

PENGUKUR SUHU BERBASIS KOMPUTER



Disusun oleh :

**Delianto
011880027
883206710250026**

**INSTITUT TEKNOLOGI INDONESIA
FAKULTAS TEKNOLOGI INDONESIA
JURUSAN TEKNIK ELEKTRO
SERPONG
1997**

PENGUKUR SUHU BERBASIS KOMPUTER

Disusun oleh :

Delianto
011880027
883206710250026

**Tugas Akhir ini diajukan untuk memenuhi persyaratan
Kurikulum sarjana strata satu (S1)
Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Indonesia
Serpong**

Serpong, Oktober 1997
Disetujui oleh :

Pembimbing,



(Ir. Endang Sriningsih MT.)

Ketua Jurusan,



(Dipl. Ing. Th. H. Lumbantoruan)

ABSTRAK

Pada Tugas Akhir ini, akan dibahas mengenai "Pengukur Suhu berbasis Komputer ". Sensor suhu yang digunakan, pada saat bekerja dapat menghasilkan tegangan berbanding langsung dengan suhu ruangan yang diukur, dimana semakin panas suhu di sekitar sensor maka semakin besar pula tegangannya.

Sensor suhu yang dipergunakan adalah IC LM 35 yang dipakai sebagai alat ukur untuk suhu udara sebesar 0° Celcius sampai dengan 150° Celcius dengan resolusi 1° Celcius. Tegangan keluaran (V_{out}) pada sensor suhu adalah penambahan 10 mVolt, setiap perubahan suhu 1° Celcius.

Keluaran tegangan dari sensor suhu yang masih berupa sinyal analog diubah menjadi sinyal digital dengan menggunakan komponen ADC 0804. Keluaran dari ADC dihubungkan ke PC (*Personal Computer*) melalui jalur paralel (*Parallel Port*) yang tersedia pada PC.

Pengukuran dan penampilan data dilakukan oleh PC, yang dijalankan oleh sebuah program yang menetap di memori. Bahasa pemrograman yang digunakan adalah PASCAL.

KATA PENGANTAR

Puji syukur Penulis panjatkan kepada Tuhan Yang Maha Esa, yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyusun Tugas Akhir ini.

Tugas Akhir ini disusun untuk memenuhi persyaratan kurikulum sarjana Strata Satu (S-1) pada jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Indonesia. Untuk memenuhi persyaratan tersebut, penulis menyusun Tugas Akhir dengan judul :

“Pengukur Suhu Berbasis Komputer”

Dalam penyusunan Tugas Akhir ini, penulis telah banyak mendapat bantuan dari berbagai pihak. Oleh karenanya penulis ingin mengucapkan banyak terima kasih kepada :

- Orang tua dan kakak penulis yang tercinta, yang telah banyak memberikan dorongan moril maupun materiil.
- Ibu Ir. Endang Sriningsih MT., selaku dosen pembimbing yang telah memberikan bimbingan dan saran kepada penulis dalam penyusunan makalah Tugas Akhir ini.
- Ibu Dr. Ir. Fatimah Z.S.P., selaku koordinator Tugas Akhir jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Indonesia.

- Bapak Dipl. Ing. Th. H. Lumbantoruan sebagai Ketua Jurusan Teknik Elektro, Institut Teknologi Indonesia.
- Rekan-rekan mahasiswa elektro terutama Ir. Herwi, Ir. Benny Purba dan angkatan 1988 yang telah memberikan dukungan dan dorongan kepada penulis, baik secara langsung maupun tidak langsung.

Akhir kata, semoga makalah Tugas Akhir yang penulis anggap masih terdapat kekurangan ini, dapat memberikan manfaat bagi para pembaca sekalian.

Serpong, Oktober 1997

Penulis

DAFTAR ISI

Lembar Pengesahan	i
Abstrak	ii
Kata Pengantar	iii
Daftar Isi	v
Bab I. Pendahuluan	1
1.1. Latar Belakang Permasalahan	1
1.2. Batasan Masalah	2
1.3. Tujuan	2
1.4. Sistematika Penulisan	2
Bab II. Teori Dasar	4
2.1. Pengubah Analog Ke Digital	4
2.2. Jalur Paralel Pada Komputer Pribadi	10
2.3. Perangkat Lunak	11
Bab III. Perancangan Alat	21
3.1. Perancangan Dan Blok Diagram	21
3.2. Perancangan Perangkat Keras (<i>Hardware</i>)	22
3.2.1. Sensor Suhu	22
3.2.2. Rangkaian Konversi Analog Ke Digital (<i>ADC</i>)	23
3.2.2.1. <i>ADC</i> 0804	23
3.2.2.2. Rangkaian Tegangan Referensi	25
3.2.2.3. Penghubung Jalur Paralel Komputer	26

3.2.2.4. Rangkaian Catu Daya	27
3.3 Perancangan Perangkat Lunak	28
Bab IV Pengujian Dan Analisa Alat	31
Bab V Kesimpulan	36
Daftar Pustaka	37
Lampiran	

DAFTAR GAMBAR

1. Gambar 2.1. Blok diagram dari sistem pengukuran	4
2. Gambar 2.2. Pengukuran, penggandaan, pengubah ADC	5
3. Gambar 2.3. ADC jenis SAR	8
4. Gambar 2.4. ADC jenis flash	9
5. Gambar 3.1. Blok diagram pengatur suhu	22
6. Gambar 3.2. Rangkaian sensor suhu LM35	23
7. Gambar 3.3. Rangkaian pengubah analog ke digital ADC 0804	25
8. Gambar 3.4. Rangkaian tegangan referensi	26
9. Gambar 3.5. Hubungan jalur paralel komputer dengan ADC 0804	26
10. Gambar 3.6. Rangkaian catu daya	28
11. Gambar 3.7. Diagram alur tahapan pembuatan program	30

DAFTAR TABEL

1. Tabel 2.1. Nomor dan nama sinyal LPT (Jalur paralel)	10
2. Tabel 2.2. Daftar alamat port	14
3. Tabel 2.3. Daftar Interrupt	18
4. Tabel 4.1. Data hasil pengukuran suhu ruangan	31
5. Tabel 4.2. Hasil konversi bilangan biner ke bilangan decimal	31
6. Tabel 4.3. Hasil konversi ADC0804	32
7. Tabel 4.4 Perbandingan hasil ukur dan perhitungan	33
8. Tabel 4.5. Kesalahan atau perbedaan suhu thermometer dengan LM35	33
9. Tabel 4.6. Perbedaan suhu antara LM35 dengan hasil baca ADC0804	34

BAB I

PENDAHULUAN

1.1. Latar Belakang Permasalahan

Dewasa ini perkembangan penggunaan komputer di segala bidang meningkat dengan pesat. Saat ini komputer sudah bukan merupakan barang mewah lagi, tetapi merupakan suatu alat yang sangat dibutuhkan untuk membantu memecahkan masalah baik di kantor maupun di rumah.

Komputer pribadi atau yang lebih dikenal dengan sebutan *PC* (*Personal Computer*), saat ini memiliki kemampuan untuk memainkan suara (*audio*) dan gambar (*video*) dengan baik.

Dari kemampuan yang sudah dimiliki oleh *PC* , timbul keinginan penulis untuk menambahkan kemampuan *PC* tersebut dengan pengukur / pengatur suhu melalui jalur paralel yang telah tersedia pada setiap *PC*.

Pengukuran suhu dapat dilakukan secara analog ataupun secara digital. Pengukuran suhu dengan menggunakan komputer dapat dilakukan dengan mudah apabila data suhu yang akan diukur, disajikan dalam format digital. Data yang diperoleh akan dengan mudah dapat diolah dan disimpan dalam komputer.

1.2. Batasan Masalah

Pokok permasalahan yang dibahas adalah untuk menjelaskan cara kerja pengukur suhu yang akan dihubungkan dengan komputer pribadi melalui jalur paralel, dan pengaturan suhu dilakukan dengan program yang menetap di *memory*. Dalam Tugas Akhir ini, rancangan yang dibuat adalah alat ukur / pengatur suhu yang bekerja dalam ruangan dengan suhu dari 0° Celcius hingga 150° Celcius. Komponen yang digunakan adalah IC LM 35, ADC 0804. Rangkaian dihubungkan dengan komputer melalui jalur paralel.

1.3. Tujuan

Tujuan penyusunan Tugas Akhir ini adalah membuat alat pengatur suhu yang dilakukan oleh komputer dengan menggunakan program yang dibuat menetap di *memori*. Menetap di *memori* mempunyai tujuan agar pengukuran tidak mengganggu pekerjaan lain yang sedang dilakukan oleh pengguna komputer.

1.4. Sistematika Penulisan

Untuk memudahkan serta mengetahui materi yang akan dibahas, makalah Tugas Akhir ini dibagi dalam 4 (empat) bab yaitu :

- Bab satu berisi latar belakang, batasan masalah, tujuan, sistematika penulisan.

