES** to indicate the **SYSTEM STARTUP .**

Step 5: First 6 LEDs glows together for few seconds to show **TEST START INDICATION** and the **START OF DPRAM CHECKING.**

Step 6: **TEST 1 – DPRAM CHECKING.**

* First two LEDs glow together for few seconds indicating that the **DPRAM1 is O.K.**
* First two LEDs blink together indicating that the **DPRAM1 is NOT O.K.**
* LEDs 3 and 4 glows together for few seconds indicating that the **DPRAM2 is O.K.**
* LEDs 3 and 4 blink together indicating that the **DPRAM2 is NOT O.K.**
* LEDs 5 and 6 glows together for few seconds indicating that the **DPRAM3 is O.K.**
* LEDs 5 and 6 blink together indicating that the **DPRAM3 is NOT O.K.**

Step 7: First 6 LEDs glows together for few seconds to indicate that the **DPRAM CHECKING is OVER and MULTIPLEXER CHECKING STARTS.**

Step 8: **TEST 2 – MULTIPLEXER CHECKING.**

LED 1 glows for few seconds indicating that the **CHANNEL 1 O.K.** else it blinks to indicate that the **CHANNEL 1 is NOT O.K.**

Above procedure is repeated for next 2 to 6 channels.

Step 9: First 6 LEDs glows together for few seconds to indicate that the **CHECKING FOR CHANNELS 1 TO 6 is OVER** and **CHECKING FOR CHANNELS 7 TO 12 STARTS.**

Step 10: LED 1 glows for few seconds to indicate that the **CHANNEL 7 is O.K.** else it blinks to indicate that the **CHANNEL 7 is NOT O.K.**

Above procedure repeated for next 8 to 12 channels.

Step 11: First 6 LEDs glows together for few seconds to indicate that the **CHECKING FOR MULTIPLEXER is OVER and the CHECKING FOR ADC STARTS.**

Step 12: **TEST 3 – ADC CHECKING.**

LED1 glows for few seconds to indicate that the **CHANNEL 1 is O.K.** and it blinks to indicate that the **CHANNEL 1 is NOT O.K.**

Above procedure repeated for next 2 to 6 channels.

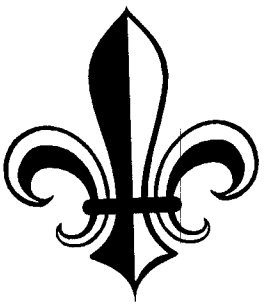
Step 13: First 6 LEDs glows together for few seconds to indicate that the **CHECKING FOR CHANNELS 1 TO 6 is OVER** and **CHECKING FOR CHANNELS 7 TO 12 STARTS.**

Step 14:. LED7 glows for few seconds to indicate that the **CHANNEL 7 is O.K.** and it blinks to indicate that the **CHANNEL 7 is NOT O.K.**

Above procedure repeated for next 7 to 12 channels.

Step 15: First 6 LEDs and last 2 alarm LEDs **BLINKS CONTINUOUSLY** to indicate that the **TESTING PROCESS is COMPLETE.**

Step 16: Switch off power supply.

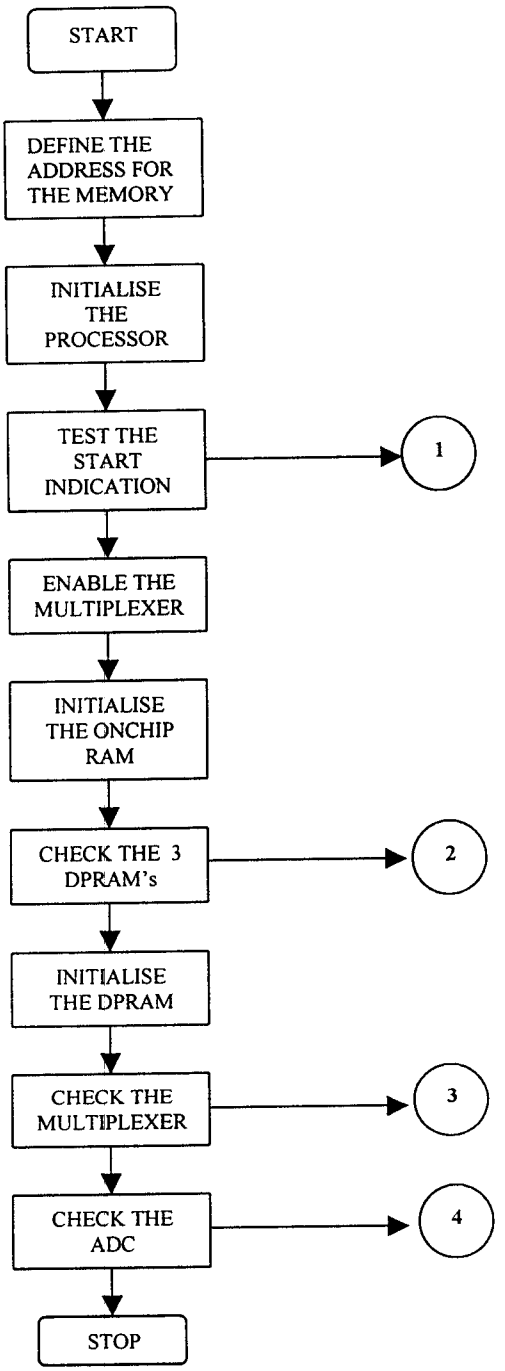


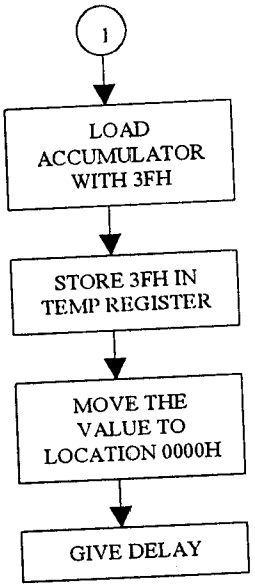
SOFTWARE ASM

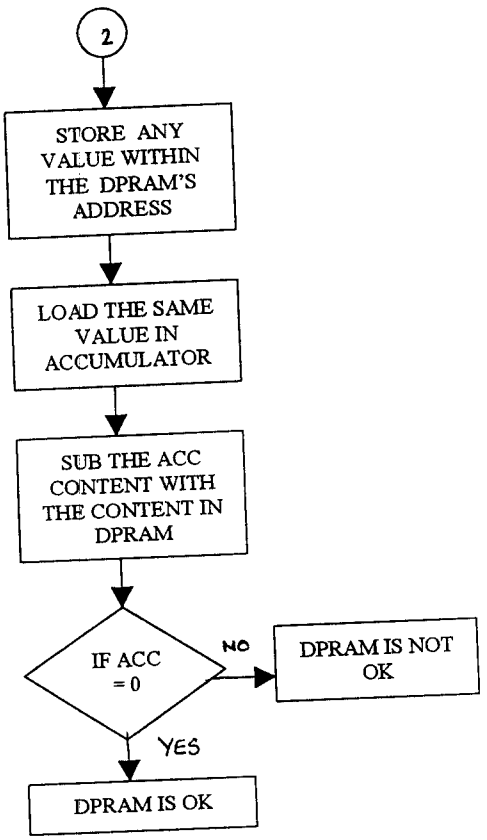
PROGRAMMING

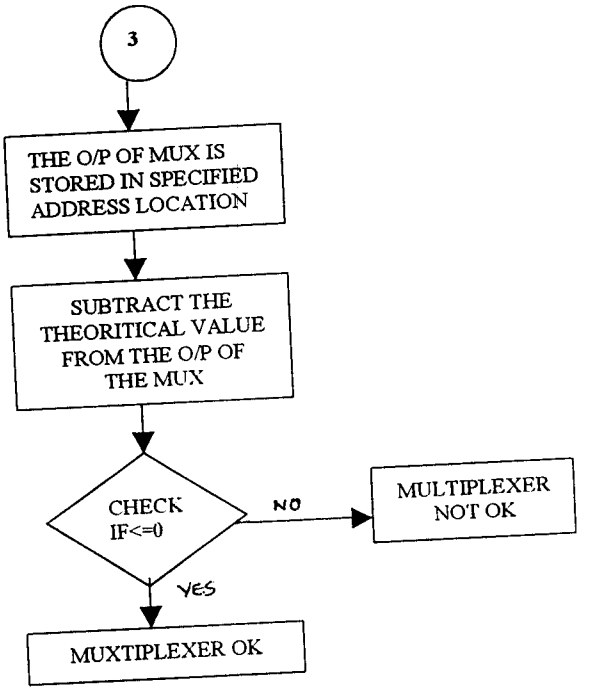
CHAPTER - 4

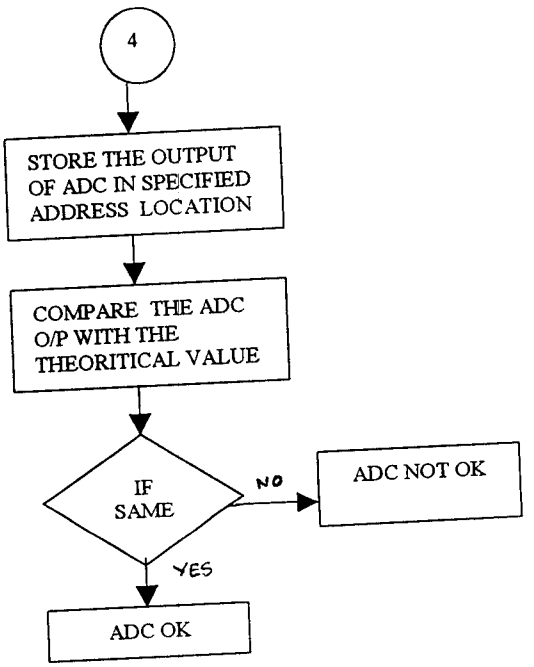
4 .1 FLOW CHART




1
LOAD ACCUMULATOR WITH 3FH
STORE 3FH IN TEMP REGISTER
MOVE THE VALUE TO LOCATION 0000H
GIVE DELAY


2
STORE ANY VALUE WITHIN THE DPRAM'S ADDRESS
LOAD THE SAME VALUE IN ACCUMULATOR
SUB THE ACC CONTENT WITH THE CONTENT IN DPRAM
IF ACC = 0
NO
DPRAM IS NOT OK
YES
DPRAM IS OK


3
THE O/P OF MUX IS STORED IN SPECIFIED ADDRESS LOCATION
SUBTRACT THE THEORITICAL VALUE FROM THE O/P OF THE MUX
CHECK IF<=0
NO
MULTIPLEXER NOT OK
YES
MUXTIPLEXER OK


4
STORE THE OUTPUT OF ADC IN SPECIFIED ADDRESS LOCATION
COMPARE THE ADC O/P WITH THE THEORITICAL VALUE
IF SAME
NO
ADC NOT OK
YES
ADC OK

4.2 SOFTWARE ASM PROGRAMMING

The test software ASM coding is written and checked using TMS320C203 simulator and then it is loaded into EPROM's and fixed on the DSP card at appropriate position.

```
*****************************************************************
```

FILE NAME: TEST.ASM

FUNCTION: TO CHECK THE DSP CARD

```
*****************************************************************

CLK     EQU   0FFE8H      ; CLK REGISTER
ICR     EQU   0FFECH      ; INTERRUPT CONTROL REGISTER
IMR     EQU   4           ; INTERRUPT MASK REGISTER
IFR     EQU   6           ; INTERRUPT FLAG REGISTER
WSGR    EQU   0FFFCH      ; WAIT-STATE GENERATOR REGISTER
TCR     EQU   0FFF8H      ; TIMER CONTROL REGISTER
PRD     EQU   0FFF9H      ; TIMER PERIOD REGISTER
TIM     EQU   0FFFAH      ; TIMER COUNTER REGISTER
```

DIRECT MEMORY ADDRESSING DEFINED IN PAGE '0'

****

```
AD_DATA     EQU   60H       ;ADC_DATA
IFF         EQU   61H       ; TO CONV. 16-BIT TO 12-BIT DATA
TWELVE      EQU   62H       ; TWELVE CHANNEL COUNTER
NEX_CS      EQU   63H       ;POINTER FOR ON-CHIP MEMORY
I300        EQU   64H       ;TEMP POINTER FOR ON-CHIP MEMORY
I2000       EQU   65H       ;TEMP POINTER FOR OFF-CHIP MEMORY
NEX_FC      EQU   66H       ;POINTER FOR OFF-CHIP MEMORY
TMP         EQU   67H       ;TEMP
TMP1        EQU   68H       ;TEMP1
COUNT       EQU   69H       ;COUNTER

    B     START
```

PROCESSOR INITIALIZATION

****

```
START:
    SETC   INTM         ;MASK ALL INTERRUPTS
    LDP    #0           ;LOAD DATA PAGE POINTER = 0
    LACC   #0E00H       ;LOAD ACC = 0E00H
    SACL   TMP          ;STORE IN TEMP
    LST    #0,TMP       ;STATUS REGISTER ( ST0 )INIT.
    ZAC                 ;LOAD ACC = 0
    SACL   IMR          ;DISABLE ALL INTERRUPTS
    RSXM                ;RESET SIGN EXTENDED MODE
```

```
    SPLK   #0,IFR          ;CLEAR ALL PENDING INTERRUPTS
    LAR    AR2,#04H        ;AR2 = 05H
```

SYSTEM STARTUP INDICATION

```
**********************************
STARTUP:
    LACC   #303FH          ;LOAD ACC = 303FH ( 11000000111111b )
    SACL   TMP             ;STORE IN TEMP
    OUT    TMP,0           ;OUT TEMP @ 0000H
    CALL   DELAY1          ;DELAY FOR
    ZAC                    ;LOAD ACC =  0
    SACL   TMP             ;STORE IN TEMP
    OUT    TMP,0           ;OUT TEMP @ 0000H
    CALL   DELAY1          ;DELAY
    MAR    *,AR2           ;ARP = AR2
    BANZ   STARTUP         ;BRANCH TO STARTUP IF AR2 <> 0
    ZAC                    ;LOAD ACC =  0
    SACL   TMP             ;STORE IN TEMP
    OUT    TMP,0           ;OUT TEMP @ 0000H
    CALL   DELAY           ;DELAY
```

TEST START INDICATION

```
************************
TSTART:
    LACC   #03FH           ;LOAD ACC = 3FH  ( 00000000111111b )
    SACL   TMP             ;STORE IN TEMP
    OUT    TMP,0           ;OUT TEMP @ 0000H
```

```
CALL   DELAY          ;DELAY
ZAC                   ;LOAD ACC = 0
SACL   TMP            ;STORE IN TEMP
OUT    TMP,0          ;OUT TEMP @ 0000H
CALL   DELAY          ;DELAY
```

ENABLE THE MUX & GIVE SOC

****

```
ENMUX:
     LACL   #10H          ;LOAD ACC = 10H
     SACL   TMP           ;STORE IN TEMP
     OUT    TMP,1000H     ;MUX ENABLE
     RPT    #90           ;DELAY FOR 3.64 MICRO SEC
     NOP
     OUT    TMP,3000H     ;SOC FOR CHANNEL '0'
```

INITIALIZE ON-CHIP RAM

****

```
INIT_ONCHIPRAM:
     ZAC                  ;LOAD ACC = 0
     LAR    AR0,#200H     ;ON-CHIP MEMORY ( B0,B1 )
     MAR    *,AR0         ;ARP = AR0
     RPT    #0FFH         ;CLEAR MEM. LOC ( 200H - 2FFH )
     SACL   *+
     MAR    *,AR0         ;ARP = AR0
     RPT    #0FFH         ;CLEAR MEM. LOC ( 300H - 3FFH)
     SACL   *+
```

```
    LAR    AR2,#07H          ;AR2 = 7H
```

DPRAM CHECK ROUTINE (WRITE TO DPRAM)

```
*****************************************
CHK_DPRAM:
WR_DPRAM1:
     LAR    AR0,#1000H       ;AR0 = 1000H
     MAR    *,AR0            ;ARP = AR0
     LACL   #055H            ;LOAD ACC = 055H
     SACL   *+               ;STORE IN MEM. LOC ( 1000H - 17FFH )
     MAR    *,AR2            ;ARP = AR2
     BANZ   WR_DPRAM1        ;BRANCH TO WR_DPRAM1 IF AR2 <> 0
     LAR    AR2,#07H         ;AR2 = 07H

 WR_DPRAM2:
     LAR    AR0, #2000H      ; AR0 = 2000H
     MAR    *, AR0           ; ARP = AR0
     LACL   #055H            ; LOAD ACC = 055H
     SACL   *+               ; STORE IN MEM. LOC (2000H - 27FFH)
     MAR    *, AR2           ; ARP = AR2
     BANZ   WR_DPRAM2        ; BRANCH TO WR_DPRAM2 IF AR2 <> 0
     LAR    AR2, #07H        ; AR2 = 07H
 WR_DPRAM3:
     LAR    AR0, #3000H      ; AR0 = 3000H
     MAR    *, AR0           ; ARP = AR0
     LACL   #055H            ; LOAD ACC = 055H
     SACL   *+               ; STORE IN MEM. LOC (3000H - 37FFH)
```

```
    MAR    *, AR2              ; ARP = AR2
    BANZ   WR_DPRAM3           ; BRANCH TO WR_DPRAM3 IF AR2 <> 0
    LAR    AR2, #07FFH         ;AR2 = 07FFH
```