- Bab dua berisi uraian-uraian mengenai komponen-komponen yang digunakan dan jalur paralel dari komputer, serta pengertian dari program menetap di *memori*.
- Bab tiga berisi penjelasan mengenai diagram blok dari sistem perancangan instrumen secara keseluruhan.
- Bab empat berisi hasil-hasil pengujian alat serta ketepatan alat dalam mengukur suhu ruangan.
- Akhir dari penulisan makalah Tugas Akhir ini adalah kesimpulan yang diambil dari unjuk kerja sistem secara menyeluruh.

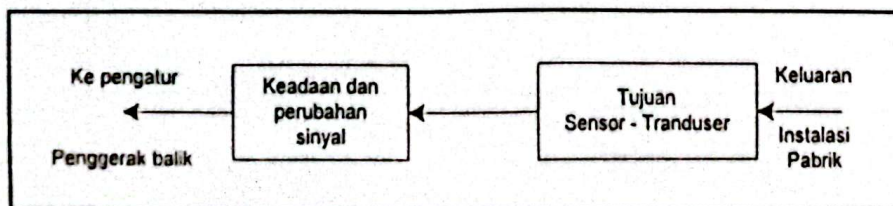
BAB II

TEORI DASAR

II.1. Pengubah Analog Ke Digital (ADC)

Dalam suatu sistem kontrol umpan balik, tanggapan instalasi diukur dan dibandingkan dengan acuan dan kesalahan secara otomatis digunakan untuk mengontrol instalasi. Dengan demikian sistem pengukuran merupakan komponen yang penting dalam sistem kontrol umpan balik, ini merupakan hubungan yang vital antara instalasi dan pengontrol (*controller*). Pengukuran diperlukan dalam banyak aplikasi rekayasa dan proses pengukuran harus otomatis, walaupun dalam aplikasi sistem kontrol.

Sistem pengukuran tipikal yang terdiri atas unit sensor - transduser atau lebih dan penataan (*devices*) pengatur sinyal yang saling berkaitan, dimana *ADC* merupakan metoda sinyal, kendatipun pencatatan data merupakan fungsi tipikal. Ini diperlihatkan secara blok diagram pada gambar 2.1.



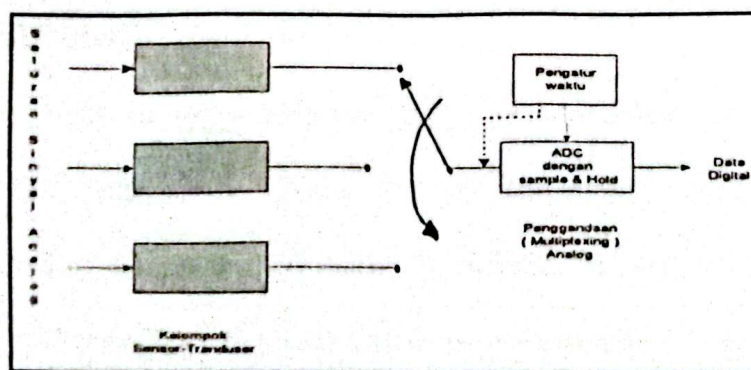
Gambar 2.1. Blok diagram dari sistem pengukuran

Piranti-piranti sensor-transduser sebagian besar merupakan komponen-komponen analog yang membangkitkan sinyal-sinyal analog, kendatipun transduser digital menjadi semakin populer dalam aplikasi kontrol digital. Jika

transduser analog digunakan, maka *ADC* harus digunakan untuk mengubah sinyal-sinyal analog pada titik-titik waktu diskrit. Sekali suatu harga di-*sampling*, maka dibuat kode menjadi representasi digital seperti kode biner langsung, kode kelabu (*grey*), kode biner decimal (*BCD*) atau kode standar Amerika untuk pertukaran informasi (*ASCII*). Perubahan pada sinyal analog dengan ragam transiennya, tidak seharusnya mempengaruhi proses *ADC* ini.

Operasi cuplik (*sample*) dan simpan (*hold*) diperlukan selama periode *sampling* masing-masing. Contohnya, nilai suatu sinyal analog dideteksi pada awal masing-masing periode *sampling* dan diasumsikan konstan selama periode *sampling* keseluruhan. Sebenarnya ini merupakan operasi penyimpanan tingkat nol (*zero order hold*).

Operasi penggandaan (*multiplexing*), *sampling* pendigitalisasian (*digitizing*), harus disinkronkan dengan benar dibawah kontrol piranti pengatur waktu yang akurat (*clock*), untuk sistem kontrol yang bekerja dengan normal. Prosedur ini diperlihatkan pada gambar 2.2.



Gambar 2.2. Pengukuran, penggandaan, pengubah *ADC*

Seluruh piranti yang membantu prosedur pengukuran pengaturan, dapat ditafsirkan sebagai komponen-komponen dari sistem pengukuran. Pemilihan komponen yang tersedia untuk aplikasi khusus atau disain komponen baru, harus tergantung sekali pada spesifikasi kerja (*performance*) untuk komponen-komponen ini.

ADC standar yang biasa dipakai adalah untuk sinyal analog adalah 0 - 5 Volt. Rangkaian ini akan menghasilkan keluaran digital dengan lebar bit yang telah ditentukan sesuai dengan kemampuannya, sebagai hasil konversi sinyal analog yang diukur.

Pemilihan suatu rangkaian *ADC* biasanya harus memperhitungkan beberapa pertimbangan sebagai berikut :

- Kecepatan konversi
- Ketelitian
- Biaya

Rangkaian *ADC* dapat dikelompokkan menjadi dua tipe utama, yaitu :

1. Jerat Tertutup (*close loop*)

Jerat ini adalah rangkaian *ADC* yang menerapkan umpan balik didalam rangkaiannya. Contoh *ADC* ini adalah *ADC* dengan jenis penghitung tunggal (*single counter*), *tracking* dan *SAR* (*Successive Approximation Register* = jenis pendekatan berturut-turut). Yang banyak digunakan adalah *ADC* jenis *SAR*, karena biasanya dibuat agar kompatibel dengan mikroprosesor, kecepatannya cukup tinggi dan praktis untuk jumlah bit keluaran yang banyak, sehingga dapat mempunyai resolusi yang

tinggi pula. Kecepatan pengubahannya tidak bergantung pada tegangan analognya, karena dapat diatur dengan program (*software*).

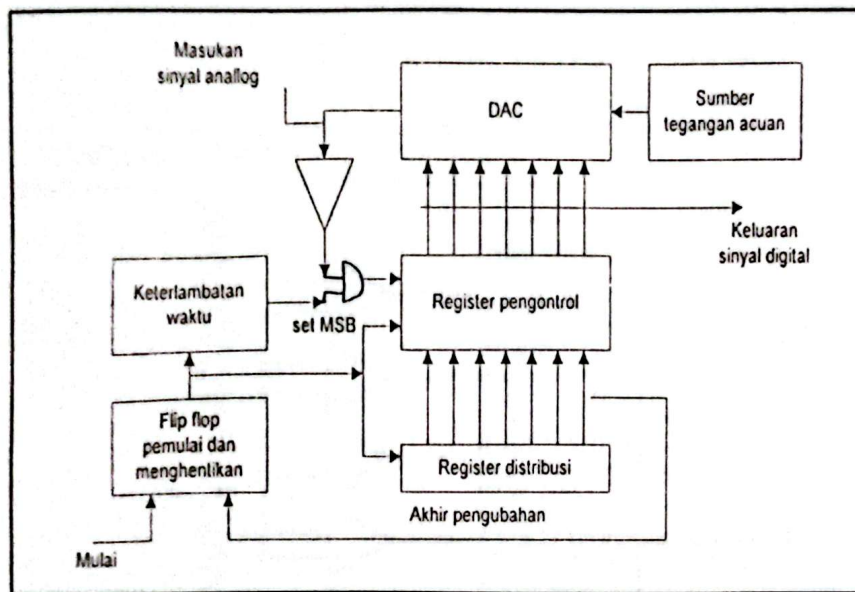
ADC jenis *SAR* membandingkan masukan analog terhadap sebuah tegangan acuan (tegangan referensi) konversi digital ke analog yang berulang-ulang dibagi menjadi dua bagian.

Diagram blok yang umum pada gambar 2.3. memperlihatkan pengubah jenis *SAR*. Konversi ini menggunakan sebuah register pengontrol digital yang mampu membukakan masukan "1" dan masukan "0" yaitu sebuah pengubah digital ke analog beserta sumber tegangan acuan, sebuah rangkaian pembanding, sebuah lup pengontrol waktu, dan register distribusi. Register distribusi menyerupai sebuah pencacah melingkar (*ring counter*) dengan sebuah angka "1" yang bersikulasi di dalamnya guna menentukan langkah mana yang berlangsung.

Pada permulaan siklus pengubahan, kedua register pengontrol dan register distribusi diset dengan angka "1" didalam bit yang paling berarti atau *MSB* (*most significant bit*) dan "0" di dalam semua bit yang kurang berarti atau *LSB* (*least significant bit*). Dengan demikian register distribusi mencatat bahwa siklus telah dimulai dan bahwa proses membacanya menyebabkan suatu tegangan keluaran pada bagian pengubah digital ke analog sebesar setengah dari tegangan referensi.