DPRAM CHECK ROUTINE (READ FROM DPRAM AND CHECK)

```
**********************************************************
RD_DPRAM:
    LAR    AR0, #1000H         ; AR0 = 1000H ( DPRAM 1 )
RD_DPRAM1:
    MAR    *, AR0              ; ARP = AR0
    LACL   *                   ; LOAD ACC = DATA @ AR0
    SUB    #055H               ; SUB 055H
    BCND   DPR1_NOK, NEQ       ; BRANCH TO DPR1_NOK IF ACC <> 0
    MAR    *, AR0              ; ARP = AR0
    ADD    #01H                ; ADD 1H (NEXT ADDR.)
    MAR    *, AR2              ; ARP = AR2
    BANZ   RD_DPRAM1           ; BRANCH TO RD_DPRAM1 IF AR2 <> 0
    LACL   #03H                ; LOAD ACC = 03H (00000000000011b)
    SACL   TMP                 ; STORE IN TEMP
    OUT    TMP, 0              ; OUT TEMP @ 0000H
    CALL   DELAY               ; DELAY
    LAR    AR0, #2000H         ; AR0 = 2000H ( DPRAM 2 )
    B      RD_DPRAM2           ; BRANCH TO RD_DPRAM2
DPR1_NOK:
    LACL   #03H                ; LOAD ACC = 03H (00000000000011b)
    SACL   TMP                 ; STORE IN TEMP
    OUT    TMP, 0              ; OUT TEMP @ 0000H
```

```
    CALL   DELAY1                ; DELAY
    ZAC                          ; LOAD ACC = 0
    SACL   TMP                   ; STORE IN TEMP
    OUT    TMP, 0                ; OUT TEMP @ 0000H
    CALL   DELAY1                ; DELAY
    B      DPR1_NOK              ; BRANCH TO DPR1_NOK
RD_DPRAM2:
    MAR    *, AR0                ; ARP = AR0
    LACL   *                     ; LOAD ACC = DATA @ AR0
    SUB    #055H                 ; SUB 055H
    BCND   DPR2_NOK, NEQ         ; BRANCH TO DPR2_NOK IF ACC <> 0
    MAR    *, AR0                ; ARP = AR0
    ADD    #01H                  ; ADD 1H (NEXT ADDR.)
    MAR    *, AR2                ; ARP = AR2
    BANZ   RD_DPRAM2             ; BRANCH TO RD_DPRAM2 IF AR2 <> 0
    LACL   #0CH                  ; LOAD ACC = OCH (00000000001100b)
    SACL   TMP                   ;STORE IN TEMP
    OUT    TMP,0                 ;OUT TEMP @ 0000H
    CALL   DELAY                 ;DELAY
    LAR    AR0,#3000H            ;AR0 = 3000H ( DPRAM 3 )
    B      RD_DPRAM3
DPR2_NOK:
    LACL   #0CH                  ;LOAD ACC = OCH ( 00000000001100b)
    SACL   TMP                   ;STORE IN TEMP
    OUT    TMP,0                 ;OUT TEMP @ 0000H
    CALL   DELAY1                ;DELAY
    ZAC                          ;LOAD ACC = 0
```

```
    SACL   TMP              ;STORE IN TEMP
    OUT    TMP,0            ;OUT TEMP @ 0000H
    CALL   DELAY1                 ;DELAY
    B      DPR2_NOK         ;BRANCH TO DPR2_NOK
RD_DPRAM3:
    MAR    *,AR0            ;ARP = AR0
    LACL   *                ;LOAD ACC = DATA @ AR0
    SUB    #055H            ;SUB 055H
    BCND   DPR3_NOK,NEQ     ;BRANCH TO DPR3_NOK IF ACC <> 0
    MAR    *,AR0            ;ARP = AR0
    ADD    #01H             ;ADD 1H (NEXT ADDR.)
    MAR    *,AR2            ;ARP = AR2
    BANZ   RD_DPRAM3        ;BRANCH TO RD_DPRAM3
    LACL   #030H            ;LOAD ACC = 030H ( 00000000110000b )
    SACL   TMP              ;STORE IN TEMP
    OUT    TMP,0            ;OUT TEMP @ 0000H
    CALL   DELAY            ;DELAY
    LAR    AR2,#07H         ;AR2 = 07H
    B      INIT_DPRAM       ;BRANCH TO INIT_DPRAM
DPR3_NOK:
    LACL   #030H            ;LOAD ACC = 030H ( 00000000110000b )
    SACL   TMP              ;STORE IN TEMP
    OUT    TMP,0            ;OUT TEMP @ 0000H
    CALL   DELAY1                 ;DELAY
    ZAC                     ;LOAD ACC = 0
    SACL   TMP              ;STORE IN TEMP
    OUT    TMP,0            ;OUT TEMP @ 0000H
```

```
    CALL   DELAY1                ;DELAY
    B      DPR3_NOK              ;BRANCH TO DPR3_NOK
```

INITIALIZE DPRAM

```
**********
INIT_DPRAM:
        ZAC                      ;LOAD ACC = 0
        SACL  TMP                ;STORE IN TEMP
        OUT   TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY               ;DELAY
        LACL #03FH               ;LOAD ACC = 03FH ( 00000000111111b )
        SACL  TMP                ;STORE IN TEMP
        OUT   TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY               ;DELAY
 INIT_DPRAM1:
      ZAC                        ;LOAD ACC = 0
      LAR    AR0,#1000H          ;AR0 = 1000H
        MAR    *,AR0             ;ARP = AR0
      RPT    #0FFH               ;CLEAR MEM. LOC ( 1000H - 17FFH )
      SACL   *+
      MAR    *,AR2               ;ARP = AR2
      BANZ   INIT_DPRAM1         ;BRANCH TO INIT_DPRAM1 IF AR2 <> 0
        LAR    AR2,#02H          ;AR2 = 02H
  INIT_DPRAM2:
      ZAC                        ;LOAD ACC = 0
      LAR    AR0,#2000H          ;AR0 = 2000H
        MAR    *,AR0             ;ARP = AR0
```

```
    RPT   #0FFH              ;CLEAR MEM. LOC ( 2000H - 27FFH )
    SACL  *+
    MAR   *,AR2              ;ARP = AR2
    BANZ  INIT_DPRAM2        ;BRANCH TO INIT_DPRAM2 IF AR2 <> 0
      LAR   AR2,#02H         ;AR2 = 02H
INIT_DPRAM3:
    ZAC                      ;LOAD ACC = 0
    LAR   AR0,#3000H         ;AR0 = 3000H
      MAR   *,AR0            ;ARP = AR0
    RPT   #0FFH              ;CLEAR MEM. LOC (3000H - 37FFH )
    SACL  *+
    MAR   *,AR2              ;ARP = AR2
      BANZ INIT_DPRAM3       ;BRANCH TO INIT_DPRAM3 IF AR2 <> 0
      ZAC                    ;LOAD ACC = 0
      SACL TMP               ;STORE IN TEMP
      OUT  TMP,0             ;OUT TEMP @ 0000H
      CALL DELAY             ;DELAY
    LACC  #0FFFH             ;LOAD ACC = 0FFFH
      SACL  IFF              ;STORE IN IFF
      LACC  #300H            ;LOAD ACC = 300H
      SACL  I300             ;STORE IN I300
    LACC  #2000H             ;LOAD ACC = 2000H
    SACL  I2000              ;STORE IN I2000
      LACC #0FFFH            ;LOAD ACC = 0FFFH
      SACL COUNT             ;STORE IN COUNT
```

MULTIPLEXER CHECK ROUTINE

****

```
MAIN1:
    LACL    COUNT           ;LOAD ACC = DATA @ COUNT
    SUB     #01H            ;SUB 1H
    SACL    COUNT           ;STORE IN COUNT
    BCND    PROCESS1,EQ     ;BRANCH TO PROCESS IF ACC = 0
    SPLK    #0,TWELVE       ;STORE TWELVE = 0
OKAY1:
    LAR     AR6,I2000       ;AR0 = I2000 ( 2000H )
    LACC    I2000           ;LOAD ACC = I2000
    SACL    NEX_FC          ;STORE IN NEX_FC ( POINTER )
    LAR     AR7,I300        ;AR0 = I300 ( 300H )
    LACC    I300            ;LOAD ACC = I300
    SACL    NEX_CS          ;STORE IN NEX_CS ( POINTER )

CHA_SEL1:
    LACL    TWELVE          ;LOAD ACC = DATA @ TWELVE
    ADD     #1              ;ADD 1H
    SACL    TWELVE          ;STORE IN TWELVE
    SUB     #0CH            ;SUB 0CH ( 12 - CHANNEL )
    BCND    SELECT1,NEQ     ;BRANCH TO SELECT1 IF ACC <> 0
    SPLK    #0,TWELVE       ;STORE TWELCE = 0
SELECT1:
    OUT     TWELVE,1000H    ;OUT TWELVE @ 1000H ( CH '0'- MUX ADDR. )
    LACL    TWELVE          ;LOAD ACC = TWELVE
    ADD     #10H            ;ADD 10H
```

```
    SACL   TMP              ;STORE IN TEMP
    OUT    TMP,1000H        ;OUT TEMP @ 1000H (MUX ADDR.)
    RPT    #100             ;DELAY FOR 4.04 MICRO SEC
    NOP
CONT1: OUT   TMP,3000H      ;OUT TEMP @ 3000H ( ADC ADDR. )
    CALL   CAPTURE1
    LACL   NEX_CS           ;POINT TO NEXT CHANNEL
    ADD    #20H             ;ADD 20H
    SACL   NEX_CS           ;STORE IN NEX_CS
    LAR    AR7,NEX_CS       ;AR7 = NEX_CS
    LACL   NEX_FC           ;POINT TO NEXT CHANNEL
    ADD    #01H             ;ADD 01H
    SACL   NEX_FC           ;STORE IN NEX_FC
    LAR    AR6,NEX_FC       ;AR6 = NEX_FC
    LACL   TWELVE           ;CHK. FOR ZEROth CHANNEL
    BCND   MAIN1,EQ         ;BRANCH TO MAIN1 IF ACC = 0
    B      CHA_SEL1
CAPTURE1:
      MAR    *,AR6          ;ARP = AR6
    IN     AD_DATA,3000H    ;GET DATA FROM ADC
      LACL   AD_DATA        ;LOAD ACC = AD_DATA
      AND    IFF            ;16BIT TO 12BIT
      SACL  AD_DATA         ;STORE IN AD_DATA
    LACL   AD_DATA          ;LOAD ACC = AD_DATA
      SACL  *               ;STORE IN AR6
      RET
PROCESS1:
```

```
MCH_1:
        LAR   AR0,#2000H        ;AR0 = 2000H
        MAR   *,AR0             ;ARP = AR0
        LACL  *                 ;LOAD ACC = AR0
      SUB   #038EH              ;SUB 038EH
      BCND  MCH1_NOK,LEQ              ;BRANCH TO MCH1_NOK IF ACC <= 0
        LACL  #01H              ;LOAD ACC = 01H ( 00000000000001b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL  DELAY                     ;DELAY
      B     MCH_2               ;BRANCH TO MCH_2
MCH1_NOK:
        LACL  #01H              ;LOAD ACC = 01H ( 00000000000001b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL  DELAY1                    ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL  DELAY1                    ;DELAY
      B     MCH1_NOK            ;BRANH TO MCH1_NOK
MCH_2:
        LAR   AR0,#2001H        ;AR0 = 2001H
        MAR   *,AR0             ;ARP = AR0
        LACL  *                 ;LOAD ACC = AR0
      SUB   #032EH              ;SUB 032EH
      BCND  MCH2_NOK,LEQ              ;BRANCH TO MCH2_NOK IF ACC <= 0
```

```
      LACL #02H              ;LOAD ACC = 02H ( 00000000000010b )
      SACL TMP               ;STORE IN TEMP
      OUT  TMP,0             ;OUT TEMP @ 0000H
      CALL DELAY                   ;DELAY
    B    MCH_3               ;BRANCH TO MCH_3
MCH2_NOK:
      LACL #02H              ;LOAD ACC = 02H ( 00000000000010b )
      SACL TMP               ;STORE IN TEMP
      OUT  TMP,0             ;OUT TEMP @ 0000H
      CALL DELAY1                  ;DELAY
      ZAC                    ;LOAD ACC = 0
      SACL TMP               ;STORE IN TEMP
      OUT  TMP,0             ;OUT TEMP @ 0000H
      CALL DELAY1                  ;DELAY
    B    MCH2_NOK            ;BRANH TO MCH2_NOK

 MCH_3:
      LAR  AR0,#2002H        ;AR0 = 2002H
      MAR  *,AR0             ;ARP = AR0
      LACL *                 ;LOAD ACC = AR0
    SUB  #029EH              ;SUB 029EH
    BCND MCH3_NOK,LEQ              ;BRANCH TO MCH3_NOK IF ACC <= 0
      LACL #04H              ;LOAD ACC = 04H ( 00000000000100b )
      SACL TMP               ;STORE IN TEMP
      OUT  TMP,0             ;OUT TEMP @ 0000H
      CALL DELAY                   ;DELAY
    B    MCH_4               ;BRANCH TO MCH_4
```

```
MCH3_NOK:
        LACL #04H               ;LOAD ACC = 04H ( 00000000000100b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
    B    MCH3_NOK               ;BRANH TO MCH3_NOK
MCH_4:
        LAR  AR0,#2003H         ;AR0 = 2003H
        MAR  *,AR0              ;ARP = AR0
        LACL *                  ;LOAD ACC = AR0
    SUB   #023FH                ;SUB 023FH
    BCND  MCH4_NOK,LEQ                  ;BRANCH TO MCH4_NOK IF ACC <= 0
      LACL #08H                 ;LOAD ACC = 08H ( 00000000001000b )
      SACL TMP                  ;STORE IN TEMP
      OUT  TMP,0                ;OUT TEMP @ 0000H
      CALL DELAY                        ;DELAY
    B    MCH_5                  ;BRANCH TO MCH_5
MCH4_NOK:
        LACL #08H               ;LOAD ACC = 08H ( 00000000001000b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
```

```
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL  DELAY1                    ;DELAY
    B     MCH4_NOK              ;BRANH TO MCH4_NOK
MCH_5:
        LAR   AR0,#2004H        ;AR0 = 2004H
        MAR   *,AR0             ;ARP = AR0
        LACL  *                 ;LOAD ACC = AR0
    SUB   #01BFH                ;SUB 01BFH
    BCND  MCH5_NOK,LEQ              ;BRANCH TO MCH5_NOK IF ACC <= 0
        LACL  #010H             ;LOAD ACC = 010H ( 00000000010000b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL  DELAY                     ;DELAY
    B     MCH_6                 ;BRANCH TO MCH_6
MCH5_NOK:
        LACL  #010H             ;LOAD ACC = 01H ( 00000000010000b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL  DELAY1                    ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL  DELAY1                    ;DELAY
    B     MCH5_NOK              ;BRANH TO MCH5_NOK
MCH_6:
        LAR   AR0,#2005H        ;AR0 = 2005H
```

```
      MAR  *,AR0              ;ARP = AR0
      LACL *                  ;LOAD ACC = AR0
    SUB   #0194H              ;SUB 0194H
    BCND  MCH6_NOK,LEQ              ;BRANCH TO MCH6_NOK IF ACC <= 0
      LACL #020H              ;LOAD ACC = 020H ( 00000000100000b )
      SACL TMP                ;STORE IN TEMP
      OUT  TMP,0              ;OUT TEMP @ 0000H
      CALL DELAY                    ;DELAY
    B     MCH_7               ;BRANCH TO MCH_7
MCH6_NOK:
      LACL #020H              ;LOAD ACC = 020H ( 00000000100000b )
      SACL TMP                ;STORE IN TEMP
      OUT  TMP,0              ;OUT TEMP @ 0000H
      CALL DELAY1                   ;DELAY
      ZAC                     ;LOAD ACC = 0
      SACL TMP                ;STORE IN TEMP
      OUT  TMP,0              ;OUT TEMP @ 0000H
      CALL DELAY1                   ;DELAY
    B     MCH6_NOK            ;BRANH TO MCH6_NOK
MCH_7:
      LACL #03FH              ;LOAD ACC = 3FH ( 00000000111111b )
      SACL TMP                ;STORE IN TEMP
      OUT  TMP,0              ;OUT TEMP @ 0000H
      CALL DELAY                    ;DELAY
      ZAC                     ;LOAD ACC = 0
      SACL TMP                ;STORE IN TEMP
      OUT  TMP,0              ;OUT TEMP @ 0000H
```

```
        CALL DELAY                  ;DELAY
        LAR   AR0,#2006H            ;AR0 = 2006H
        MAR  *,AR0                  ;ARP = AR0
        LACL *                      ;LOAD ACC = AR0
      SUB   #0175H                  ;SUB 0175H
      BCND  MCH7_NOK,LEQ            ;BRANCH TO MCH7_NOK IF ACC <= 0
        LACL #01H                   ;LOAD ACC = 01H ( 00000000000001b )
        SACL TMP                    ;STORE IN TEMP
        OUT   TMP,0                 ;OUT TEMP @ 0000H
        CALL DELAY                  ;DELAY
      B     MCH_8                   ;BRANCH TO MCH_8
MCH7_NOK:
        LACL #01H                   ;LOAD ACC = 01H ( 00000000000001b )
        SACL TMP                    ;STORE IN TEMP
        OUT   TMP,0                 ;OUT TEMP @ 0000H
        CALL DELAY1                 ;DELAY
        ZAC                         ;LOAD ACC = 0
        SACL TMP                    ;STORE IN TEMP
        OUT   TMP,0                 ;OUT TEMP @ 0000H
        CALL DELAY1                 ;DELAY
      B     MCH7_NOK                ;BRANH TO MCH7_NOK
MCH_8:
        LAR   AR0,#2007H            ;AR0 = 2007H
        MAR  *,AR0                  ;ARP = AR0
        LACL *                      ;LOAD ACC = AR0
      SUB   #0147H                  ;SUB 0147H
      BCND  MCH8_NOK,LEQ            ;BRANCH TO MCH8_NOK IF ACC <= 0
```

```
        LACL #02H               ;LOAD ACC = 02H ( 000000000000010b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY                      ;DELAY
    B    MCH_9                  ;BRANCH TO MCH_9
MCH8_NOK:
        LACL #02H               ;LOAD ACC = 02H ( 000000000000010b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
    B    MCH8_NOK               ;BRANH TO MCH8_NOK
MCH_9:
        LAR  AR0,#2008H         ;AR0 = 2008H
        MAR  *,AR0              ;ARP = AR0
        LACL *                  ;LOAD ACC = AR0
    SUB   #0EDH                 ;SUB 0EDH
    BCND  MCH9_NOK,LEQ              ;BRANCH TO MCH9_NOK IF ACC <= 0
        LACL #04H               ;LOAD ACC = 04H ( 000000000000100b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY                      ;DELAY
    B    MCH_10                 ;BRANCH TO MCH_10
MCH9_NOK:
```

```
        LACL #04H               ;LOAD ACC = 04H ( 00000000000100b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
    B    MCH9_NOK               ;BRANH TO MCH9_NOK
MCH_10:
        LAR  AR0,#2009H         ;AR0 = 2009H
        MAR  *,AR0              ;ARP = AR0
        LACL *                  ;LOAD ACC = AR0
    SUB   #068H                 ;SUB 068H
    BCND  MCH10_NOK,LEQ             ;BRANCH TO MCH10_NOK IF ACC <= 0
        LACL #08H               ;LOAD ACC = 08H ( 00000000001000b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY                      ;DELAY
    B    MCH_11                 ;BRANCH TO MCH_11
MCH10_NOK:
        LACL #08H               ;LOAD ACC = 08H ( 00000000001000b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL TMP                ;STORE IN TEMP
```

```
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
    B     MCH10_NOK             ;BRANH TO MCH10_NOK
MCH_11:
        LAR   AR0,#200AH        ;AR0 = 200AH
        MAR  *,AR0              ;ARP = AR0
        LACL *                  ;LOAD ACC = AR0
    SUB    #045H                ;SUB 045H
    BCND   MCH11_NOK,LEQ            ;BRANCH TO MCH11_NOK IF ACC <= 0
        LACL #010H              ;LOAD ACC = 010H ( 000000000010000b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY                      ;DELAY
    B     MCH_12                ;BRANCH TO MCH_12
MCH11_NOK:
        LACL #010H              ;LOAD ACC = 01H ( 000000000010000b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
    B     MCH11_NOK             ;BRANH TO MCH11_NOK
MCH_12:
        LAR   AR0,#200BH        ;AR0 = 200BH
        MAR  *,AR0              ;ARP = AR0
```

```
    LACL *                  ;LOAD ACC = AR0
    SUB   #01DH             ;SUB 01DH
    BCND  MCH12_NOK,LEQ         ;BRANCH TO MCH12_NOK IF ACC <= 0
    LACL #020H              ;LOAD ACC = 020H ( 00000000100000b )
    SACL  TMP               ;STORE IN TEMP
    OUT   TMP,0             ;OUT TEMP @ 0000H
    CALL DELAY                  ;DELAY
    B     ADC_CHECK         ;BRANCH TO ADC_CHECK
MCH12_NOK:
    LACL #020H              ;LOAD ACC = 020H ( 00000000100000b )
    SACL  TMP               ;STORE IN TEMP
    OUT   TMP,0             ;OUT TEMP @ 0000H
    CALL DELAY1                 ;DELAY
    ZAC                     ;LOAD ACC = 0
    SACL  TMP               ;STORE IN TEMP
    OUT   TMP,0             ;OUT TEMP @ 0000H
    CALL DELAY1                 ;DELAY
    B     MCH12_NOK         ;BRANH TO MCH12_NOK
```

ADC CHECK ROUTINE

****

```
ADC_CHECK:
    ZAC                     ;LOAD ACC = 0
    SACL    TMP             ;STORE IN TEMP
    OUT   TMP,0             ;OUT TEMP @ 0000H
    CALL    DELAY               ;DELAY
    LACL   #03FH            ;LOAD ACC = 03FH ( 00000000111111b )
```

```
        SACL   TMP          ;STORE IN TEMP
        OUT    TMP,0        ;OUT TEMP @ 0000H
        CALL   DELAY        ;DELAY
        ZAC                 ;LOAD ACC = 0
        SACL   TMP          ;STORE IN TEMP
        OUT    TMP,0        ;OUT TEMP @ 0000H
        CALL   DELAY        ;DELAY
        LACC   #0FFFH       ;LOAD ACC = 0FFFH
        SACL   IFF          ;STORE IN IFF
        LACC   #300H        ;LOAD ACC = 300H
        SACL   I300         ;STORE IN I300
        LACC   #2000H       ;LOAD ACC = 2000H
        SACL   I2000        ;STORE IN I2000
        LACC   #0FFFH       ;LOAD ACC = 0FFFH
        SACL   COUNT        ;STORE IN COUNT
MAIN:
        LACL   COUNT        ;LOAD ACC = DATA @ COUNT
        SUB    #01H         ;SUB 1H
        SACL   COUNT        ;STORE IN COUNT
        BCND   PROCESS,EQ   ;BRANCH TO PROCESS IF ACC = 0
        SPLK   #0,TWELVE    ;STORE TWELVE = 0
OKAY:
        LAR    AR6,I2000    ;AR0 = I2000 ( 2000H )
        LACC   I2000        ;LOAD ACC = I2000
        SACL   NEX_FC       ;STORE IN NEX_FC ( POINTER )
        LAR    AR7,I300     ;AR0 = I300 ( 300H )
        LACC   I300         ;LOAD ACC = I300
```

```
      SACL   NEX_CS              ;STORE IN NEX_CS ( POINTER )
CHA_SEL:
      LACL   TWELVE              ;LOAD ACC = DATA @ TWELVE
      ADD    #1                  ;ADD 1H
      SACL   TWELVE              ;STORE IN TWELVE
      SUB    #0CH                ;SUB 0CH ( 12 - CHANNEL )
      BCND   SELECT,NEQ          ;BRANCH TO SELECT IF ACC <> 0
      SPLK   #0,TWELVE           ;STORE TWELCE = 0
SELECT: OUT   TWELVE,1000H       ;OUT TWELVE @ 1000H ( CH '0'- MUX
ADDR. )
      LACL   TWELVE              ;LOAD ACC = TWELVE
      ADD    #10H                ;ADD 10H
      SACL   TMP                 ;STORE IN TEMP
      OUT    TMP,1000H           ;OUT TEMP @ 1000H (MUX ADDR.)
      RPT    #100                ;DELAY FOR 4.04 MICRO SEC
      NOP
CONT:  OUT    TMP,3000H           ;OUT TEMP @ 3000H ( ADC ADDR. )
      CALL   CAPTURE
      LACL   NEX_CS                ;POINT TO NEXT CHANNEL
      ADD    #20H                  ;ADD 20H
      SACL   NEX_CS                ;STORE IN NEX_CS
      LAR    AR7,NEX_CS            ;AR7 = NEX_CS
      LACL   NEX_FC                ;POINT TO NEXT CHANNEL
      ADD    #01H                  ;ADD 01H
      SACL   NEX_FC                ;STORE IN NEX_FC
      LAR    AR6,NEX_FC            ;AR6 = NEX_FC
      LACL   TWELVE                ;CHK. FOR ZEROth CHANNEL
      BCND   MAIN,EQ               ;BRANCH TO MAIN IF ACC = 0
```

```
      B     CHA_SEL
CAPTURE:
      MAR   *,AR6              ;ARP = AR6
      IN    AD_DATA,3000H      ;GET DATA FROM ADC
      LACL  AD_DATA            ;LOAD ACC = AD_DATA
      AND   IFF                ;16BIT TO 12BIT
      SACL   AD_DATA           ;STORE IN AD_DATA
      LACL  AD_DATA            ;LOAD ACC = AD_DATA
      SACL   *                 ;STORE IN AR6
      RET
        PROCESS:
CH_1:
      LAR    AR0,#2000H        ;AR0 = 2000H
      MAR    *,AR0             ;ARP = AR0
      LACL   *                 ;LOAD ACC = DATA @ AR0
      SUB    #03F0H            ;SUB 03F0H
      SUB   #050H              ;SUB 050H
      BCND   CH1_NOK,GEQ       ;BRANCH TO CH1_NOK IF ACC >= 0
      MAR    *,AR0             ;ARP = AR0
      LACL   *                 ;LOAD ACC = DATA @ AR0
      SUB    #03F0H            ;SUB 03F0H
      ADD   #050H              ;SUB 050H
      BCND   CH1_NOK,LEQ       ;BRANCH TO CH1_NOK IF ACC <= 0
      LACL   #01H              ;LOAD ACC = 01H ( 00000000000001b )
     SACL    TMP               ;STORE IN TEMP
     OUT     TMP,0             ;OUT TEMP @ 0000H
      CALL   DELAY                    ;DELAY
```

```
      B CH_2                  ;BRANCH TO CH_2
CH1_NOK:
        LACL #01H             ;LOAD ACC = 01H ( 000000000000001b )
        SACL TMP              ;STORE IN TEMP
        OUT  TMP,0            ;OUT TEMP @ 0000H
        CALL DELAY1                 ;DELAY
        ZAC                   ;LOAD ACC = 0
        SACL TMP              ;STORE IN TEMP
        OUT  TMP,0            ;OUT TEMP @ 0000H
        CALL DELAY1                 ;DELAY
        B    CH1_NOK          ;BRANCH TO CH1_NOK
CH_2:
        LAR  AR0,#2001H       ;AR0 = 2001H
        MAR  *,AR0            ;ARP = AR0
        LACL *                ;LOAD ACC = DATA @ AR0
        SUB  #038EH           ;SUB 038EH
      SUB  #050H              ;SUB 050H
        BCND CH2_NOK,GEQ      ;BRANCH TO CH2_NOK IF ACC >= 0
        MAR  *,AR0            ;ARP = AR0
        LACL *                ;LOAD ACC = DATA @ AR0
        SUB  #038EH           ;SUB 038EH
     ADD  #050H               ;SUB 050H
        BCND CH2_NOK,LEQ      ;BRANCH TO CH2_NOK IF ACC <= 0
        LACL #02H             ;LOAD ACC = 02H ( 000000000000010b )
        SACL TMP              ;STORE IN TEMP
        OUT  TMP,0            ;OUT TEMP @ 0000H
        CALL DELAY                  ;DELAY
```

```
        B       CH_3            ;BRANCH TO CH_3
CH2_NOK:
        LACL #02H               ;LOAD ACC = 02H ( 00000000000010b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        B       CH2_NOK         ;BRANCH TO CH2_NOK
CH_3:
        LAR   AR0,#2002H        ;AR0 = 2002H
        MAR   *,AR0             ;ARP = AR0
        LACL  *                 ;LOAD ACC = DATA @ AR0
        SUB   #02FEH            ;SUB 02FEH
        SUB   #050H             ;SUB 050H
        BCND CH3_NOK,GEQ        ;BRANCH TO CH3_NOK IF ACC >= 0
        MAR   *,AR0             ;ARP = AR0
        LACL  *                 ;LOAD ACC = DATA @ AR0
        SUB   #02FEH            ;SUB 02FEH
        ADD   #050H             ;SUB 050H
        BCND CH3_NOK,LEQ        ;BRANCH TO CH3_NOK IF ACC <= 0
        LACL #04H               ;LOAD ACC = 04H ( 00000000000100b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY                      ;DELAY
```

```
        B       CH_4            ;BRANCH TO CH_4
CH3_NOK:
        LACL #04H               ;LOAD ACC = 04H ( 000000000000100b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        B       CH3_NOK         ;BRANCH TO CH3_NOK
CH_4:
        LAR  AR0,#2003H         ;AR0 = 2003H
        MAR  *,AR0              ;ARP = AR0
        LACL *                  ;LOAD ACC = DATA @ AR0
        SUB  #029FH             ;SUB 029FH
        SUB  #050H              ;SUB 050H
        BCND CH4_NOK,GEQ        ;BRANCH TO CH4_NOK IF ACC >= 0
        MAR  *,AR0              ;ARP = AR0
        LACL *                  ;LOAD ACC = DATA @ AR0
        SUB  #029FH             ;SUB 029FH
        ADD  #050H              ;SUB 050H
        BCND CH4_NOK,LEQ        ;BRANCH TO CH4_NOK IF ACC <= 0
        LACL #08H               ;LOAD ACC = 01H ( 000000000001000b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY                      ;DELAY
```

```
        B       CH_5            ;BRANCH TO CH_5
CH4_NOK:
        LACL #08H               ;LOAD ACC = 08H ( 00000000001000b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        B       CH4_NOK         ;BRANCH TO CH4_NOK
CH_5:
        LAR  AR0,#2004H         ;AR0 = 2004H
        MAR  *,AR0              ;ARP = AR0
        LACL *                  ;LOAD ACC = DATA @ AR0
        SUB  #021FH             ;SUB 021FH
        SUB  #050H              ;SUB 050H
        BCND CH5_NOK,GEQ        ;BRANCH TO CH5_NOK IF ACC >= 0
        MAR  *,AR0              ;ARP = AR0
        LACL *                  ;LOAD ACC = DATA @ AR0
        SUB  #021FH             ;SUB 021FH
    ADD  #050H                  ;SUB 050H
        BCND CH5_NOK,LEQ        ;BRANCH TO CH5_NOK IF ACC <= 0
        LACL #010H              ;LOAD ACC = 010H ( 00000000010000b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY                      ;DELAY
```

```
        B       CH_6            ;BRANCH TO CH_6
CH5_NOK:
        LACL #010H              ;LOAD ACC = 010H ( 000000000010000b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        B       CH5_NOK         ;BRANCH TO CH5_NOK
CH_6:
        LAR   AR0,#2005H        ;AR0 = 2005H
        MAR   *,AR0             ;ARP = AR0
        LACL  *                 ;LOAD ACC = DATA @ AR0
        SUB   #01F4H            ;SUB 01F4H
        SUB   #050H             ;SUB 050H
        BCND CH6_NOK,GEQ        ;BRANCH TO CH6_NOK IF ACC >= 0
        MAR   *,AR0             ;ARP = AR0
        LACL  *                 ;LOAD ACC = DATA @ AR0
        SUB   #01F4H            ;SUB 01F4H
        ADD   #050H             ;SUB 050H
        BCND CH6_NOK,LEQ        ;BRANCH TO CH6_NOK IF ACC <= 0
        LACL #020H              ;LOAD ACC = 020H ( 000000000100000b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY                      ;DELAY
```

```
        B       CH_7            ;BRANCH TO CH_7
CH6_NOK:
        LACL #020H              ;LOAD ACC = 020H ( 00000000100000b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        B       CH6_NOK         ;BRANCH TO CH6_NOK
CH_7:
        LACL #03FH
        SACL TMP
        OUT  TMP,0
        CALL DELAY
        ZAC
        SACL TMP
        OUT  TMP,0
        CALL DELAY
        LAR  AR0,#2006H         ;AR0 = 2006H
        MAR  *,AR0              ;ARP = AR0
        LACL *                  ;LOAD ACC = DATA @ AR0
        SUB  #01D5H             ;SUB 01D5H
        SUB  #050H              ;SUB 050H
        BCND CH7_NOK,GEQ        ;BRANCH TO CH7_NOK IF ACC >= 0
        MAR  *,AR0              ;ARP = AR0
```

```
        LACL *              ;LOAD ACC = DATA @ AR0
        SUB  #01D5H         ;SUB 01D5H
       ADD   #050H          ;SUB 050H
        BCND CH7_NOK,LEQ    ;BRANCH TO CH7_NOK IF ACC <= 0
        LACL #01H           ;LOAD ACC = 01H ( 00000000000001b )
        SACL TMP            ;STORE IN TEMP
        OUT  TMP,0          ;OUT TEMP @ 0000H
        CALL DELAY                ;DELAY
        B    CH_8           ;BRANCH TO CH_8
CH7_NOK:
        LACL #01H           ;LOAD ACC = 01H ( 00000000000001b )
        SACL TMP            ;STORE IN TEMP
        OUT  TMP,0          ;OUT TEMP @ 0000H
        CALL DELAY1               ;DELAY
        ZAC                 ;LOAD ACC = 0
        SACL TMP            ;STORE IN TEMP
        OUT  TMP,0          ;OUT TEMP @ 0000H
        CALL DELAY1               ;DELAY
        B    CH7_NOK        ;BRANCH TO CH7_NOK
CH_8:
       LAR  AR0,#2007H      ;AR0 = 2007H
       MAR  *,AR0           ;ARP = AR0
       LACL *               ;LOAD ACC = DATA @ AR0
       SUB  #01A7H          ;SUB 01A7H
      SUB   #050H           ;SUB 050H
      BCND CH8_NOK,GEQ      ;BRANCH TO CH8_NOK IF ACC >= 0
      MAR  *,AR0            ;ARP = AR0
```

```
        LACL *              ;LOAD ACC = DATA @ AR0
        SUB  #01A7H         ;SUB 01A7H
        ADD  #050H          ;SUB 050H
        BCND CH8_NOK,LEQ    ;BRANCH TO CH8_NOK IF ACC <= 0
        LACL #02H           ;LOAD ACC = 02H ( 00000000000010b )
        SACL TMP            ;STORE IN TEMP
        OUT  TMP,0          ;OUT TEMP @ 0000H
        CALL DELAY                ;DELAY
        B    CH_9           ;BRANCH TO CH_9
CH8_NOK:
        LACL #02H           ;LOAD ACC = 02H ( 00000000000010b )
        SACL TMP            ;STORE IN TEMP
        OUT  TMP,0          ;OUT TEMP @ 0000H
        CALL DELAY1               ;DELAY
        ZAC                 ;LOAD ACC = 0
        SACL TMP            ;STORE IN TEMP
        OUT  TMP,0          ;OUT TEMP @ 0000H
        CALL DELAY1               ;DELAY
        B    CH8_NOK        ;BRANCH TO CH8_NOK
CH_9:
        LAR  AR0,#2008H     ;AR0 = 2008H
        MAR  *,AR0          ;ARP = AR0
        LACL *              ;LOAD ACC = DATA @ AR0
        SUB  #014DH         ;SUB 014DH
        SUB  #050H          ;SUB 050H
        BCND CH9_NOK,GEQ    ;BRANCH TO CH9_NOK IF ACC >= 0
        MAR  *,AR0          ;ARP = AR0
```

```
        LACL *                  ;LOAD ACC = DATA @ AR0
        SUB  #014DH             ;SUB 014DH
        ADD  #050H              ;SUB 050H
        BCND CH9_NOK,LEQ        ;BRANCH TO CH9_NOK IF ACC <= 0
        LACL #04H               ;LOAD ACC = 04H ( 00000000000100b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY                      ;DELAY
        B    CH_10              ;BRANCH TO CH_10
CH9_NOK:
        LACL #04H               ;LOAD ACC = 04H ( 00000000000100b )
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL TMP                ;STORE IN TEMP
        OUT  TMP,0              ;OUT TEMP @ 0000H
        CALL DELAY1                     ;DELAY
        B    CH9_NOK            ;BRANCH TO CH9_NOK
CH_10:
        LAR  AR0,#2009H         ;AR0 = 2009H
        MAR  *,AR0              ;ARP = AR0
        LACL *                  ;LOAD ACC = DATA @ AR0
        SUB  #0C8H              ;SUB 0C8H
        SUB  #050H              ;SUB 050H
        BCND CH10_NOK,GEQ       ;BRANCH TO CH10_NOK IF ACC >= 0
        MAR  *,AR0              ;ARP = AR0
```

```
        LACL *                  ;LOAD ACC = DATA @ AR0
        SUB   #0C8H             ;SUB 0C8H
        ADD   #050H             ;SUB 050H
        BCND CH10_NOK,LEQ       ;BRANCH TO CH10_NOK IF ACC <= 0
        LACL #08H               ;LOAD ACC = 08H ( 00000000001000b )
        SACL TMP                ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY                    ;DELAY
        B       CH_11           ;BRANCH TO CH_11
CH10_NOK:
        LACL #08H               ;LOAD ACC = 08H ( 00000000001000b )
        SACL TMP                ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                   ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL TMP                ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                   ;DELAY
        B       CH10_NOK        ;BRANCH TO CH10_NOK
CH_11:
       LAR   AR0,#200AH         ;AR0 = 200AH
       MAR  *,AR0               ;ARP = AR0
       LACL *                   ;LOAD ACC = DATA @ AR0
       SUB   #0A5H              ;SUB 0A5H
      SUB   #050H               ;SUB 050H
       BCND CH11_NOK,GEQ        ;BRANCH TO CH11_NOK IF ACC >= 0
       MAR  *,AR0               ;ARP = AR0
```