Pada saat yang sama, sebuah pulsa memasuki susunan pengatur waktu keterlambatan. Sementara *DAC* dan pembanding telah diam, pulsa yang terlambat ini dimasukan ke gerbang bersama keluaran pembanding.

Bila bit paling berarti berikutnya diset didalam register pengontrol melalui pengatur waktu, bit paling berarti *MSB* bisa tetap dalam keadaan "1" ataupun kembali ke keadaan "0", bergantung pada keluaran pembanding. Angka tunggal "1" didalam register distribusi digeser ke posisi berikutnya dan mengawasi jumlah perbandingan yang dilakukan.

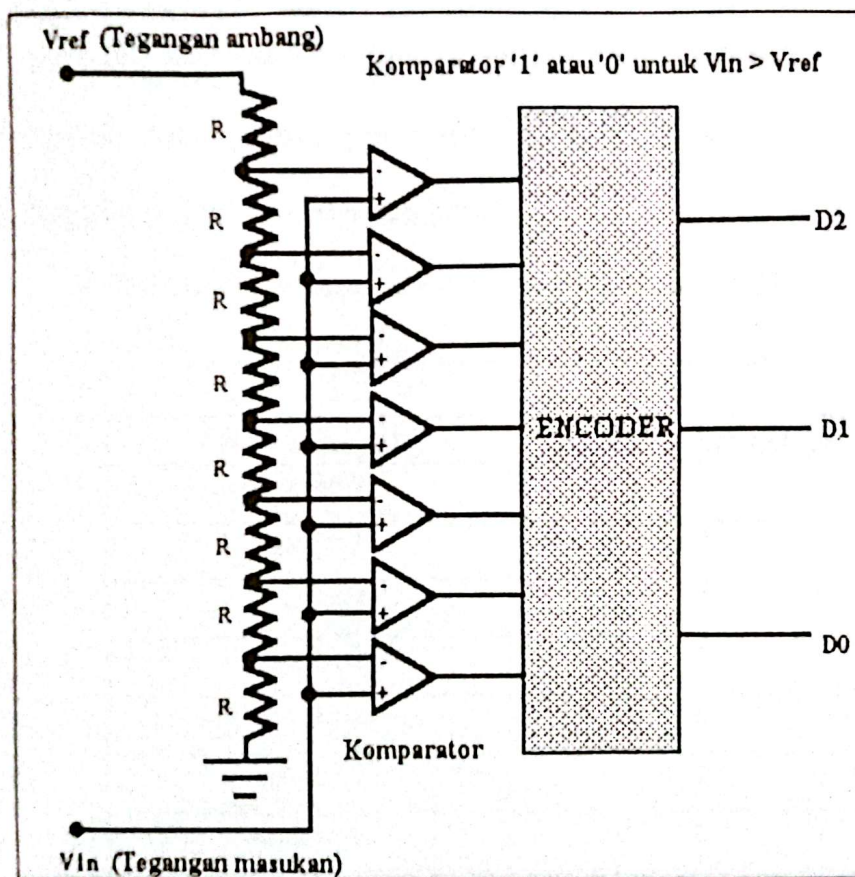


Gambar 2.3. *ADC* jenis *SAR*

2. Jerat Terbuka (*open loop*)

Rangkaian pada gambar 2.4. di bawah adalah *ADC* jenis *flash* yang merupakan *ADC* yang paling sederhana, dan waktu yang dibutuhkan untuk melakukan suatu konversi yang sangat cepat (≤ 100 ns). *ADC* ini terdiri dari beberapa komparator beda tegangan, yang dilengkapi dengan resistor-resistor pembagi tegangan, yang memberikan tegangan ambang (*threshold*) untuk setiap komparator. Jika masukan tegangan lebih besar dari tegangan

ambang tersebut, maka keluaran komparator akan menjadi tinggi sampai semua komparator memberikan keluarannya, yang kemudian masuk ke rangkaian encoder untuk diterjemahkan menjadi bit biner yang sesuai. ADC jenis ini kurang praktis untuk jumlah bit yang besar, karena membutuhkan komparator sebanyak ($n-1$ buah komparator).



Gambar 2.4. ADC jenis *Flash*

11.2. Jalur Paralel Pada Komputer Pribadi (PC)

Nama lain dari paralel *port* adalah printer *port*, karena memang ia dirancang untuk melayani pencetak paralel. Nama *register*, nama sinyal dan sifat sinyal, semuanya disesuaikan dengan pencetak. Data *port* yang paralel 8 bit, yang seharusnya mampu bekerja dua arah (mengirim dan menerima sinyal), dibuat permanen 1 arah saja. Data *port* hanya dipakai untuk mengirim data ke pencetak. Namun sekarang telah dikeluarkan paralel port dua arah, tetapi untuk menjaga kompatibilitas alat pengatur suhu ini, dianggap bahwa paralel port masih satu arah, sehingga alat dapat dipergunakan pada setiap komputer pribadi.

Berikut adalah tabel berisikan *pin assignment* dari soket DB-25 :

Tabel 2.1. Nomor dan nama sinyal LPT (jalur paralel)

Nama Sinyal	Pin ke (soket DB-25)
<i>data bit 0</i>	2
<i>data bit 1</i>	3
<i>data bit 2</i>	4
<i>data bit 3</i>	5
<i>data bit 4</i>	6
<i>data bit 5</i>	7
<i>data bit 6</i>	8
<i>data bit 7</i>	9
<i>strobe</i>	1
<i>autofeed</i>	14
<i>init</i>	16
<i>select in</i>	17

<i>error</i>	15
<i>select</i>	13
<i>paper end</i>	12
<i>acknowledge</i>	10
<i>busy</i>	11
<i>ground</i>	18..25

Alamat dari paralel *port* yang dipergunakan adalah 378 Hexadecimal atau 888 dalam sistem bilangan decimal. Untuk pengiriman data ke *data port*, dapat langsung mengirimkannya ke alamat tersebut. Untuk membaca masukan pada kaki pena yang lain, alamat yang dipergunakan adalah :

Alamat dasar + 1, untuk sinyal : *busy, ack, paper end, select, error*.

Alamat dasar + 2, untuk sinyal : *irq enable, select input, initialize, autofeed, strobe*.

II.3. Perangkat Lunak (*Software*)

Perangkat lunak pada Tugas Akhir ini menggunakan bahasa program PASCAL melalui *compiler* TURBO PASCAL. Program bersifat *residen*.

Yang dimaksud dengan *residen* adalah sekali program dijalankan, maka program akan terus menetap di memori komputer. Cara untuk menghilangkan program dari memori adalah dengan melakukan *warm boot* atau dengan mematikan komputer. Selain itu juga dapat membuat program menjadi program *TSR (Terminate and Stay Resident)*. Yang dimaksud

dengan *TSR* adalah program setelah aktif di memori, dapat juga dihentikan dengan menggunakan tombol yang telah ditentukan, tanpa harus mematikan atau melakukan *warm boot*. Dalam sebuah *PC* memiliki dua jenis memori yaitu *RAM* dan *ROM*. *RAM* (*Random Access Memory*) adalah jenis memori yang dapat dibaca dan sekaligus ditulisi. *RAM* menyimpan informasi hanya sementara, karena bila tegangan supply dimatikan maka semua isi didalamnya akan hilang. Berbeda dengan *RAM*, *ROM* (*Read Only Memory*) adalah jenis memori yang hanya dapat diisi sekali saja dan selanjutnya hanya dibaca, dan walaupun tegangan supply dimatikan isi di dalam *ROM* tetap ada. Sifat-sifat dari sebuah program *residen* adalah menempati suatu area atau lokasi tertentu di memori, letak dari area tersebut adalah sesudah sistem operasi, dapat bekerja tanpa mengganggu program lain, menempati memori komputer selama komputer hidup. Prinsip kerja program *residen* adalah menggunakan *sistem interrupt*.

Langkah-langkah pembuatan program *residen* adalah :

- Menentukan variabel-variabel yang dibutuhkan oleh suatu program *residen*.
- Menentukan ukuran *Stack* yang dibutuhkan oleh program *residen*.
Ukuran *Stack* dapat ditentukan dengan cara memperkirakan besarnya tanda-tanda pengenalan atau variabel program *residen*.

- Menentukan ukuran memori yang dibutuhkan oleh program *residen*. Bila pembuatan program *residen* memakai perintah *keep*, besarnya memori akan diatur secara otomatis oleh TURBO PASCAL. Tetapi bila menggunakan Interrupt 27h dan Interrupt 21h service 31h, besarnya memori harus ditentukan sendiri.
- Mengubah atau membelokkan vektor interrupt tertentu pada tabel vektor interrupt ke vektor interrupt program *residen*. Bila diperlukan vektor interrupt yang asli diamankan atau disimpan terlebih dahulu.