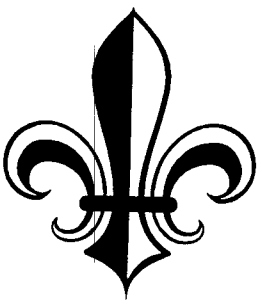
```
        LACL *                  ;LOAD ACC = DATA @ AR0
        SUB   #0A5H             ;SUB 0A5H
        ADD   #050H             ;SUB 050H
        BCND CH11_NOK,LEQ       ;BRANCH TO CH11_NOK IF ACC <= 0
        LACL #010H              ;LOAD ACC = 010H ( 00000000010000b )
        SACL TMP                ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY                    ;DELAY
        B       CH_12           ;BRANCH TO CH_12
CH11_NOK:
        LACL #010H              ;LOAD ACC = 010H ( 00000000010000b )
        SACL TMP                ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                   ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL TMP                ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                   ;DELAY
        B       CH11_NOK        ;BRANCH TO CH11_NOK
CH_12:
       LAR   AR0,#200BH         ;AR0 = 200BH
       MAR   *,AR0              ;ARP = AR0
       LACL *                   ;LOAD ACC = DATA @ AR0
       SUB   #07DH              ;SUB 07DH
      SUB   #050H               ;SUB 050H
      BCND CH12_NOK,GEQ         ;BRANCH TO CH12_NOK IF ACC >= 0
      MAR   *,AR0               ;ARP = AR0
```

```
        LACL *                  ;LOAD ACC = DATA @ AR0
        SUB   #07DH             ;SUB 07DH
        ADD   #050H             ;SUB 050H
        BCND CH12_NOK,LEQ       ;BRANCH TO CH12_NOK IF ACC <= 0
        LACL #020H              ;LOAD ACC = 020H ( 00000000100000b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY                    ;DELAY
        B       FINISH          ;BRANCH TO FINISH
CH12_NOK:
        LACL #020H              ;LOAD ACC = 020H ( 00000000100000b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                   ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                   ;DELAY
        B       CH12_NOK        ;BRANCH TO CH12_NOK
FINISH:
        LACC #303FH             ;LOAD ACC = 303FH ( 110000001111111b )
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP @ 0000H
        CALL DELAY1                   ;DELAY
        ZAC                     ;LOAD ACC = 0
        SACL  TMP               ;STORE IN TEMP
        OUT   TMP,0             ;OUT TEMP 2 0000H
```

```
        CALL DELAY1                 ;DELAY
        B    FINISH                 ;BRANCH TO FINISH
DELAY:
        LAR  AR0,#05H               ;AR0 = 05H
LOOP1:  LAR  AR1,#0FFFFH            ;AR1 = 0FFFFH
LOOP2:  MAR  *,AR1                  ;ARP = AR1
        NOP
        NOP
        BANZ LOOP2                  ;BRANCH TO LOOP2 IF AR1 <> 0
        MAR  *,AR0                  ;ARP = AR0
        BANZ LOOP1                  ;BRANCH TO LOOP1 IF AR0 <> 0
        RET
DELAY1:
        LAR  AR0,#02H               ;AR0 = 02H
LOOP3:  LAR  AR1,#05FFFH            ;AR1 = 5FFFH
LOOP4:  MAR  *,AR1                  ;ARP = AR1
        NOP
        NOP
        BANZ LOOP4                  ;BRANCH TO LOOP4 IF AR1 <> 0
        MAR  *,AR0                  ;ARP = AR1
        BANZ LOOP3                  ;BRANCH TO LOOP3 IF AR0 <> 0
        RET
        .END
```

****

END OF PROGRAM



CONCLUSION

CHAPTER 5

CONCLUSION

The test jig for testing the DSP card has been successfully designed, fabricated, and tested. The test hardware can be used to test the DSP card consisting of any type of DSP processor of any make and the test software needs some modification since all processors do not support the same instruction set.

Till date there is no separate system to test the DSP card, through our project we have successfully eliminated the over comings.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. TEXAS INSTRUMENTS – TMS320C2XX USER MANUAL

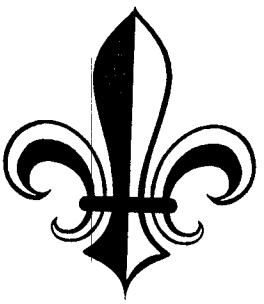
2. D.ROY CHOUDHURY – SHAIL JAIN, LINEAR INTEGRATED CIRCUITS. New Age International (P) Ltd., New Delhi.

3. MALVINO, Electronic Principles, 5th Edition, Tata Mc Graw Hill, 1995.

4. JOHN.G.PROAKIS & DIMITRIS G.MANOLAKIS – DIGITAL SIGNAL PROCESSING- PRINCIPLE ,ALGORITHM AND APPLICATION.

5. WEBSITES

   www.ti.com

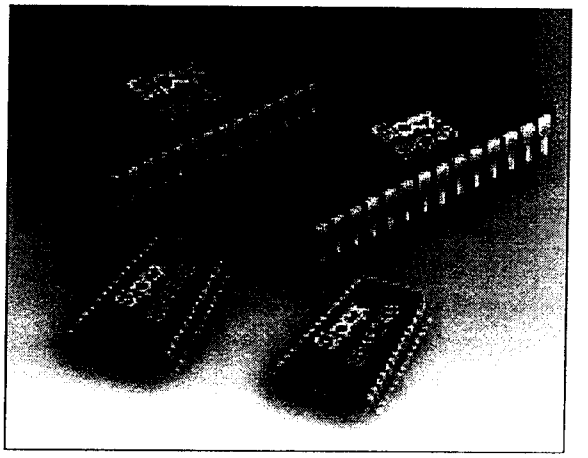


APPENDIX

SP574B/674B/1674B/774B

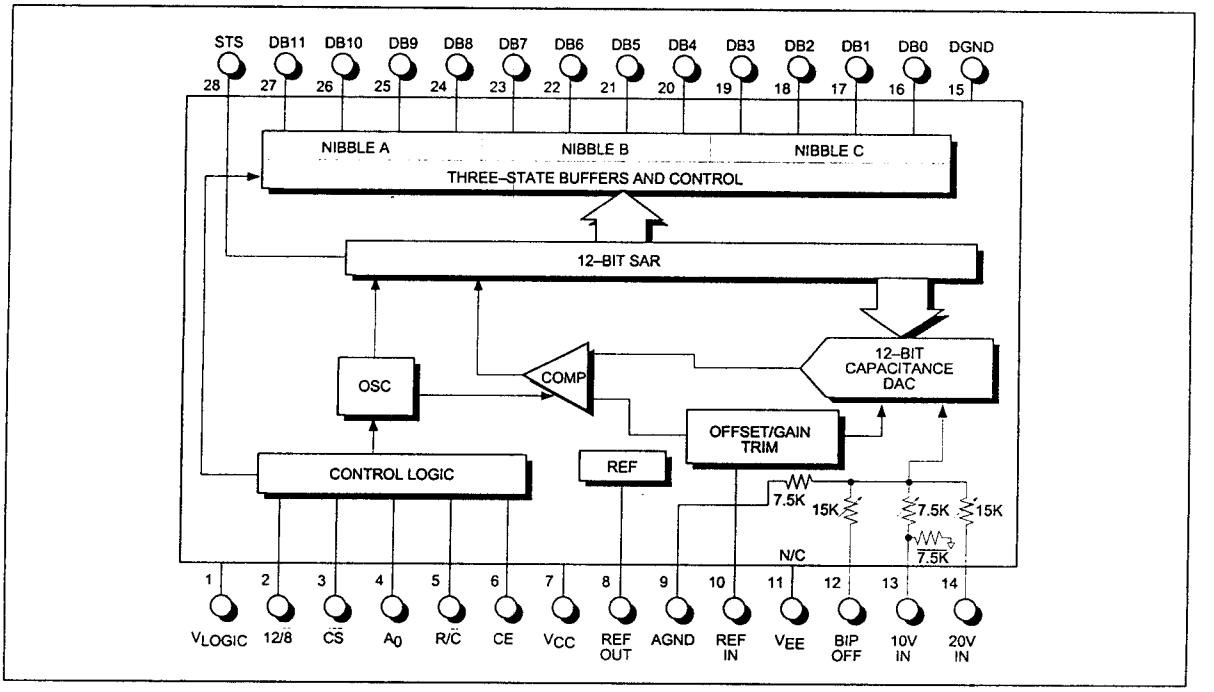
12–Bit Sampling A/D Converters

- Complete Monolithic 12–Bit A/D Converters with Sample–Hold, Reference, Clock and Tri–state Outputs
- Full Nyquist Sampling at All Sample Rates
- Choice of Sampling Rates — 40kHz, 66kHz, 100kHz or 125kHz
- Low Power Dissipation — 110mW
- 12–Bit Linearity Over Temperature
- Commercial, Industrial and Military Temperature Ranges
- Next–Generation Replacement for 574A, 674A, 1674A, 774A Devices



DESCRIPTION...

The **SP574B/674B/1674B/774B (SPx74B) Series** are complete 12–bit successive–approximation A/D converters integrated on a single die with tri-state output latches, an internal reference, clock and a sample–hold. The new "B–Series" features true Nyquist sampling while maintaining compatibility with prior versions. They are drop–in replacements for the older 574A/674A/1674A/774A type devices.

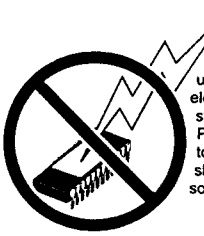


© Copyright 2000 Sipex Corporation

ABSOLUTE MAXIMUM RATINGS

V_{CC} to Digital Common ..... 0 to +16.5V
V_{LOGIC} to Digital Common ..... 0 to +7V
Analog Common to Digital Common ..... ±1V
Control Inputs to Digital Common ..... –0.5V to V_{LOGIC} +0.5V
(CE, CS, A_0, 12/8, R/C)
Analog Input Voltage Range ..... ±FS ±30%
Analog Inputs to Analog Common ..... ±16.5V
(REF IN, BIP OFF, $10V_{IN}$)
$20V_{IN}$ to Analog Common ..... ±24V
REF OUT ..... Indefinite short to common
..... Momentary short to V_{CC}
Power Dissipation 1000mW
Lead Temperature, Soldering ..... 300°C, 10Sec
J/C ..... 45°C/W
MTBF–25°C Ground Base ..... 2.915 million hours
MTBF–125°C Missile Launch ..... 10.16 thousand hours

*Inputs exceeding +30% or –30% of FS will cause erratic performance.



CAUTION:
ESD (ElectroStatic Discharge) sensitive device. Permanent damage may occur on unconnected devices subject to high energy electrostatic fields. Unused devices must be stored in conductive foam or shunts. Personnel should be properly grounded prior to handling this device. The protective foam should be discharged to the destination socket before devices are removed.

SPECIFICATIONS

(Typical @ 25°C with V_{CC} = +15V, V_{EE} = 0V, V_{LOGIC} = +5V unless otherwise noted.)

PARAMETER	MIN.	TYP.	MAX.	UNIT	CONDITIONS
RESOLUTION					
All models			12	Bits	
ANALOG INPUTS					
Input Ranges					
Bipolar		±5, ±10		V	
Unipolar		0 to +10, 0 to +20		V	
Input Impedance					
SP574B/SP674B					
10 Volt Input	3.75		6.25	kΩ	
20 Volt Input	15		25	kΩ	
SP1674B/SP774B					
10 Volt Input	1.875		3.125	kΩ	
20 Volt Input	7.45		12.42	kΩ	
Nyquist Frequency					
SP574B		20		kHz	
SP674B		33		kHz	
SP1674B		50		kHz	
SP774B		62.5		kHz	
DIGITAL INPUTS					
Logic Inputs CE, $\overline{CS}$ R/$\overline{C}$, A_0, 12/$\overline{8}$					
Logic 1	+2.4		+5.5	V	
Logic 0	–0.3		+0.8	V	
Current		±0.1	±50	µA	–0.3V to +5.5V Input
			±5	µA	0V to +5.5V Input
Capacitance		5		pF	
12/$\overline{8}$ Control Input		Hardwire to V_{LOGIC} or DIGITAL COMMON			
DIGITAL OUTPUTS					
Logic Outputs DB_{11}–DB_0, STS					
Logic 1	+2.4			V	$I_{SOURCE} \leq 500\mu A$
Logic 0			+0.4	V	$I_{SINK} \leq 1.6mA$
Leakage (High Z State)			±40	µA	Data bits only
Capacitance		5		pF	
Parallel Data Output Codes					
Unipolar		Positive true binary			
Bipolar		Positive true offset binary			
INTERNAL REFERENCE					
Output Voltage			10.00 ±0.1	V	
Output Current		2		mA	Note 1

© Copyright 2000 Sipex Corporation

SPECIFICATIONS *(continued)*

(Typical @ 25°C with V_{CC} = +15V, V_{EE} = 0V, V_{LOGIC} = +5V unless otherwise noted.)

PARAMETER	MIN.	TYP.	MAX.	UNIT	CONDITIONS
CONVERSION TIME					
SP574B					
12–Bit Conversion	13		25	µs	
8–Bit Conversion	10		19	µs	
SP674B					
12–Bit Conversion	9		15	µs	
8–Bit Conversion	6		11.2	µs	
SP1674B					
12–Bit Conversion	5		10	µs	
8–Bit Conversion	4		7.6	µs	
SP774B					
12–Bit Conversion	4		8	µs	
8–Bit Conversion	3		6	µs	
ACCURACY					
Linearity Error					
–A, –J, –S			±1.0	LSB	@ 25°C and T_{MIN} to T_{MAX}
–B, –K, –T			±0.5	LSB	@ 25°C and T_{MIN} to T_{MAX}
Differential Linearity Error					Note 2
–A, –J, –S	11			Bits	@ 25°C
	11			Bits	T_{MIN} to T_{MAX}
–B, –K, –T	12			Bits	@ 25°C
	12			Bits	T_{MIN} to T_{MAX}
Offset					Note 3
Unipolar		±3		LSB	
Bipolar					
–A, –J, –S		±10		LSB	
–B, –K, –T		±4		LSB	
Full Scale (Gain) Error					% of full scale; T_{MIN} to T_{MAX}
–A			±0.3	%FS	Note 4
			±0.6	%FS	No adjustment @ 25°C
–B			±0.3	%FS	With adjustment @ 25°C
			±0.45	%FS	No adjustment @ 25°C
–J			±0.15	%FS	With adjustment @ 25°C
			±0.5	%FS	No adjustment @ 25°C
–K			±0.22	%FS	With adjustment @ 25°C
			±0.4	%FS	No adjustment @ 25°C
–S			±0.12	%FS	With adjustment @ 25°C
			±0.8	%FS	No adjustment @ 25°C
–T			±0.5	%FS	With adjustment @ 25°C
			±0.6	%FS	No adjustment @ 25°C
			±0.25	%FS	With adjustment @ 25°C
STABILITY					
Unipolar Offset					
–J			±10	ppm/°C	T_{MIN} to T_{MAX}
–K, –A, –S			±5	ppm/°C	T_{MIN} to T_{MAX}
–B, –T			±2.5	ppm/°C	T_{MIN} to T_{MAX}
Bipolar Offset					
–J, –A, –S			±10	ppm/°C	T_{MIN} to T_{MAX}
–K, –B, –T			±5	ppm/°C	T_{MIN} to T_{MAX}
Gain (Scale Factor)					
–J, –A, –S			±50	ppm/°C	T_{MIN} to T_{MAX}
–K, –B, –T			±25	ppm/°C	T_{MIN} to T_{MAX}

© Copyright 2000 Sipex Corporation

SN54/74LS373
SN54/74LS374

OCTAL TRANSPARENT LATCH WITH 3-STATE OUTPUTS; OCTAL D-TYPE FLIP-FLOP WITH 3-STATE OUTPUT

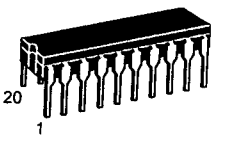
OCTAL TRANSPARENT LATCH WITH 3-STATE OUTPUTS; OCTAL D-TYPE FLIP-FLOP WITH 3-STATE OUTPUT

LOW POWER SCHOTTKY

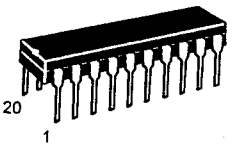
The SN54/74LS373 consists of eight latches with 3-state outputs for bus organized system applications. The flip-flops appear transparent to the data (data changes asynchronously) when Latch Enable (LE) is HIGH. When LE is LOW, the data that meets the setup times is latched. Data appears on the bus when the Output Enable (OE) is LOW. When OE is HIGH the bus output is in the high impedance state.

The SN54/74LS374 is a high-speed, low-power Octal D-type Flip-Flop featuring separate D-type inputs for each flip-flop and 3-state outputs for bus oriented applications. A buffered Clock (CP) and Output Enable (OE) is common to all flip-flops. The SN54/74LS374 is manufactured using advanced Low Power Schottky technology and is compatible with all Motorola TTL families.

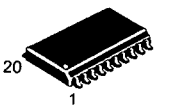
- Eight Latches in a Single Package
- 3-State Outputs for Bus Interfacing
- Hysteresis on Latch Enable
- Edge-Triggered D-Type Inputs
- Buffered Positive Edge-Triggered Clock
- Hysteresis on Clock Input to Improve Noise Margin
- Input Clamp Diodes Limit High Speed Termination Effects



J SUFFIX
CERAMIC
CASE 732-03



N SUFFIX
PLASTIC
CASE 738-03



DW SUFFIX
SOIC
CASE 751D-03

ORDERING INFORMATION

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXDW	SOIC

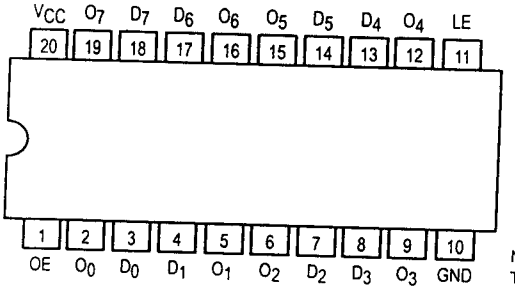
PIN NAMES

		LOADING (Note a) HIGH	LOW
D_0-D_7	Data Inputs	0.5 U.L.	0.25 U.L.
LE	Latch Enable (Active HIGH) Input	0.5 U.L.	0.25 U.L.
CP	Clock (Active HIGH going edge) Input	0.5 U.L.	0.25 U.L.
OE	Output Enable (Active LOW) Input	0.5 U.L.	0.25 U.L.
O_0-O_7	Outputs (Note b)	65 (25) U.L.	15 (7.5) U.L.

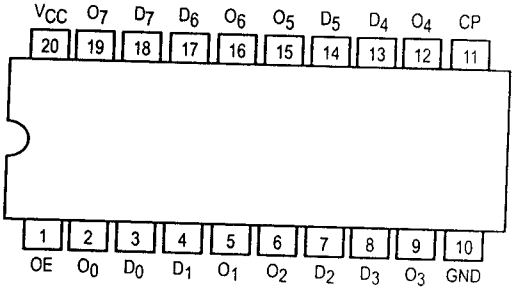
NOTES:
a) 1 TTL Units Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b) The Output LOW drive factor is 7.5 U.L. for Military (54) and 25 U.L. for Commercial (74) Temperature Ranges. The Output HIGH drive factor is 25 U.L. for Military (54) and 65 U.L. for Commercial (74) Temperature Ranges.

CONNECTION DIAGRAM DIP (TOP VIEW)

SN54/74LS373



SN54/74LS374



NOTE:
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

TRUTH TABLE

LS373

D_n	LE	OE	O_n
H	H	L	H
L	H	L	L
X	L	L	Q_0
X	X	H	Z*

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = High Impedance

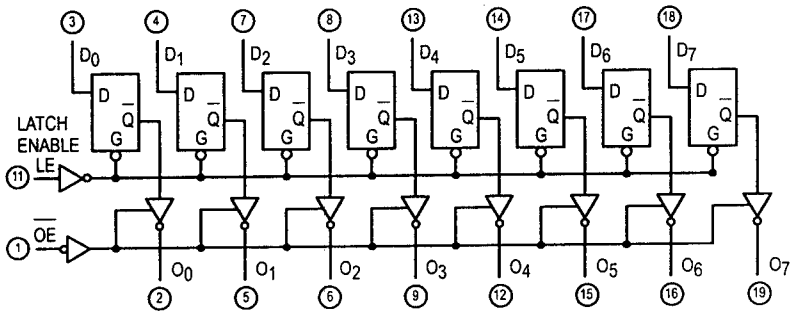
LS374

D_n	LE	OE	O_n
H	↑	L	H
L	↑	L	L
X	X	H	Z*

* Note: Contents of flip-flops unaffected by the state of the Output Enable input ($\overline{OE}$).

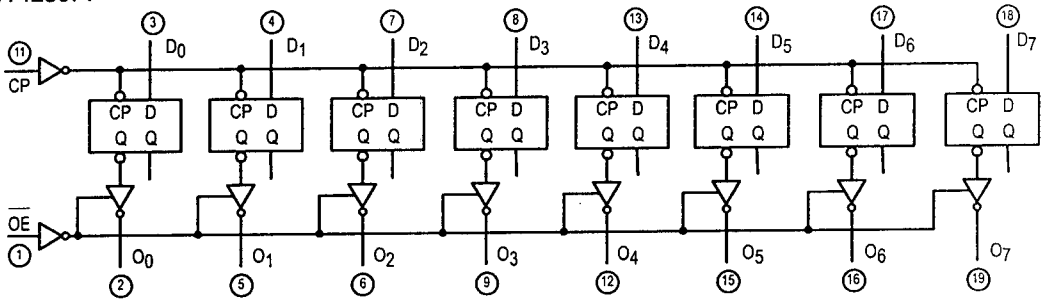
LOGIC DIAGRAMS

SN54LS/74LS373



V_{CC} = PIN 20
GND = PIN 10
◯ = PIN NUMBERS

SN54LS/74LS374



GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54 74			−1.0 −2.6	mA
I_{OL}	Output Current — Low	54 74			12 24	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits Min	Limits Typ	Limits Max	Unit	Test Conditions	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54	2.4	3.4		V	V_{CC} = MIN, I_{OH} = MAX, V_{IN} = V_{IH} or V_{IL} per Truth Table	
		74	2.4	3.1		V		
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I_{OL} = 12 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 24 mA	
I_{OZH}	Output Off Current HIGH				20	μA	V_{CC} = MAX, V_{OUT} = 2.7 V	
I_{OZL}	Output Off Current LOW				−20	μA	V_{CC} = MAX, V_{OUT} = 0.4 V	
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
I_{IL}	Input LOW Current				0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{OS}	Short Circuit Current (Note 1)		−30		−0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{CC}	Power Supply Current				−130	mA	V_{CC} = MAX	
					40	mA	V_{CC} = MAX	

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C, V_{CC} = 5.0 V)

Symbol	Parameter	Limits LS373 Min	LS373 Typ	LS373 Max	LS374 Min	LS374 Typ	LS374 Max	Unit	Test Conditions
f_{MAX}	Maximum Clock Frequency				35	50		MHz	C_L = 45 pF, R_L = 667 Ω
t_{PLH} t_{PHL}	Propagation Delay, Data to Output		12 12	18 18				ns	
t_{PLH} t_{PHL}	Clock or Enable to Output		20 18	30 30		15 19	28 28	ns	
t_{PZH} t_{PZL}	Output Enable Time		15 25	28 36		20 21	28 28	ns	
t_{PHZ} t_{PLZ}	Output Disable Time		12 15	20 25		12 15	20 25	ns	C_L = 5.0 pF

AC SETUP REQUIREMENTS (T_A = 25°C, V_{CC} = 5.0 V)

Symbol	Parameter	Limits LS373 Min	LS373 Max	LS374 Min	LS374 Max	Unit
t_W	Clock Pulse Width	15		15		ns
t_s	Setup Time	5.0		20		ns
t_h	Hold Time	20		0		ns

DEFINITION OF TERMS

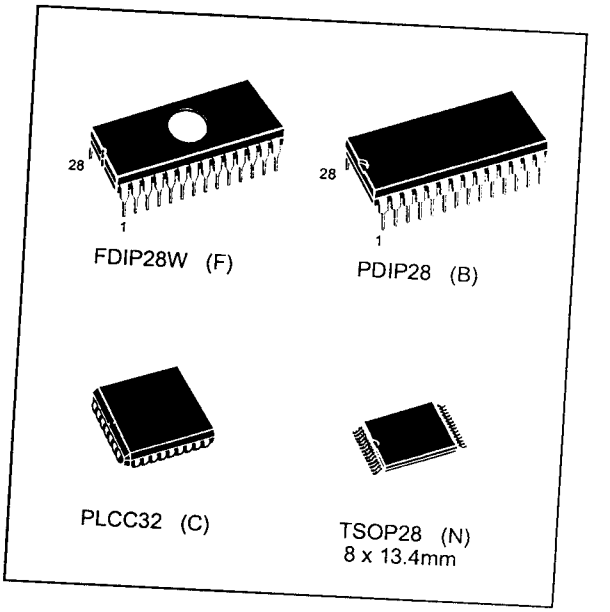
SETUP TIME (t_s) — is defined as the minimum time required for the correct logic level to be present at the logic input prior to LE transition from HIGH-to-LOW in order to be recognized and transferred to the outputs.

HOLD TIME (t_h) — is defined as the minimum time following the LE transition from HIGH-to-LOW that the logic level must be maintained at the input in order to ensure continued recognition.

M27C512

512 Kbit (64Kb x8) UV EPROM and OTP EPROM

- 5V ± 10% SUPPLY VOLTAGE in READ OPERATION
- FAST ACCESS TIME: 45ns
- LOW POWER "CMOS" CONSUMPTION:
  - Active Current 30mA
  - Standby Current 100µA
- PROGRAMMING VOLTAGE: 12.75V ± 0.25V
- PROGRAMMING TIMES of AROUND 6sec. (PRESTO IIB ALGORITHM)
- ELECTRONIC SIGNATURE
  - Manufacturer Code: 20h
  - Device Code: 3Dh



DESCRIPTION

The M27C512 is a 512 Kbit EPROM offered in the two ranges UV (ultra violet erase) and OTP (one time programmable). It is ideally suited for applications where fast turn-around and pattern experimentation are important requirements and is organized as 65,536 by 8 bits.