Data diperoleh dari jalur paralel komputer yang dikenal sebagai port, pengertian port adalah suatu saluran yang menghubungkan mikroprosesor dengan peralatan luar. Dengan adanya port dapat dimungkinkan pengaturan-pengaturan peralatan luar secara *software*, misalnya pengaksesan port untuk mengatur *printer*, adapter video, komunikasi antar komputer, dan sebagainya. Untuk dapat menggunakan port, maka port-port tersebut diberi nomor atau alamat tertentu. Setiap alamat dihubungkan dengan peralatan yang berbeda-beda, misalnya port dengan alamat 3B0H – 3BBHexadecimal yang merupakan port untuk mengakses adapter video monokrom, port 3BCH-3BFH untuk adapter printer yang pertama (LPT1). Berikut ini merupakan daftar alamat port pada komputer IBM PC :

Tabel 2.2. Daftar Alamat Port

ALAMAT PORT (Hexadecimal)	Tujuan
0000-000F	<i>Direct memory access (DMA) Controller 8237A</i>
0010	<i>Angka test dari pabrik</i>
0020-0021	<i>Programmable interrupt controller 8259</i>
0040-0043	<i>Programmable timer control 8253</i>
0060-0063	<i>Programmable peripheral interface 8255</i>
0080-0083	<i>DMA page register</i>
00A0-00AF	<i>Non maskable interrupt register</i>
00E0-00FF	<i>Numeric coprocessor usage</i>
01F0-01F8	<i>AT fixed disk interface</i>
0200-020F	<i>Game controller</i>
0210-0217	<i>Expansion unit</i>
0220-024F	<i>Reserved</i>
0278-027F	<i>LPT3</i>
02E8-02EF	<i>Com4</i>
02F0-02F7	<i>Reserved</i>
0300-031F	<i>Prototype card</i>
0320-032F	<i>XT fixed disk interface</i>
0378-037F	<i>LPT2</i>
0380-038C	<i>SDLC / Secondary B-Sync interface</i>
03A0-03A9	<i>Primary B-Sync interface</i>
03B0-03BB	<i>Monokrom display</i>
03BC-03BF	<i>LPT1</i>
03C0-03CF	<i>EGA display control</i>
03D0-03DF	<i>Color graphic adapter</i>
03E0-03E7	<i>Reserved</i>
	<i>Bersambung...</i>

03E8-03EF	<i>Com3</i>
03F0-03F7	<i>Floppy disk controller</i>
03F8-03FF	<i>Com1</i>

Dalam pemrograman residen banyak dipergunakan sistem interrupt. Suatu mikroprocessor (CPU) didalam melaksanakan tugasnya selalu dibantu oleh perangkat keras atau peralatan penunjang, misalnya perangkat untuk pengaturan layar (adapter video), perangkat untuk pengaturan printer, disk drive, dsb. Supaya perangkat-perangkat tersebut bekerja dengan baik, maka *CPU* melakukan suatu rutin pelayanan pemrosesan terhadap perangkat tersebut. Semakin banyak perangkat yang terpasang pada komputer, semakin banyak pula rutin pelayanan yang harus diberikan oleh *CPU*. Untuk mengetahui bahwa suatu perangkat memerlukan pelayanan dari *CPU*, perangkat tersebut mengirimkan sinyal-sinyal tertentu ke *CPU* sebagai tanda bahwa ia ingin dilayani. Bila *CPU* menerima sinyal tersebut, barulah *CPU* melakukan pelayanan pemrosesan.

Berdasarkan uraian di atas, maka yang dimaksud dengan interrupt (interupsi) adalah suatu proses pemberian sinyal ke *CPU* yang dilakukan oleh perangkat keras dengan maksud supaya *CPU* menunda pekerjaan yang dilakukannya, dan melakukan pekerjaan yang diminta oleh perangkat keras tersebut (rutin pelayanan interrupt / *interrupt service routine*). Bila *CPU* selesai melakukan pekerjaan tersebut, ia akan

melanjutkan kembali pekerjaan semula (yang tertunda tadi. Walaupun demikian bukan berarti proses interrupt bisa terjadi setiap saat, karena kerja dari interupsi berdasarkan prioritas. Hanya prioritas tertinggi-lah yang akan dilayani terlebih dahulu. Dengan adanya sistem interrupt ini, kerja CPU akan jauh lebih efisien dan efektif. Untuk dapat melaksanakan operasi interrupt CPU dilengkapi dengan peralatan penunjang yang disebut dengan interrupt controller. Pada komputer IBM PC terdapat 256 macam interrupt yang dapat dikelompokkan menjadi 7 katagori, yaitu mikroprosesor interrupt, hardware interrupt, software interrupt, DOS interrupt, BASIC interrupt, address interrupt dan general use interrupt. Keterangan dari masing-masing katagori interupsi tersebut adalah sebagai berikut :

- **Mikroprosesor Interrupt**

Mikroprosesor interrupt merupakan interrupt khusus yang disediakan untuk operasi yang dilakukan oleh mikroprosesor. Ada 5 buah interrupt yang termasuk dalam mikroprosesor interrupt, yaitu interrupt 00H, 01H, 02H, 03H dan 04H. 4 buah interupsi diantaranya (interrupt 00H, 01H, 03H, dan 04H) dihasilkan oleh mikroprosesor itu sendiri. Sedangkan interrupt lainnya (interrupt 02H diaktifkan oleh sinyal yang dihasilkan oleh peralatan external).

- **Hardware Interrupt**

Hardware interrupt adalah suatu proses interrupt terhadap CPU yang disebabkan oleh kerja perangkat keras (Hardware), misalnya timer

interrupt (Int 8H) dan keyboard interrupt (Int 9H) yang termasuk ke dalam hardware interrupt ini adalah interrupt 02H, 08H, 09H dan 0BH-0FH.

- **Software Interrupt**

Software interrupt adalah proses interrupt yang disebabkan oleh perangkat lunak, misalnya DOS Interrupt (Int 21H) maupun interrupt lain yang dibuat (diprogram oleh user)

- **DOS Interrupt**

DOS interrupt baru tersedia pada saat DOS baru mulai digunakan. Banyak program-program aplikasi yang menggunakan interrupt ini untuk menangani berbagai macam operasi. Yang termasuk dalam DOS interrupt adalah interrupt 20H-FFH.

- **BASIC Interrupt**

BASIC interrupt merupakan interrupt yang khusus digunakan untuk BASIC. Yang termasuk dalam BASIC interrupt adalah interrupt 80H-F0H.

- **Address Interrupt**

Address interrupt merupakan interrupt yang dikhususkan untuk menangani tabel-tabel tertentu seperti kontrol video, parameter disk dan karakter grafik (128-255). Yang termasuk dalam address interrupt adalah interrupt 1DH-1FH.

- General Use Interrupt

General use interrupt merupakan interrupt cadangan untuk keperluan umum yang ditujukan untuk program-program aplikasi.

Secara keseluruhan, ke-256 interrupt tersebut dapat dilihat pada tabel 2.3.

Daftar Interrupt.

Tabel 2.3. Daftar Interrupt.

Nomor Interrupt (Hexadecimal)	Nama Interrupt Service Routine
Int 00	<i>Divide by zero</i>
Int 01	<i>Single step execution</i>
Int 02	<i>Non maskable interrupt (NMI)</i>
Int 03	<i>Breakpoint</i>
Int 04	<i>Aritmathic overflow</i>
Int 05	<i>Print screen</i>
Int 06	<i>Reserved</i>
Int 07	<i>Reserved</i>
Int 08	<i>Clock tick (Timer)</i>
Int 09	<i>Keyboard</i>
Int 0A	<i>IO channel</i>
Int 0B	<i>Com1</i>
Int 0C	<i>Com2</i>
Int 0D	<i>Fixed disk / hard disk</i>
Int 0E	<i>Floppy disk</i>
Int 0F	<i>LPT1</i>
Int 10	<i>Video display</i>
Int 11	<i>Equipment flag</i>

Int 12	<i>Memory size</i>
Int 13	<i>Disk service</i>
Int 14	<i>Communication service RS-232</i>
Int 15	<i>Cassette service</i>
Int 16	<i>Keyboard service (keyboard IO)</i>
Int 17	<i>Printer service</i>
Int 18	<i>ROM BASIC call</i>
Int 19	<i>Bootstrap loader (Warm boot)</i>
Int 1A	<i>Time and date</i>
Int 1B	<i>Keyboard break (control break)</i>
Int 1C	<i>Clock tick</i>
Int 1D	<i>Video control</i>
Int 1E	<i>Disk parameter table</i>
Int 1F	<i>Graphic character (ASCII) 128-255</i>
Int 20	<i>DOS program terminate</i>
Int 21	<i>DOS function call</i>
Int 22	<i>DOS terminate address</i>
Int 23	<i>DOS Ctrl Exit address</i>
Int 24	<i>DOS fatal error factor</i>
Int 25	<i>DOS absolute disk read</i>
Int 26	<i>DOS absolute disk write</i>
Int 27	<i>Terminate and stay resident</i>
Int 28 – Int 3F	<i>Reserved for DOS</i>
Int 40 – Int 5F	<i>Reserved</i>
Int 60 – Int 67	<i>Reserved for user software</i>
Int 68 – Int 7F	<i>Not used</i>
Int 80 – Int 85	<i>Reserved by BASIC</i>
Int 86 – Int F0	<i>BASIC Interpreter</i>
Int F1 – Int FF	<i>Not used</i>

Masing-masing nomor interrupt ditempatkan dalam suatu urutan tertentu dengan cara peng-alamatan CS:IP (CS adalah segmen alamat dari interrupt dan IP adalah offset alamat dari interrupt). Yang membentuk suatu tabel. Tabel ini disebut dengan tabel vektor interupsi. Tabel vektor interupsi menempati area memori pada alamat absolute 0000H – 03FFH. Sebenarnya ketujuh katagori interrupt yang telah dibahas di atas dapat digolongkan lagi menjadi 2 bagian besar berdasarkan saat tersedianya atau penempatan suatu interrupt, yaitu ROM BIOS interrupt dan DOS interrupt.