The FDIP28W (window ceramic frit-seal package) has transparent lid which allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written to the device by following the programming procedure.

For applications where the content is programmed only one time and erasure is not required, the M27C512 is offered in PDIP28, PLCC32 and TSOP28 (8 x 13.4 mm) packages.

Table 1. Signal Names

A0-A15	Address Inputs
Q0-Q7	Data Outputs
$\overline{E}$	Chip Enable
$\overline{G}V_{PP}$	Output Enable / Program Supply
V_{CC}	Supply Voltage
V_{SS}	Ground

Figure 1. Logic Diagram

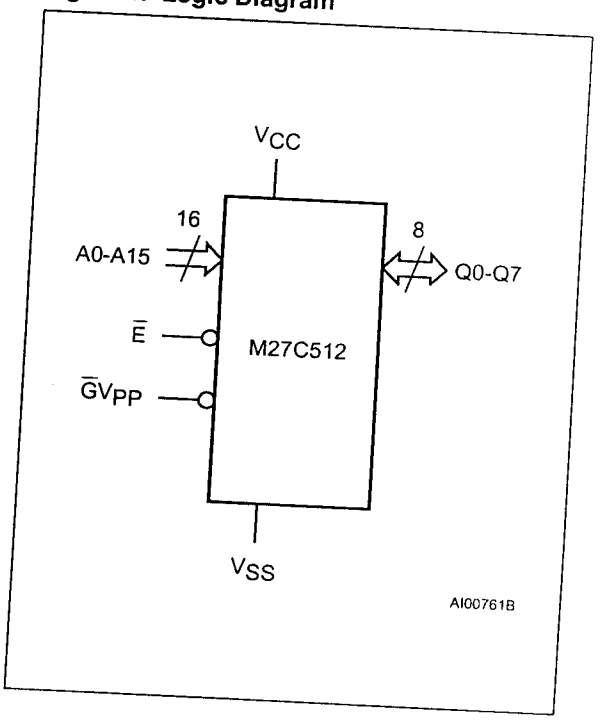


Figure 2A. DIP Pin Connections

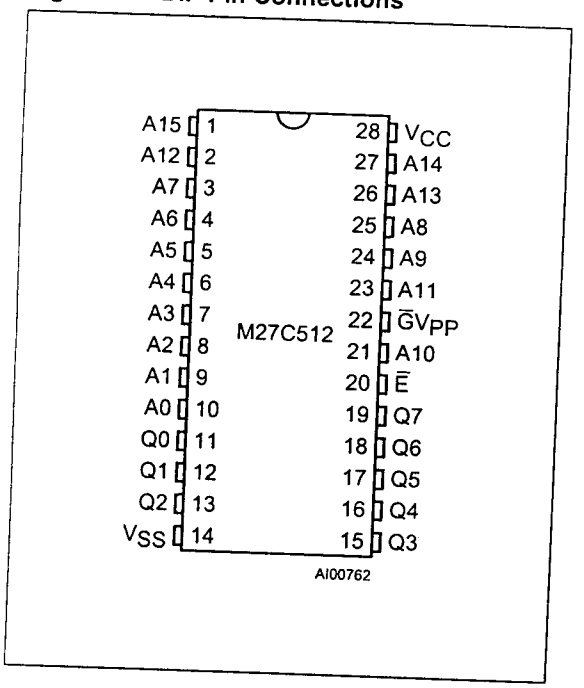
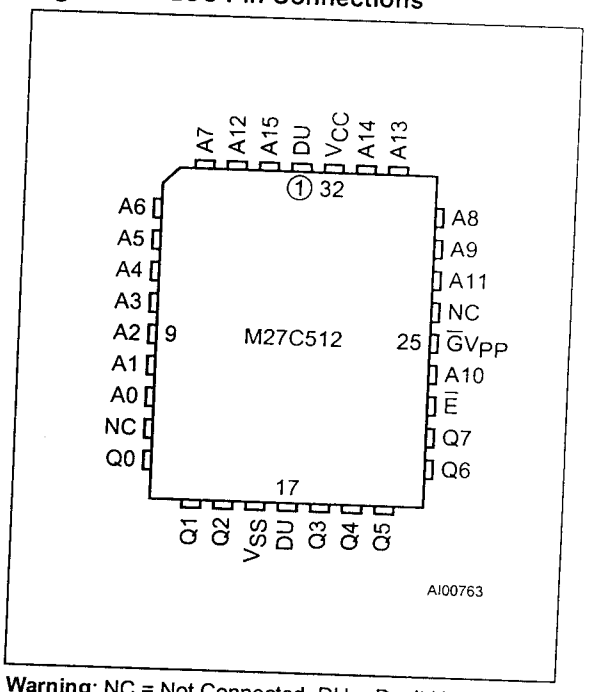
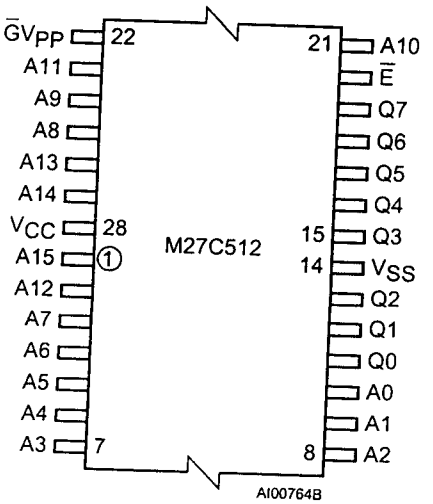


Figure 2B. LCC Pin Connections



Warning: NC = Not Connected, DU = Don't Use

Figure 2C. TSOP Pin Connections



DEVICE OPERATION

The modes of operations of the M27C512 are listed in the Operating Modes table. A single power supply is required in the read mode. All inputs are TTL levels except for $\overline{G}V_{PP}$ and 12V on A9 for Electronic Signature.

Read Mode

The M27C512 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{E}$) is the power control and should be used for device selection. Output Enable ($\overline{G}$) is the output control and should be used to gate data to the output pins, independent of device selection. Assuming that the addresses are stable, the address access time (t_{AVQV}) is equal to the delay from $\overline{E}$ to output (t_{ELQV}). Data is available at the output after a delay of t_{GLQV} from the falling edge of $\overline{G}$, assuming that $\overline{E}$ has been low and the addresses have been stable for at least t_{AVQV}-t_{GLQV}.

Standby Mode

The M27C512 has a standby mode which reduces the active current from 30mA to 100µA The M27C512 is placed in the standby mode by applying a CMOS high signal to the $\overline{E}$ input. When in the standby mode, the outputs are in a high impedance state, independent of the $\overline{G}V_{PP}$ input.

Table 2. Absolute Maximum Ratings (1)

Symbol	Parameter	Value	Unit
T_A	Ambient Operating Temperature (3)	–40 to 125	°C
T_{BIAS}	Temperature Under Bias	–50 to 125	°C
T_{STG}	Storage Temperature	–65 to 150	°C
V_{IO} (2)	Input or Output Voltages (except A9)	–2 to 7	V
V_{CC}	Supply Voltage	–2 to 7	V
V_{A9} (2)	A9 Voltage	–2 to 13.5	V
V_{PP}	Program Supply Voltage	–2 to 14	V

Notes: 1. Except for the rating "Operating Temperature Range", stresses above those listed in the Table "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only and operation of the device at these or any other conditions above those indicated in the Operating sections of this specification is not implied. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability. Refer also to the STMicroelectronics SURE Program and other relevant quality documents.
2. Minimum DC voltage on Input or Output is –0.5V with possible undershoot to –2.0V for a period less than 20ns. Maximum DC voltage on Output is V_{CC} +0.5V with possible overshoot to V_{CC} +2V for a period less than 20ns.
3. Depends on range.

Table 3. Operating Modes

Mode	$\overline{E}$	$\overline{G}V_{PP}$	A9	Q0 - Q7
Read	V_{IL}	V_{IL}	X	Data Out
Output Disable	V_{IL}	V_{IH}	X	Hi-Z
Program	V_{IL} Pulse	V_{PP}	X	Data In
Program Inhibit	V_{IH}	V_{PP}	X	Hi-Z
Standby	V_{IH}	X	X	Hi-Z
Electronic Signature	V_{IL}	V_{IL}	V_{ID}	Codes

Note: X = V_{IH} or V_{IL}, V_{ID} = 12V ± 0.5V

Table 4. Electronic Signature

Identifier	A0	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0	Hex Data
Manufacturer's Code	V_{IL}	0	0	1	0	0	0	0	0	20h
Device Code	V_{IH}	0	0	1	1	1	1	0	1	3Dh

Two Line Output Control

Because EPROMs are usually used in larger memory arrays, the product features a 2 line control function which accommodates the use of multiple memory connection. The two line control function allows:

a. the lowest possible memory power dissipation,

b. complete assurance that output bus contention will not occur.

For the most efficient use of these two control lines, $\overline{E}$ should be decoded and used as the primary device selecting function, while $\overline{G}$ should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This ensures that all deselected memory devices are in their low power standby mode and that the output pins are only active when data is required from a particular memory device.

FINAL

COM'L: H-5/7/10/15/25, Q-10/15/25 IND: H-10/15/20/25

PALCE22V10 Family

24-Pin EE CMOS Versatile PAL Device

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- As fast as 5-ns propagation delay and 142.8 MHz f_{MAX} (external)
- Low-power EE CMOS
- 10 macrocells programmable as registered or combinatorial, and active high or active low to match application needs
- Varied product term distribution allows up to 16 product terms per output for complex functions
- Peripheral Component Interconnect (PCI) compliant (-5/-7/-10)
- Global asynchronous reset and synchronous preset for initialization
- Power-up reset for initialization and register preload for testability
- Extensive third-party software and programmer support through FusionPLD partners
- 24-pin SKINNYDIP, 24-pin SOIC, 24-pin Flat-pack and 28-pin PLCC and LCC packages save space
- 5-ns and 7.5-ns versions utilize split lead-frames for improved performance

GENERAL DESCRIPTION

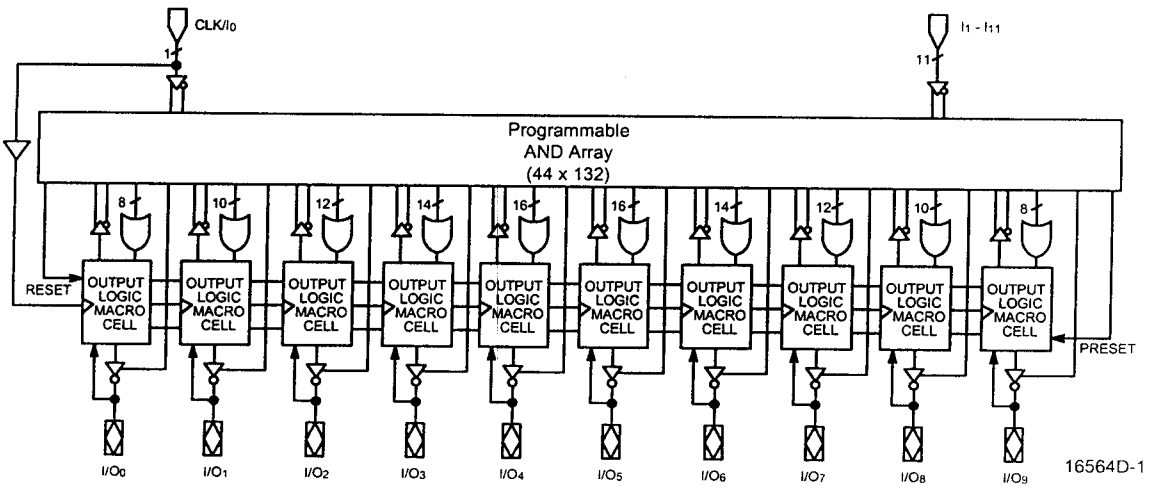
The PALCE22V10 provides user-programmable logic for replacing conventional SSI/MSI gates and flip-flops at a reduced chip count.

The PAL device implements the familiar Boolean logic transfer function, the sum of products. The PAL device is a programmable AND array driving a fixed OR array. The AND array is programmed to create custom product terms, while the OR array sums selected terms at the outputs.

The product terms are connected to the fixed OR array with a varied distribution from 8 to16 across the outputs (see Block Diagram). The OR sum of the products feeds the output macrocell. Each macrocell can be programmed as registered or combinatorial, and active high or active low. The output configuration is determined by two bits controlling two multiplexers in each macrocell.

AMD's FusionPLD program allows PALCE22V10 designs to be implemented using a wide variety of popular industry-standard design tools. By working closely with the FusionPLD partners, AMD certifies that the tools provide accurate, quality support. By ensuring that third-party tools are available, costs are lowered because a designer does not have to buy a complete set of new tools for each device. The FusionPLD program also greatly reduces design time since a designer can use a tool that is already installed and familiar.

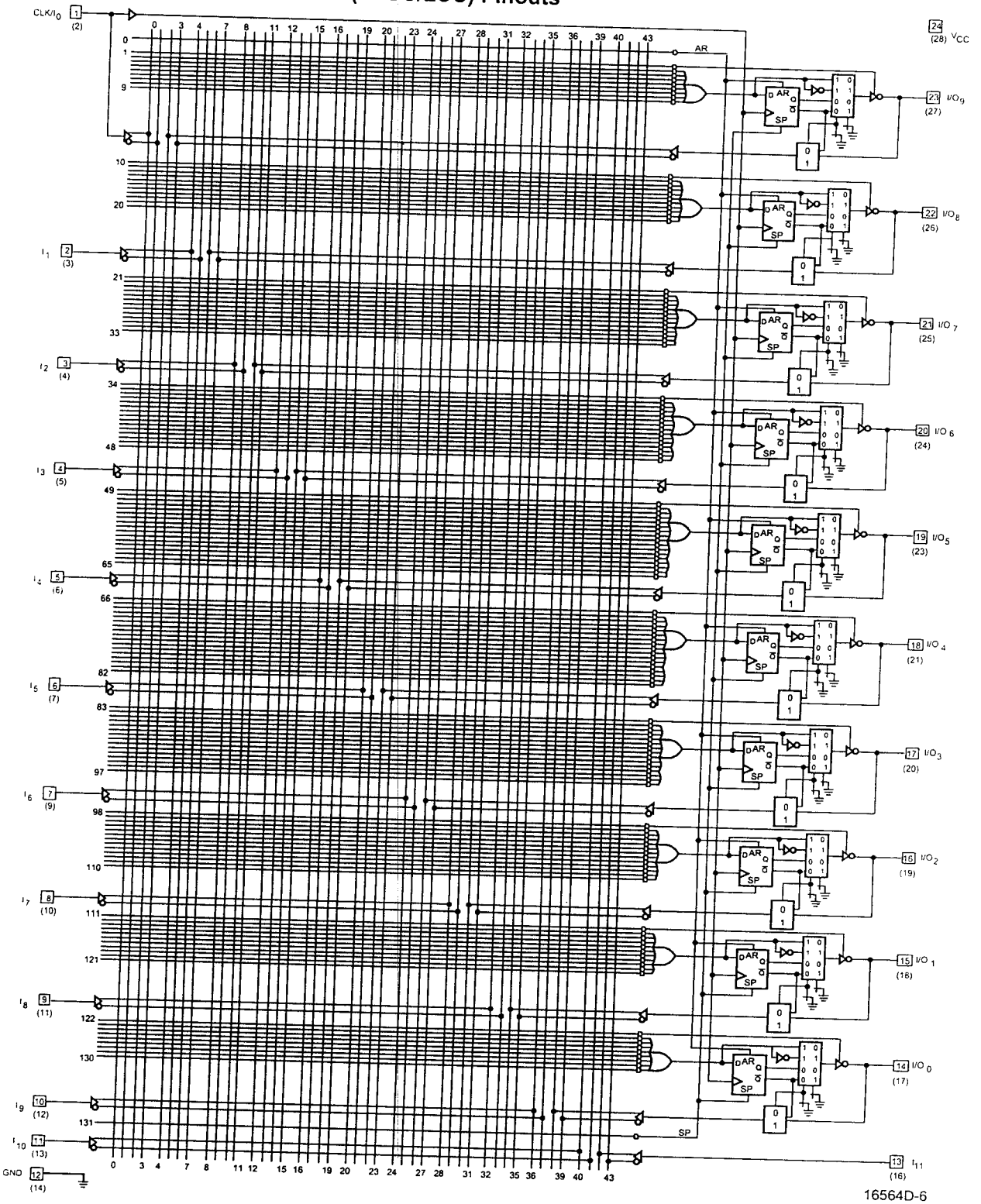
BLOCK DIAGRAM



Publication# 16564 Rev. D Amendment /0
Issue Date: February 1996

LOGIC DIAGRAM

SKINNYDIP/SOIC/FLATPACK (PLCC/LCC) Pinouts



16564D-6

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... –65°C to +150°C

Ambient Temperature with Power Applied ..... –55°C to +125°C

Supply Voltage with Respect to Ground ..... –0.5 V to +7.0 V

DC Input Voltage ..... –0.5 V to V_{CC} + 1.0 V

DC Output or I/O Pin Voltage ..... –0.5 V to V_{CC} + 1.0 V

Static Discharge Voltage ..... 2001 V

Latchup Current (T_A = 0°C to +75°C) ..... 100 mA

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Ambient Temperature (T_A) Operating in Free Air ..... 0°C to +75°C