- **ROM BIOS Interrupt**

Yang dimaksud dengan ROM BIOS interrupt adalah program BIOS interrupt yang disimpan di dalam ROM. Jumlah interupsi untuk ROM BIOS adalah sebanyak 32 buah, yaitu interrupt 00H – 1FH, yang terletak mulai dari alamat absolute 0000H – 007FH.

- **DOS Interrupt**

Yang dimaksud dengan DOS Interrupt adalah program BIOS interrupt yang baru ada atau dapat digunakan setelah sistem operasi (DOS) diaktifkan. Letak dari DOS interrupt adalah mulai dari alamat absolute 0080H – 03FFH. Sebenarnya pada DOS interrupt ini, terdapat pula interrupt-interrupt lain yang disediakan untuk user, BASIC dan cadangan (kosong). Yang digunakan untuk DOS adalah nomor interupsi 20H – 27H. Sedangkan interrupt cadangan dapat dipakai untuk pemrograman sendiri (pembuatan program residen).

BAB III

PERANCANGAN ALAT

III.1. Perancangan dan Blok Diagram

Pembuatan perangkat memerlukan perancangan terlebih dahulu, agar mencapai karakteristik kerja yang diinginkan. Peralatan yang dibuat harus memakai komponen seminimal mungkin, namun tidak mengurangi fungsi kerja dari alat tersebut.

Perancangan dan pembuatan alat pengatur suhu / thermostat berbasis komputer, meliputi :

a) Perancangan perangkat keras (*hardware*) yang terdiri dari :

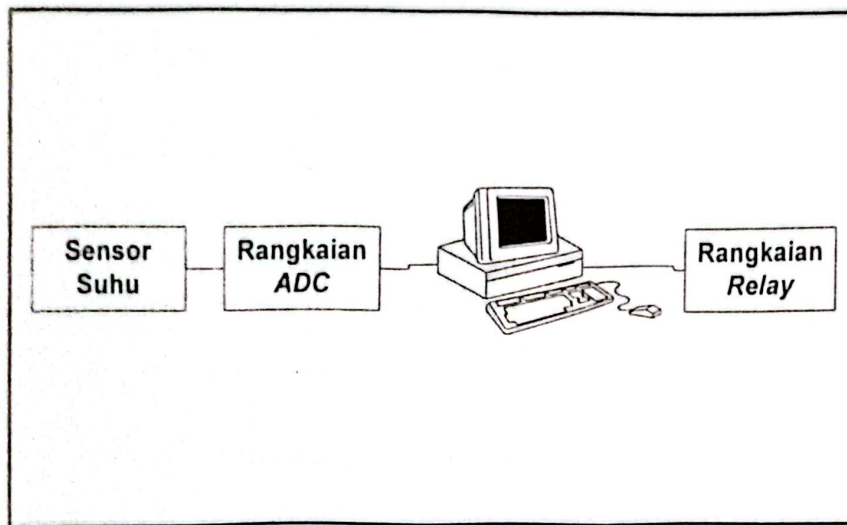
- * Sensor suhu.
- * Rangkaian konversi analog ke digital (*ADC*) yang terdiri atas:
 - ⇒ *ADC* 0804.
 - ⇒ Rangkaian tegangan referensi.
 - ⇒ Rangkaian penghubung antara jalur paralel komputer dengan *ADC* 0804.
- * Rangkaian catu daya.

b) Perancangan perangkat lunak (*Software*) :

Merupakan proses data yang dijalankan program komputer, dan mempunyai fungsi menampilkan nilai hasil konversi dan membandingkan hasil ukur dengan batasan suhu yang diinginkan. Dari hasil banding tersebut, program akan mengirimkan sinyal untuk mengaktifkan atau

mematikan *relay* yang nantinya akan dihubungkan dengan alat yang diinginkan. Misalnya pendingin udara atau pemanas ruangan dan selain itu dapat juga dihubungkan dengan sirine alarm.

Blok diagram dari alat pengatur suhu / thermostat dapat dilihat pada gambar berikut (gambar 3.1).



Gambar 3.1. Blok diagram pengatur suhu / thermostat berbasis komputer.

III.2. Perancangan Perangkat Keras (*Hardware*)

III.2.1. Sensor Suhu

Sensor suhu yang dipakai adalah sensor suhu LM 35. Batas ukur dari komponen ini adalah -55° Celcius sampai dengan 150° Celcius. Tetapi di dalam Tugas Akhir ini, perancangan yang dibuat hanya dapat mengukur suhu antara 0° Celcius sampai dengan 150° Celcius. Tegangan keluaran (V_{out}) dari komponen ini adalah penambahan 10 mV untuk setiap 1°

III 221 ADC 0804

IC ini merupakan ADC 8 bit dengan jenis ADC 0804 yang

Untuk mendapatkan operasi yang kontinu, maka pers. 2 dan pers.

tegangan acuan untuk menentukan kondisi tegangan masukan negatif, sedangkan pena 9 adalah untuk menentukan batas atas skala-penuh. Caranya adalah dengan mengeset trimpot agar menghasilkan tegangan sebesar $\frac{1}{2}$ kali dari nilai maksimum masukan yang diinginkan.

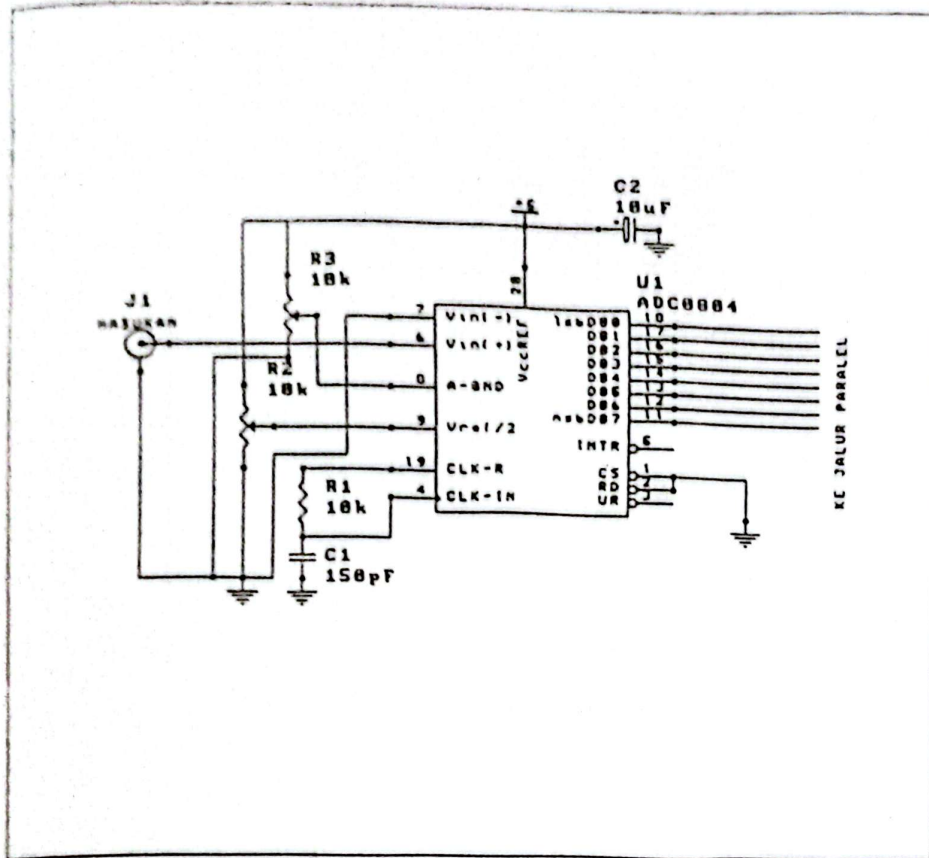
Tegangan yang merupakan sinyal analog yang berada pada pena 6 akan dikonversikan menjadi data 8 bit yang tersedia pada pena 11 (D7) sampai dengan pena 18 (D0). Data 8 bit ini mempunyai langkah bebas sebesar 255. Batas tegangan masukan adalah 0 sampai +2 Volt DC, sehingga untuk perhitungan penambahan 1 harga biner dari kenaikan tegangan masukan tersebut adalah sebesar 0,008 Volt. Data 8 bit dengan harga biner dibentuk dari tegangan 0 Volt menjadi 0000 0000 dan tegangan 2 volt menjadi 1111 1111. Perhitungan dari 8 bit tersebut diselesaikan dalam bentuk persamaan di bawah ini :

$$\text{Jumlah langkah} = 2^n - 1 \quad \text{adalah} \quad 2^8 - 1 = 255$$

dimana n adalah jumlah bit.