Supply Voltage (V_{CC}) with Respect to Ground ..... +4.75 V to +5.25 V

Operating Ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
V_{OH}	Output HIGH Voltage	I_{OH} = –3.2 mA V_{IN} = V_{IH} or V_{IL}, V_{CC} = Min	2.4		V
V_{OL}	Output LOW Voltage	I_{OL} = 16 mA V_{IN} = V_{IH} or V_{IL}, V_{CC} = Min		0.4	V
V_{IH}	Input HIGH Voltage	Guaranteed Input Logical HIGH Voltage for all Inputs (Note 1)	2.0		V
V_{IL}	Input LOW Voltage	Guaranteed Input Logical LOW Voltage for all Inputs (Note 1)		0.8	V
I_{IH}	Input HIGH Leakage Current	V_{IN} = V_{CC}, V_{CC} = Max (Note 2)		10	µA
I_{IL}	Input LOW Leakage Current	V_{IN} = 0 V, V_{CC} = Max (Note 2)		–100	µA
I_{OZH}	Off-State Output Leakage Current HIGH	V_{OUT} = V_{CC}, V_{CC} = Max, V_{IN} = V_{IL} or V_{IH} (Note 2)		10	µA
I_{OZL}	Off-State Output Leakage Current LOW	V_{OUT} = 0 V, V_{CC} = Max, V_{IN} = V_{IL} or V_{IH} (Note 2)		–100	µA
I_{SC}	Output Short-Circuit Current	V_{OUT} = 0.5 V, V_{CC} = Max (Note 3)	–30	–130	mA
I_{CC} (Static)	Supply Current	Outputs Open, (I_{OUT} = 0 mA), V_{CC} = Max		125	mA
I_{CC} (Dynamic)	Supply Current	Outputs Open, (I_{OUT} = 0 mA), V_{CC} = Max, f = 25 MHz		140	mA

Notes:

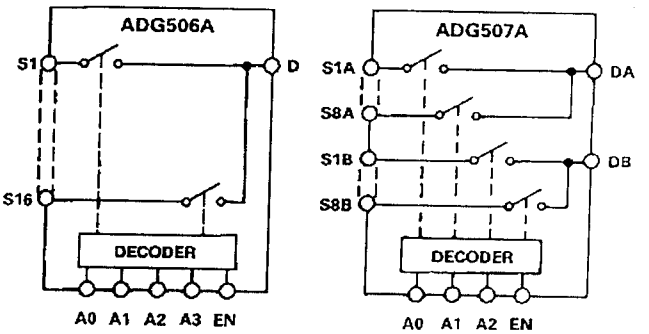
1. *These are absolute values with respect to the device ground and all overshoots due to system and tester noise are included.*
2. *I/O pin leakage is the worst case of I_{IL} and I_{OZL} (or I_{IH} and I_{OZH}).*
3. *Not more than one output should be tested at a time. Duration of the short-circuit test should not exceed one second. V_{OUT} = 0.5 V has been chosen to avoid test problems caused by tester ground degradation.*

CMOS
8-/16-Channel Analog Multiplexers

ADG506A/ADG507A

FEATURES
- 44 V Supply Maximum Rating
- V_{SS} to V_{DD} Analog Signal Range
- Single/Dual Supply Specifications
- Wide Supply Ranges (10.8 V to 16.5 V)
- Extended Plastic Temperature Range (–40°C to +85°C)
- Low Power Dissipation (28 mW max)
- Low Leakage (20 pA typ)
- Available in 28-Lead DIP, SOIC, PLCC, TSSOP and LCCC Packages
- Superior Alternative to:
  - DG506A, HI-506
  - DG507A, HI-507

FUNCTIONAL BLOCK DIAGRAM



GENERAL DESCRIPTION

The ADG506A and ADG507A are CMOS monolithic analog multiplexers with 16 channels and dual 8 channels, respectively. The ADG506A switches one of 16 inputs to a common output, depending on the state of four binary addresses and an enable input. The ADG507A switches one of eight differential inputs to a common differential output, depending on the state of three binary addresses and an enable input. Both devices have TTL and 5 V CMOS logic compatible digital inputs.

The ADG506A and ADG507A are designed on an enhanced LC^2MOS process, which gives an increased signal capability of V_{SS} to V_{DD} and enables operation over a wide range of supply voltages. The devices can operate comfortably anywhere in the 10.8 V to 16.5 V single or dual supply range. These multiplexers also feature high switching speeds and low R_{ON}.

PRODUCT HIGHLIGHTS

1. Single/Dual Supply Specifications with a Wide Tolerance
   The devices are specified in the 10.8 V to 16.5 V range for both single and dual supplies.
2. Extended Signal Range
   The enhanced LC^2MOS processing results in a high breakdown and an increased analog signal range of V_{SS} to V_{DD}.
3. Break-Before-Make Switching
   Switches are guaranteed break-before-make so input signals are protected against momentary shorting.
4. Low Leakage
   Leakage currents in the range of 20 pA make these multiplexers suitable for high precision circuits.

ORDERING GUIDE

Model[1]	Temperature Range	Package Option[2]
ADG506AKN	–40°C to +85°C	N-28
ADG506AKR	–40°C to +85°C	R-28
ADG506AKP	–40°C to +85°C	P-28A
ADG506ABQ	–40°C to +85°C	Q-28
ADG506ATQ	–55°C to +125°C	Q-28
ADG506ATE	–55°C to +125°C	E-28A
ADG507AKN	–40°C to +85°C	N-28
ADG507AKR	–40°C to +85°C	R-28
ADG507AKP	–40°C to +85°C	P-28A
ADG507AKRU	–40°C to +85°C	RU-28
ADG507ABQ	–40°C to +85°C	Q-28
ADG507ATQ	–55°C to +125°C	Q-28
ADG507ATE	–55°C to +125°C	E-28A

NOTES
[1]To order MIL-STD-883, Class B processed parts, add /883B to part number. See Analog Devices' *Military/Aerospace Reference Manual* (1994) for military data sheet.
[2]E = Leadless Ceramic Chip Carrier (LCCC); N = Plastic DIP; P = Plastic Leaded Chip Carrier (PLCC); Q = Cerdip; R = 0.3" Small Outline IC (SOIC); RU = Thin Shrink Small Outline Package (TSSOP).

REV. C

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 World Wide Web Site: http://www.analog.com
Fax: 781/326-8703
© Analog Devices, Inc., 1998

ADG506A/ADG507A–SPECIFICATIONS

Dual Supply (V_{DD} = +10.8 V to +16.5 V, V_{SS} = –10.8 V to –16.5 V unless otherwise noted)

Parameter	ADG506A ADG507A K Version +25°C	K Version –40°C to +85°C	ADG506A ADG507A B Version +25°C	B Version –40°C to +85°C	ADG506A ADG507A T Version +25°C	T Version –55°C to +125°C	Units	Comments
ANALOG SWITCH								
Analog Signal Range	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V min	
	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V max	
R_{ON}	280		280		280		Ω typ	–10 V ≤ V_S ≤ +10 V, I_{DS} = 1 mA; Test Circuit 1
	450	600	450	600	450	600	Ω max	
	300	400	300	400			Ω max	V_{DD} = 15 V (±10%), V_{SS} = –15 V (±10%)
					300	400	Ω max	V_{DD} = 15 V (±5%), V_{SS} = –15 V (±5%)
R_{ON} Drift	0.6		0.6		0.6		%/°C typ	–10 V ≤ V_S ≤ +10 V, I_{DS} = 1 mA
R_{ON} Match	5		5		5		% typ	–10 V ≤ V_S ≤ +10 V, I_{DS} = 1 mA
I_S (OFF), Off Input Leakage	0.02		0.02		0.02		nA typ	V1 = ±10 V, V2 = ∓10 V; Test Circuit 2
	1	50	1	50	1	50	nA max	
I_D (OFF), Off Output Leakage	0.04		0.04		0.04		nA typ	V1 = ±10 V, V2 = ∓10 V; Test Circuit 3
ADG506A	1	200	1	200	1	200	nA max	
ADG507A	1	100	1	100	1	100	nA max	
I_D (ON), On Channel Leakage	0.04		0.04		0.04		nA typ	V1 = ±10 V, V2 = ∓10 V; Test Circuit 4
ADG506A	1	200	1	200	1	200	nA max	
ADG507A	1	100	1	100	1	100	nA max	
I_{DIFF}, Differential Off Output Leakage (ADG507A Only)		25		25		25	nA max	V1 = ±10 V, V2 = ∓10 V; Test Circuit 5
DIGITAL CONTROL								
V_{INH}, Input High Voltage		2.4		2.4		2.4	V min	
V_{INL}, Input Low Voltage		0.8		0.8		0.8	V max	
I_{INL} or I_{INH}		1		1		1	μA max	V_{IN} = 0 to V_{DD}
C_{IN} Digital Input Capacitance	8		8		8		pF max	
DYNAMIC CHARACTERISTICS								
$t_{TRANSITION}$[1]	200		200		200		ns typ	V1 = ±10 V, V2 = +10 V; Test Circuit 6
	300	400	300	400	300	400	ns max	
t_{OPEN}[1]	50		50		50		ns typ	Test Circuit 7
	25	10	25	10	25	10	ns min	
t_{ON} (EN)[1]	200		200		200		ns typ	Test Circuit 8
	300	400	300	400	300	400	ns max	
t_{OFF} (EN)[1]	200		200		200		ns typ	Test Circuit 8
	300	400	300	400	300	400	ns max	
OFF Isolation	68		68		68		dB typ	V_{EN} = 0.8 V, R_L = 1 kΩ, C_L = 15 pF,
	50		50		50		dB min	V_S = 7 V rms, f = 100 kHz
C_S (OFF)	5		5		5		pF typ	V_{EN} = 0.8 V
C_D (OFF)								
ADG506A	44		44		44		pF typ	V_{EN} = 0.8 V
ADG507A	22		22		22		pF typ	
Q_{INJ}, Charge Injection	4		4		4		pC typ	R_S = 0 Ω, V_S = 0 V; Test Circuit 9
POWER SUPPLY								
I_{DD}	0.6		0.6		0.6		mA typ	V_{IN} = V_{INL} or V_{INH}
		1.5		1.5		1.5	mA max	
I_{SS}	20		20		20		μA typ	V_{IN} = V_{IN} or V_{INH}
		0.2		0.2		0.2	mA max	
Power Dissipation	10		10		10		mW typ	
		28		28		28	mW max	

NOTES

[1]Sample tested at +25°C to ensure compliance.

Specifications subject to change without notice.

Single Supply (V_{DD} = +10.8 V to +16.5 V, V_{SS} = GND = 0 V unless otherwise noted)

Parameter	ADG506A ADG507A K Version +25°C	ADG506A ADG507A K Version −40°C to +85°C	ADG506A ADG507A B Version +25°C	ADG506A ADG507A B Version −40°C to +85°C	ADG506A ADG507A T Version +25°C	ADG506A ADG507A T Version −55°C to +125°C	Units	Comments
ANALOG SWITCH								
Analog Signal Range	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V min	
	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V_{DD}	V max	
R_{ON}	500		500		500		Ω typ	0 V ≤ V_S ≤ +10 V, I_{DS} = 0.5 mA; Test Circuit 1
	700	1000	700	1000	700	1000	Ω max	
R_{ON} Drift	0.6		0.6		0.6		%/°C typ	0 V ≤ V_S ≤ +10 V, I_{DS} = 0.5 mA
R_{ON} Match	5		5		5		% typ	0 V ≤ V_S ≤ +10 V, I_{DS} = 0.5 mA
I_S (OFF), Off Input Leakage	0.02		0.02		0.02		nA typ	V1 = +10 V/0 V, V2 = 0 V/ +10 V;
	1	50	1	50	1	50	nA max	Test Circuit 2
I_D (OFF), Off Output Leakage	0.04		0.04		0.04		nA typ	V1 = +10 V/0 V, V2 = 0 V/ +10 V;
ADG506A	1	200	1	200	1	200	nA max	Test Circuit 3
ADG507A	1	100	1	100	1	100	nA max	
I_D (ON), On Channel Leakage	0.04		0.04		0.04		nA typ	V1 = +10 V/0 V, V2 = 0 V/ +10 V;
ADG506A	1	200	1	200	1	200	nA max	Test Circuit 4
ADG507A	1	100	1	100	1	100	nA max	
I_{DIFF}, Differential Off Output Leakage (ADG507A Only)		25		25		25	nA max	V1 = +10 V/0 V, V2 = 0 V/ +10 V; Test Circuit 5
DIGITAL CONTROL								
V_{INH}, Input High Voltage		2.4		2.4		2.4	V min	
V_{INL}, Input Low Voltage		0.8		0.8		0.8	V max	
I_{INL} or I_{INH}		1		1		1	μA max	V_{IN} = 0 to V_{DD}
C_{IN} Digital Input Capacitance	8		8		8		pF max	
DYNAMIC CHARACTERISTICS								
$t_{TRANSITION}$[1]	300		300		300		ns typ	V1 = +10 V/0 V, V2 = +10 V; Test Circuit 6
	450	600	450	600	450	600	ns max	
t_{OPEN}[1]	50		50		50		ns typ	Test Circuit 7
	25	10	25	10	25	10	ns min	
t_{ON} (EN)[1]	250		250		250		ns typ	Test Circuit 8
	450	600	450	600	450	600	ns max	
t_{OFF} (EN)[1]	250		250		250		ns typ	Test Circuit 8
	450	600	450	600	450	600	ns max	
OFF Isolation	68		68		68		dB typ	V_{EN} = 0.8 V, R_L = 1 kΩ, C_L = 15 pF,
	50		50		50		dB min	V_S = 3.5 V rms, f = 100 kHz
C_S (OFF)	5		5		5		pF typ	V_{EN} = 0.8 V
C_D (OFF)								
ADG506A	44		44		44		pF typ	V_{EN} = 0.8 V
ADG507A	22		22		22		pF typ	
Q_{INJ}, Charge Injection	4		4		4		pC typ	R_S = 0 Ω, V_S = 0 V; Test Circuit 9
POWER SUPPLY								
I_{DD}	0.6		0.6		0.6		mA typ	V_{IN} = V_{INL} or V_{INH}
		1.5		1.5		1.5	mA max	
Power Dissipation	10		10		10		mW typ	
		25		25		25	mW max	

NOTES

[1]Sample tested at +25°C to ensure compliance.

Specifications subject to change without notice.

Truth Table (ADG506A)

A3	A2	A1	A0	EN	On Switch
X	X	X	X	0	NONE
0	0	0	0	1	1
0	0	0	1	1	2
0	0	1	0	1	3
0	0	1	1	1	4
0	1	0	0	1	5
0	1	0	1	1	6
0	1	1	0	1	7
0	1	1	1	1	8
1	0	0	0	1	9
1	0	0	1	1	10
1	0	1	0	1	11
1	0	1	1	1	12
1	1	0	0	1	13
1	1	0	1	1	14
1	1	1	0	1	15
1	1	1	1	1	16

Truth Table (ADG507A)

A2	A1	A0	EN	On Switch Pair
X	X	X	0	NONE
0	0	0	1	1
0	0	1	1	2
0	1	0	1	3
0	1	1	1	4
1	0	0	1	5
1	0	1	1	6
1	1	0	1	7
1	1	1	1	8

X = Don't Care

Typical Performance Characteristics–ADG506A/ADG507A

The multiplexers are guaranteed functional with reduced single or dual supplies down to 4.5 V.

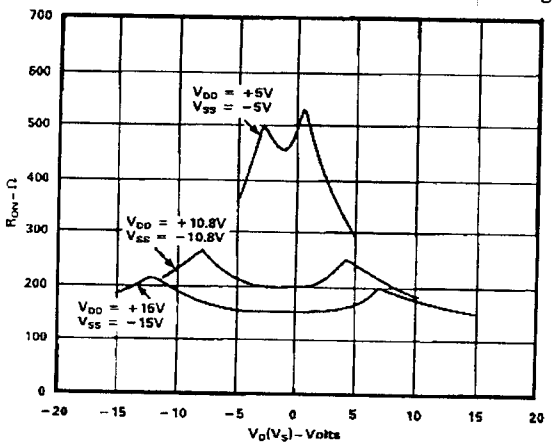


Figure 1. R_{ON} as a Function of V_D (V_S): Dual Supply Voltage, T_A = +25°C

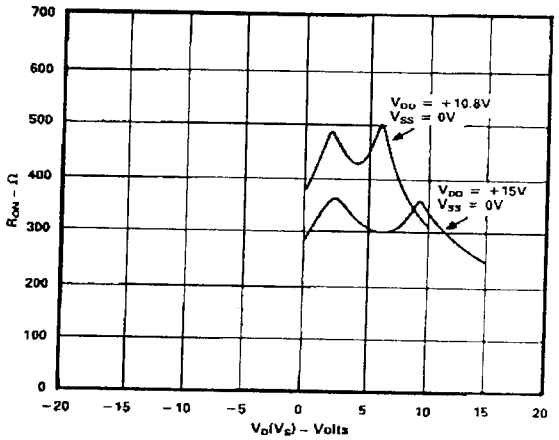


Figure 4. R_{ON} as a Function of V_D (V_S) Single Supply Voltage, T_A = +25°C

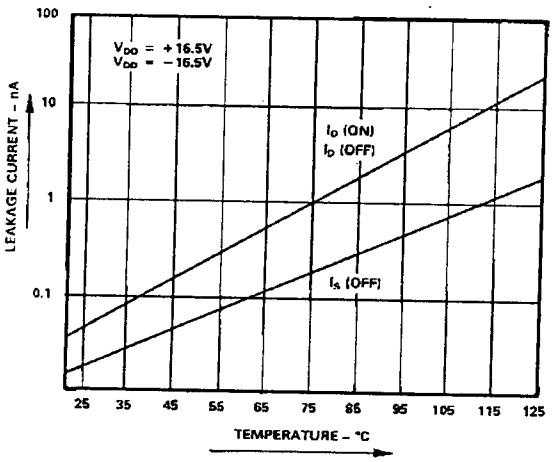


Figure 2. Leakage Current as a Function of Temperature (Note: Leakage Currents Reduce as the Supply Voltages Reduce)

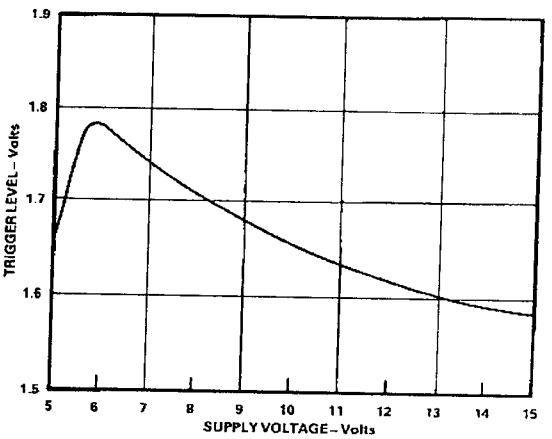


Figure 5. Trigger Levels vs. Power Supply Voltage, Dual or Single Supply, T_A = +25°C

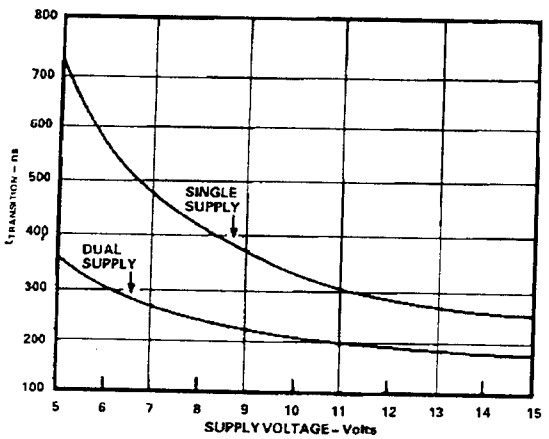


Figure 3. $t_{TRANSITION}$ vs. Supply Voltage: Dual and Single Supplies, T_A = + 25°C (Note: For V_{DD} and /V_{SS}/ < 10 V; V1 = V_{DD}/V_{SS}, V2 = V_{SS}/V_{DD}. See Test Circuit 6)

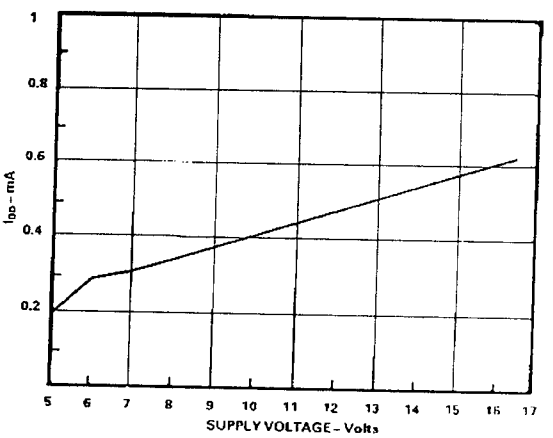


Figure 6. I_{DD} vs. Supply Voltage: Dual or Single Supply, T_A = +25°C

OCTAL BUS TRANSCEIVER

SN54/74LS245

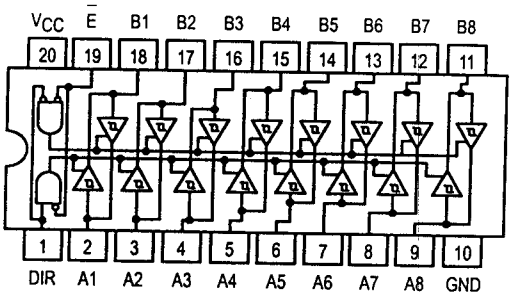
OCTAL BUS TRANSCEIVER

LOW POWER SCHOTTKY

The SN54/74LS245 is an Octal Bus Transmitter/Receiver designed for 8-line asynchronous 2-way data communication between data buses. Direction Input (DR) controls transmission of Data from bus A to bus B or bus B to bus A depending upon its logic level. The Enable input (E) can be used to isolate the buses.

- Hysteresis Inputs to Improve Noise Immunity
- 2-Way Asynchronous Data Bus Communication
- Input Diodes Limit High-Speed Termination Effects
- ESD > 3500 Volts

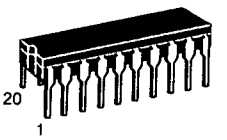
LOGIC AND CONNECTION DIAGRAMS DIP (TOP VIEW)



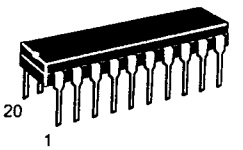
TRUTH TABLE

INPUTS		OUTPUT
E	DIR	
L	L	Bus B Data to Bus A
L	H	Bus A Data to Bus B
H	X	Isolation

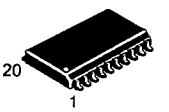
H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial



J SUFFIX
CERAMIC
CASE 732-03



N SUFFIX
PLASTIC
CASE 738-03



DW SUFFIX
SOIC
CASE 751D-03

ORDERING INFORMATION

SN54LSXXXJ Ceramic
SN74LSXXXN Plastic
SN74LSXXXDW SOIC

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			−3.0	mA
		54 74			−12 −15	mA
I_{OL}	Output Current — Low	54 74			12 24	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits Min	Limits Typ	Limits Max	Unit	Test Conditions	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
$V_{T+}-V_{T-}$	Hysteresis		0.2	0.4		V	V_{CC} = MIN	
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54, 74	2.4	3.4		V	V_{CC} = MIN, I_{OH} = −3.0 mA	
		54, 74	2.0			V	V_{CC} = MIN, I_{OH} = MAX	
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I_{OL} = 12 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 24 mA	
I_{OZH}	Output Off Current HIGH				20	μA	V_{CC} = MAX, V_{OUT} = 2.7 V	
I_{OZL}	Output Off Current LOW				−200	μA	V_{CC} = MAX, V_{OUT} = 0.4 V	
I_{IH}	Input HIGH Current	A or B, DR or $\overline{E}$			20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
		DR or $\overline{E}$			0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
		A or B			0.1	mA	V_{CC} = MAX, V_{IN} = 5.5 V	
I_{IL}	Input LOW Current				−0.2	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Output Short Circuit Current (Note 1)		−40		−225	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current Total, Output HIGH				70	mA	V_{CC} = MAX	
	Total, Output LOW				90			
	Total at HIGH Z				95			

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C, V_{CC} = 5.0 V, $T_{RISE}/T_{FALL} \leq$ 6.0 ns)

Symbol	Parameter	Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
t_{PLH} t_{PHL}	Propagation Delay, Data to Output		8.0 8.0	12 12	ns	C_L = 45 pF, R_L = 667 Ω
t_{PZH}	Output Enable Time to HIGH Level		25	40	ns	
t_{PZL}	Output Enable Time to LOW Level		27	40	ns	
t_{PLZ}	Output Disable Time from LOW Level		15	25	ns	C_L = 5.0 pF, R_L = 667 Ω
t_{PHZ}	Output Disable Time from HIGH Level		15	25	ns	

SN54/74LS273

OCTAL D FLIP-FLOP WITH CLEAR

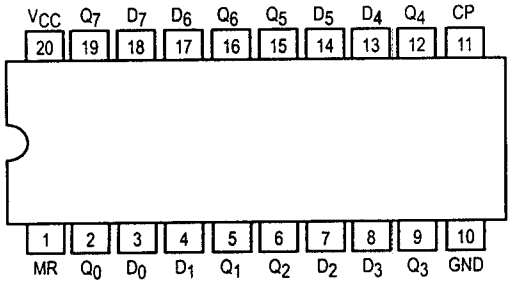
The SN54/74LS273 is a high-speed 8-Bit Register. The register consists of eight D-Type Flip-Flops with a Common Clock and an asynchronous active LOW Master Reset. This device is supplied in a 20-pin package featuring 0.3 inch lead spacing.

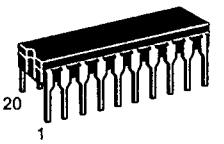
- 8-Bit High Speed Register
- Parallel Register
- Common Clock and Master Reset
- Input Clamp Diodes Limit High-Speed Termination Effects

OCTAL D FLIP-FLOP WITH CLEAR

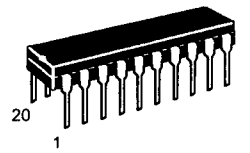
LOW POWER SCHOTTKY

CONNECTION DIAGRAM DIP (TOP VIEW)

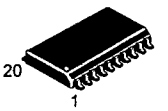




J SUFFIX
CERAMIC
CASE 732-03



N SUFFIX
PLASTIC
CASE 738-03



DW SUFFIX
SOIC
CASE 751D-03

PIN NAMES		LOADING (Note a)	
		HIGH	LOW
CP	Clock (Active HIGH Going Edge) Input	0.5 U.L.	0.25 U.L.
$\overline{D_0}$–D_7	Data Inputs	0.5 U.L.	0.25 U.L.
MR	Master Reset (Active LOW) Input	0.5 U.L.	0.25 U.L.
Q_0–Q_7	Register Outputs (Note b)	10 U.L.	5 (2.5) U.L.

NOTES:
a) 1 TTL Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b) The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

TRUTH TABLE

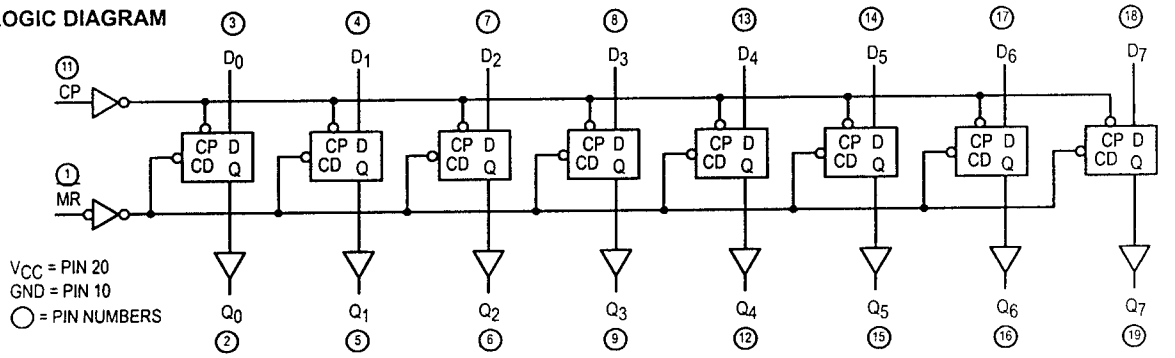
$\overline{MR}$	CP	D_X	Q_X
L	X	X	L
H	⎍	H	H
H	⎍	L	L

H = HIGH Logic Level
L = LOW Logic Level
X = Immaterial

ORDERING INFORMATION

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXDW	SOIC

LOGIC DIAGRAM



FUNCTIONAL DESCRIPTION

The SN54/74LS273 is an 8-Bit Parallel Register with a common Clock and common Master Reset.

When the MR input is LOW, the Q outputs are LOW, independent of the other inputs. Information meeting the setup and hold time requirements of the D inputs is transferred to the Q outputs on the LOW-to-HIGH transition of the clock input.

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			−0.4	mA
I_{OL}	Output Current — Low	54 74			4.0 8.0	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits Min	Limits Typ	Limits Max	Unit	Test Conditions	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54	2.5	3.5		V	V_{CC} = MIN, I_{OH} = MAX, V_{IN} = V_{IH} or V_{IL} per Truth Table	
		74	2.7	3.5		V		
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I_{OL} = 4.0 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 8.0 mA	
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{IL}	Input LOW Current				−0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Short Circuit Current (Note 1)		−20		−100	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current				27	mA	V_{CC} = MAX	

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

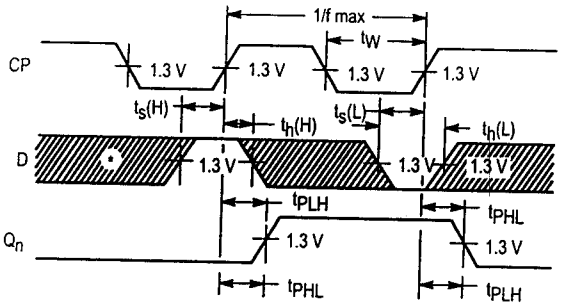
AC CHARACTERISTICS (T_A = 25°C, V_{CC} = 5.0 V)

Symbol	Parameter	Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
f_{MAX}	Maximum Input Clock Frequency	30	40		MHz	Figure 1
t_{PHL}	Propagation Delay, MR to Q Output		18	27	ns	Figure 2
t_{PLH} t_{PHL}	Propagation Delay, Clock to Output		17 18	27 27	ns	Figure 1

AC SETUP REQUIREMENTS (T_A = 25°C, V_{CC} = 5.0 V)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
t_w	Pulse Width, Clock or Clear	20			ns	Figure 1
t_s	Data Setup Time	20			ns	Figure 1
t_h	Hold Time	5.0			ns	Figure 1
t_{rec}	Recovery Time	25			ns	Figure 2

AC WAVEFORMS



*The shaded areas indicate when the input is permitted to change for predictable output performance.

Figure 1. Clock to Output Delays, Clock Pulse Width, Frequency, Setup and Hold Times Data to Clock

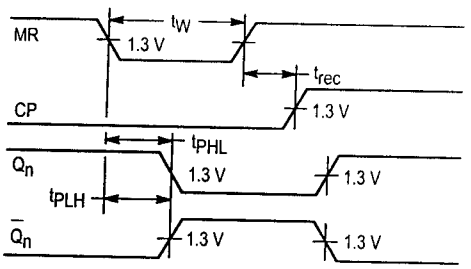


Figure 2. Master Reset to Output Delay, Master Reset Pulse Width, and Master Reset Recovery Time

DEFINITION OF TERMS

SETUP TIME (t_s) — is defined as the minimum time required for the correct logic level to be present at the logic input prior to the clock transition from LOW-to-HIGH in order to be recognized and transferred to the outputs.

HOLD TIME (t_h) — is defined as the minimum time following the clock transition from LOW-to-HIGH that the logic level must be maintained at the input in order to ensure continued recognition. A negative HOLD TIME indicates that the correct logic level may be released prior to the clock transition from LOW-to-HIGH and still be recognized.

RECOVERY TIME (t_{rec}) — is defined as the minimum time required between the end of the reset pulse and the clock transition from LOW-to-HIGH in order to recognize and transfer HIGH data to the Q outputs.