$$\text{Untuk perubahan 1 bit} = 2 \text{ V} / 256 = 0,008 \text{ Volt.}$$

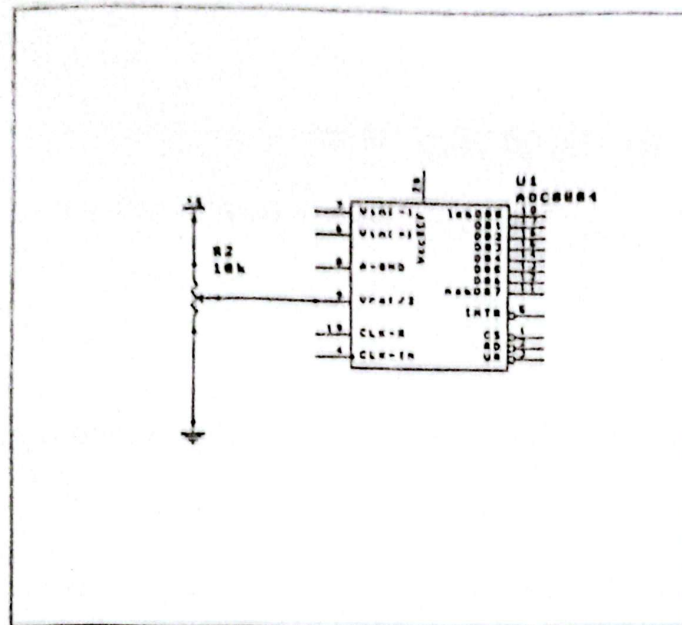
Pada halaman berikut dapat dilihat rangkaian pengubah analog ke digital.



Gambar 3.3. Rangkaian pengubah analog ke digital ADC 0804

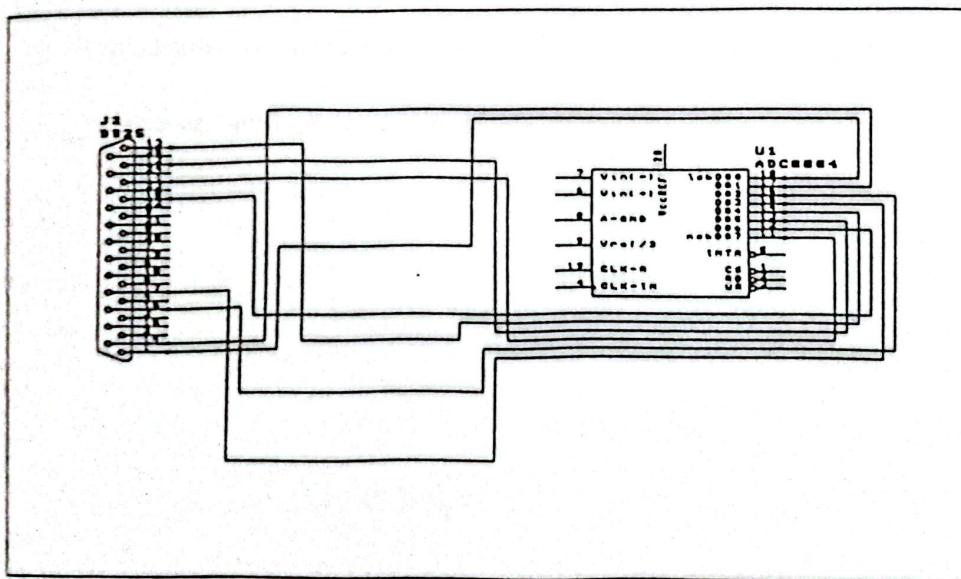
III.2.2.2. Rangkaian Tegangan Referensi

Pada pena 9 ADC 0804 memerlukan tegangan masukan sebesar $V_{ref} / 2$ yang diperoleh dari trimpot 10k Ohm. Tegangan diatur hingga mencapai tegangan yang diinginkan yaitu 1 Volt, penentuan besar tegangan tersebut karena nilai maksimum masukan yang akan diukur adalah 2 Volt. Rangkaian tegangan referensi dapat dilihat pada halaman berikut.



Gambar 3.4. Rangkaian tegangan referensi *ADC 0804*.

III.2.2.3. Penghubung Jalur Paralel Komputer Dengan *ADC 0804*

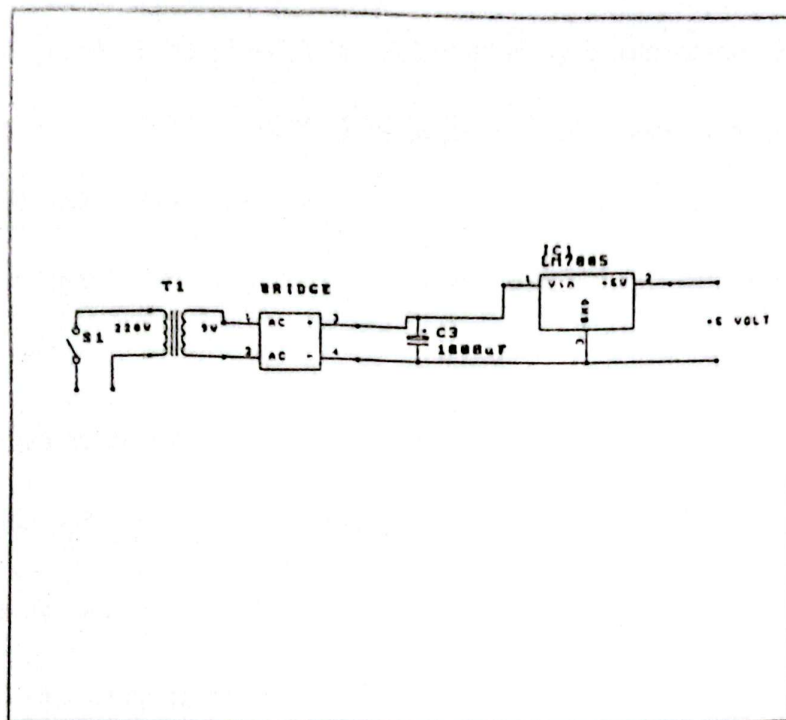


Gambar 3.5. Hubungan jalur paralel komputer dengan *ADC 0804*

Pada gambar 3.5., terlihat hubungan pena-pena dari *ADC* yang dihubungkan dengan jalur paralel dari komputer *PC*.

III.2.2.4. Rangkaian Catu Daya

Tidak ada hal khusus dalam rangkaian catu daya ini, karena seperti kebanyakan rangkaian catu daya praktis, catu daya ini memanfaatkan kemampuan regulasi dari IC regulator jenis LM7805 yang mempunyai kemampuan untuk menjaga tegangan keluaran pada batas 5 Volt. Sumber tegangan yang diperlukan dalam merancang alat pengukur ini adalah sebesar 5 Volt. Penggunaan sumber tegangan luar bertujuan untuk tidak membebani daya yang dikeluarkan dari komputer. Gambar rangkaian catu daya dapat dilihat pada halaman berikut.



Gambar 3.6. Rangkaian catu daya

III.3. Perancangan Perangkat Lunak

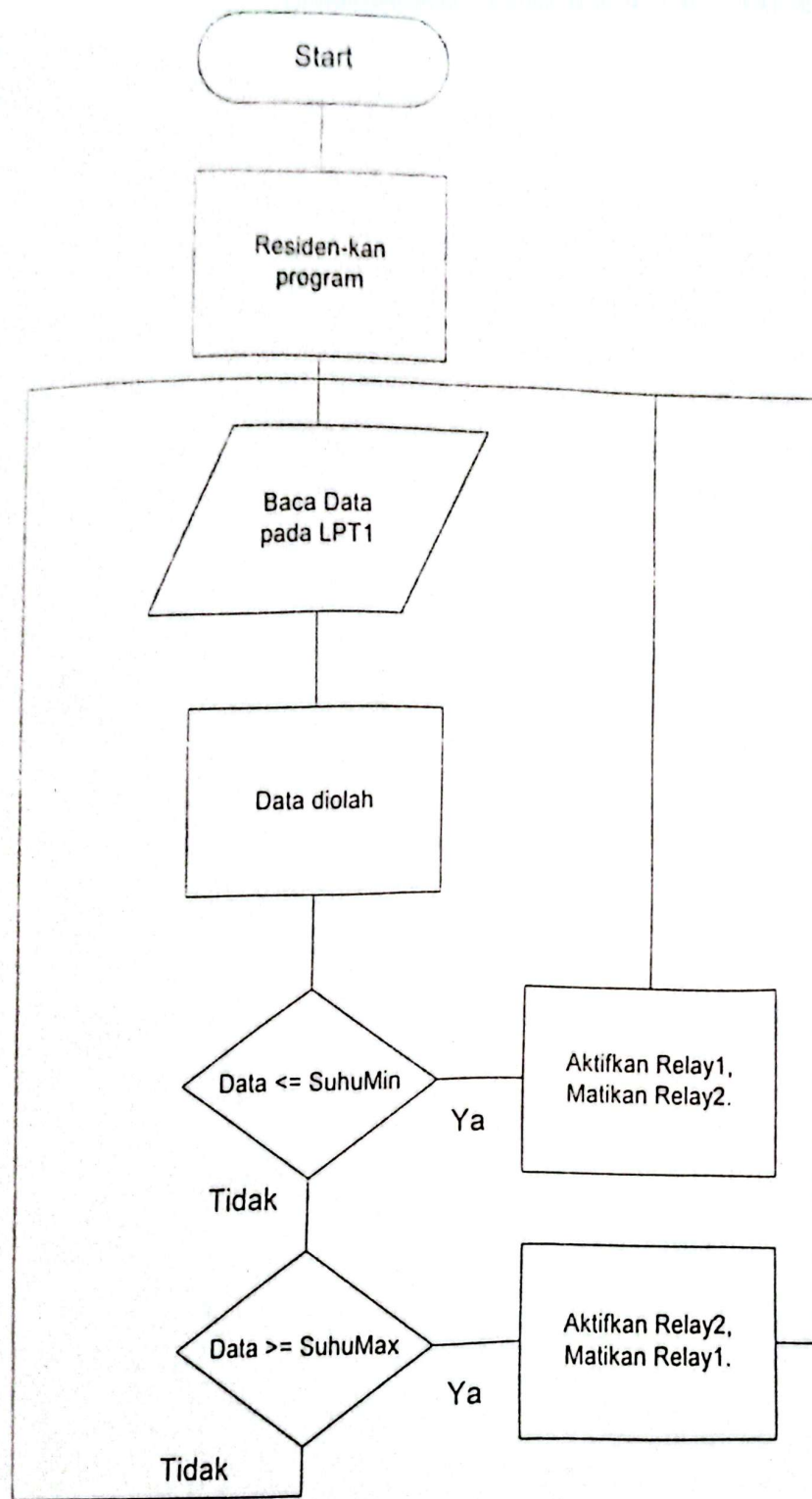
Perangkat lunak ini adalah suatu program khusus untuk memproses data yang dihasilkan oleh perangkat keras. Perangkat keras yang dimaksud adalah ADC 0804. Pembacaan data dilakukan melalui pembacaan isi alamat jalur paralel. Pada komputer ini, penulis menggunakan jalur paralel LPT1:, dengan alamat 888Decimal atau 378Hexadecimal. Hasil pembacaan dari alamat tersebut harus diolah terlebih dahulu, karena ada bit yang harus di-invers terlebih dahulu. Setelah hasil akhir dari pengolahan data sudah selesai, maka program akan membandingkan suhu ruangan dengan batas suhu yang diminta. Dengan batas tersebut, program akan menjaga suhu ruangan dengan mengaktifkan

relay1 bila keadaan suhu ruangan mencapai $\leq$ suhu_minimum, dan program akan mengaktifkan relay2 bila keadaan suhu ruangan mencapai $\geq$ suhu maksimum. Program yang berjalan ini akan aktif terus pada memori komputer (*residen*).

Dalam pembuatan program komputer ini, secara umum terdiri dari tahapan:

- Mengaktifkan program pada memori komputer
- Pembacaan data pada jalur paralel
- Pengolahan data
- Membandingkan data
- Pencetakan hasil data.

Dari tahapan-tahapan tersebut di atas, maka dapat dibuat diagram alur seperti pada gambar di halaman berikut.



Gambar 3.7. Diagram Alur tahapan pembuatan program.

Bab IV

Pengujian Dan Analisa Alat

IV.1. Pengujian Fungsi Alat

Untuk mengetahui apakah alat ini bekerja dengan baik atau tidak, maka dilakukan pengujian fungsi alat dengan melakukan beberapa kali pengukuran terhadap suhu ruangan dengan dibantu alat pengering rambut, agar suhu dapat dirubah-rubah. Selain itu hasil konversi dari keluaran sensor suhu dapat pula dilihat pada tampilan led dan tampilan pada layar komputer.

Pada pembanding sensor suhu dipergunakan sebuah thermometer suhu ruangan buatan German dengan merk : TFA yang menggunakan sistem kerja air raksa. Batas pengukuran thermometer tersebut adalah 0 s/d 50 derajat Celcius. Pengukuran tegangan pada keluaran LM35 digunakan multimeter digital merk Okano dengan batas pengukuran pada posisi 2Volt.

IV.2. Hasil Pengukuran

Pengukuran terhadap ruangan dilakukan sebanyak 8 kali, dengan hasil sebagai berikut :

Tabel 4.1. Data hasil pengukuran suhu ruangan

No.	Thermometer TtA	Vout ICLM35	Keluaran dari ADC0804
1	26,5	0,263	00100010
2	30	0,305	00101000
3	31	0,310	00101001
4	33	0,325	00101010
5	33,5	0,330	00101011
6	34	0,338	00101100
7	34,5	0,343	00101101
8	36	0,352	00101110

Pada perhitungan keluaran ADC0804, bilangan biner tersebut harus diubah menjadi bilangan decimal terlebih dahulu. Dari hasil perhitungan diperoleh :

Tabel 4.2. Hasil konversi Bilangan Biner ke Bilangan Decimal

No.	Bilangan Biner	Bilangan Decimal
1	00100010	34
2	00101000	40
3	00101001	41
4	00101010	42
5	00101011	43
6	00101100	44
7	00101101	45
8	00101110	46

Untuk memperoleh nilai suhu yang diukur, maka harus dilakukan perhitungan sebagai berikut :

Output *ADC0804* dibagi dengan bilangan 255 (2^n-1), setelah itu dikalikan dengan 2 Volt (*Vref* dikali dengan 2). *Vref* ditentukan sebesar 1 Volt, agar pengukuran dapat dilakukan hingga 2 Volt.

Sebagai contoh :

Hasil output *ADC 0804*= 34 Decimal.

Maka nilai yang diukur adalah : $(34 / 255) \times 2 \text{ Volt} = 0,266 \text{ Volt}$.

Karena 1 derajat Celcius = 10 mV, maka berarti suhu ruangan yang diukur adalah 26,6 derajat Celcius. Dari perhitungan tersebut diperoleh tabel sebagai berikut :

Tabel 4.3. Hasil konversi *ADC0804*

No.	Output <i>ADC0804</i>	Hasil perhitungan (V)	Hasil perhitungan (°C)
1	34	0,266	26,6
2	40	0,313	31,3
3	41	0,321	32,1
4	42	0,329	32,9
5	43	0,337	33,7
6	44	0,345	34,5
7	45	0,352	35,2
8	46	0,360	36,0

Tabel 4.4. Perbandingan hasil ukur dan Perhitungan (°C)

No.	Thermometer T1A	LM35	ADC0804
1	26,5	26,3	26,6
2	30	30,5	31,3
3	31	31	32,1
4	33	32,5	32,9
5	33,5	33	33,7
6	34	33,8	34,5
7	34,5	34,3	35,2
8	36	35,2	36,0

Tabel 4.5. Kesalahan atau perbedaan suhu Thermometer dengan LM35

No.	Thermometer	LM35	% Kesalahan
1	26,5	26,3	0,75
2	30	30,5	1,6
3	31	31	0
4	33	32,5	1,51
5	33,5	33	1,49
6	34	33,8	0,58
7	34,5	34,3	0,57
8	36	35,2	2,22

Cara memperhitungan % kesalahan = Perbedaan suhu / suhu thermometer dikalikan 100%. Pengukur suhu yang dijadikan referensi adalah Thermometer TtA.

Kesalahan atau perbedaan ukur pada kedua sensor suhu tersebut adalah :
1.09 %.

Tabel 4.6. Perbedaan suhu antara LM35 dengan hasil baca ADC0804.

No.	LM35	ADC0804	% Kesalahan
1	26,3	26,6	1,14
2	30,5	31,3	2,62
3	31	32,1	3,54
4	32,5	32,9	1,23
5	33	33,7	2,12
6	33,8	34,5	2,07
7	34,3	35,2	2,62
8	35,2	36,0	2,27

Kesalahan rata-rata antara hasil ukur dengan menggunakan multimeter digital dibandingkan dengan ADC0804 adalah sebesar 2.20%.

BAB V

KESIMPULAN

1. Penggunaan ADC0804 mempunyai keterbatasan pada tingkat ketelitian dari pengubah sinyal tersebut sebesar $2^8 - 1 = 255$ tingkatan.
2. Kesalahan ukur rata-rata menggunakan ADC0804 adalah sebesar 2,20%. Sedangkan perbedaan hasil ukur antara LM35 dengan Thermometer adalah sebesar 1,09%.
3. Pengukuran suhu dengan menggunakan IC LM35 sangat tergantung dengan perangkat penerjemah agar dapat dibaca langsung hasil pengukurannya.

DAFTAR PUSTAKA

- 1) Paul Bergsman, "Controlling The World With Your PC", HighText Publication Inc. 1994.
- 2) Ir. Busono, "Komputer Dan Turbo Pascal", PT. Elex Media Komputindo. 1991.
- 3) Steve D.S., "Membuat Program Full Residen Dan Semi Residen", Dinastindo 1995.
- 4) Barry B. Brey, "The Intel Microprocessor", Macmilli Publishing Company 1991.
- 5) Yogyanto H.M., "Pascal Tingkat Lanjutan", Andi Offset Yogyakarta. 1989.

LAMPIRAN A

{SM 1024.0.0}

PROGRAM Adc_0804;

{Dibuat oleh Delianto (011880027) }

{Program diresidenkan dengan menggunakan interrupt 1CH}

USES DOS, CRT;

VAR Lpt_Num, Lpt_Port_Address : WORD;
 RD : INTEGER;
 Volts : REAL;
Lo_Nibble, Hi_Nibble, Waktu, min, max, x, y : BYTE;

Procedure Tetap;

Interrupt;

Begin

Lpt_Port_Address:=888;

Waktu:=waktu+1;

If Waktu>18 then

begin

Waktu:=0;

Lo_Nibble := PORT[Lpt_Port_Address + 2];

Hi_Nibble := PORT[Lpt_Port_Address + 1];

Lo_Nibble := Lo_Nibble AND \$0F;

Hi_Nibble := Hi_Nibble AND \$F0;

RD := Hi_Nibble OR Lo_Nibble;

If (RD And 128)=128 then

RD:=RD And 127

else

RD:=RD OR 128;

If (RD And 1)=1 then

RD:=RD AND 254

else

RD:=RD OR 1;

If (RD And 2)=2 then

RD:=RD AND 253

ELSE

RD:=RD OR 2;

If (RD And 8)=8 then

RD:=RD AND 247

ELSE

RD:=RD OR 8;

Volts := RD * 2 / 255*100;

```

X:=whereX;
Y:=whereY;
gotoxy(60,1);
write('Suhu = ',Volts:5:3);
gotoxy(X,Y);
If rd<min then Port[Lpt_Port_Address]:=1;
If rd>max then Port[Lpt_Port_Address]:=2;
end;
end;

```

```

Procedure Tanya;
Begin
ClrScr;
GotoXY(15,10);
Write ('Masukan suhu minimum yang diinginkan ? '); Readln (min);
GotoXY(15,11);
Write ('Masukan suhu maximum yang diinginkan ? '); Readln (max);
End;

```

```

BEGIN { non residen }
Tanya;
SetIntVec($1c,@Tetap);
keep(0);
END.

```


Absolute Maximum Ratings (Note 1)

Military/Aerospace specified devices are required.
Contact the National Semiconductor Sales Office/
Representatives for availability and specifications.

Supply Voltage (V_{CC}) (Note 3)	6.5V
Logic Control Inputs	-0.3V to +18V
All Other Input and Outputs	-0.3V to ($V_{CC} + 0.3V$)
Lead Temp. (Soldering, 10 seconds)	260°C
Plastic Package (plastic)	200°C
Plastic Package (ceramic)	200°C
Surface Mount Package	215°C
Welder Probe (50 seconds)	220°C
Welder (10 seconds)	220°C

Storage Temperature Range	-65°C to +150°C
Package Dissipation at $T_A = 25^\circ\text{C}$	875 mW
ESD Susceptibility (Note 10)	500V

Operating Ratings (Notes 1 & 2)

Temperature Range	T_{MIN} T_A T_{MAX}
ADC0801/02LJ	-55°C $\leq T_A \leq$ 125°C
ADC0801/02/03/04LCJ	-40°C $\leq T_A \leq$ 85°C
ADC0801/02/03/05LCN	-40°C $\leq T_A \leq$ 85°C
ADC0804LCN	0°C $\leq T_A \leq$ 70°C
ADC0802/03/04LCV	0°C $\leq T_A \leq$ 70°C
ADC0802/03/04LCWM	0°C $\leq T_A \leq$ 70°C
Range of V_{CC}	4.5 V_{DC} to 6.3 V_{DC}

Electrical Characteristics

Following specifications apply for $V_{CC} = 5 V_{DC}$, $T_{MIN} \leq T_A \leq T_{MAX}$ and $f_{CLK} = 640 \text{ kHz}$ unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Accuracy: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			$\pm 1/2$	LSB
Accuracy: Total Unadjusted Error (Note 8)	$V_{REF}/2 = 2.500 V_{DC}$			$\pm 1/2$	LSB
Accuracy: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			$\pm 1/2$	LSB
Accuracy: Total Unadjusted Error (Note 8)	$V_{REF}/2 = 2.500 V_{DC}$			± 1	LSB
Accuracy: Total Unadjusted Error (Note 8)	$V_{REF}/2$ -No Connection			± 1	LSB
Input Resistance (Pin 9)	ADC0801/02/03/05 ADC0804 (Note 9)	2.5 0.75	8.0 1.1		k Ω k Ω
Analog Input Voltage Range	(Note 4) $V(-)$ or $V(+)$	Gnd-0.05		$V_{CC}-0.05$	V_{DC}
Common-Mode Error	Over Analog Input Voltage Range		$\pm 1/2$	$\pm 1/2$	LSB
Supply Sensitivity	$V_{CC} = 5 V_{DC} \pm 10\%$ Over Allowed $V_{IN}(+)$ and $V_{IN}(-)$ Voltage Range (Note 4)		$\pm 1/2$	$\pm 1/2$	LSB

Electrical Characteristics

Following specifications apply for $V_{CC} = 5 V_{DC}$ and $T_A = 25^\circ\text{C}$ unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Conversion Time	$f_{CLK} = 640 \text{ kHz}$ (Note 6)	103		114	μs
Conversion Time	(Note 5, 6)	66		73	$1/f_{CLK}$
Clock Frequency	$V_{CC} = 5V$, (Note 5)	100	640	1460	kHz
Clock Duty Cycle	(Note 5)	40		60	%
Conversion Rate in Free-Running Mode	$\overline{INTA}$ tied to $\overline{WR}$ with $\overline{CS} = 0 V_{DC}$, $f_{CLK} = 640 \text{ kHz}$	8770		9708	conv/s
Width of $\overline{WR}$ Input (Start Pulse Width)	$\overline{CS} = 0 V_{DC}$ (Note 7)	100			ns
Access Time (Delay from Falling Edge of $\overline{RD}$ to Output Data Valid)	$C_L = 100 \text{ pF}$		135	200	ns
TRI-STATE Control (Delay from Rising Edge of $\overline{RD}$ to Hi-Z State)	$C_L = 10 \text{ pF}$, $R_L = 10k$ (See TRI-STATE Test Circuits)		125	200	ns
Delay from Falling Edge of $\overline{WR}$ or $\overline{RD}$ to Reset of $\overline{INTA}$			300	450	ns
Input Capacitance of Logic Control Inputs			5	7.5	pF
TRI-STATE Output Capacitance (Data Buffers)			5	7.5	pF

CONTROL INPUTS (Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately)

(1) Logical "1" Input Voltage (Except Pin 4 CLK IN)	$V_{CC} = 5.25 V_{DC}$	2.0		15	V_{DC}
---	------------------------	-----	--	----	----------

Electrical Characteristics (Continued)

Operating specifications apply for $V_{CC} = 1V_{CC}$ and $T_{min} \leq T_A \leq T_{max}$, unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
ANALOG INPUTS (Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately)					
Logical "0" Input Voltage (except Pin 4 CLK IN)	$V_{CC} = 4.75 V_{CC}$			0.8	V_{CC}
Logical "1" Input Current (all inputs)	$V_{IN} = 5 V_{CC}$		0.005	1	μA_{CC}
Logical "0" Input Current (all inputs)	$V_{IN} = 0 V_{CC}$	-1	-0.005		μA_{CC}
CLK IN AND CLOCK R					
CLK IN (Pin 4) Positive Going Threshold Voltage		2.7	3.1	3.5	V_{CC}
CLK IN (Pin 4) Negative Going Threshold Voltage		1.5	1.5	2.1	V_{CC}
CLK IN (Pin 4) Hysteresis ($V_H - V_L$)		0.6	1.3	2.0	V_{CC}
Logical "0" CLK R Output Voltage	$I_O = 360 \mu A$ $V_{CC} = 4.75 V_{CC}$			0.4	V_{CC}
Logical "1" CLK R Output Voltage	$I_O = -360 \mu A$ $V_{CC} = 4.75 V_{CC}$	2.4			V_{CC}
DATA OUTPUTS AND INTR					
Logical "0" Output Voltage Data Outputs	$I_{OUT} = 1.6 mA, V_{CC} = 4.75 V_{CC}$			0.4	V_{CC}
INTR Output	$I_{OUT} = 1.0 mA, V_{CC} = 4.75 V_{CC}$			0.4	V_{CC}
Logical "1" Output Voltage	$I_O = -360 \mu A, V_{CC} = 4.75 V_{CC}$	2.4			V_{CC}
Logical "1" Output Voltage	$I_O = -10 \mu A, V_{CC} = 4.75 V_{CC}$	4.5			V_{CC}
TRISTATE Disabled Output Leakage (All Data Buffers)	$V_{OUT} = 0 V_{CC}$ $V_{OUT} = 5 V_{CC}$	-3		3	μA_{CC} μA_{CC}
	V_{OUT} Short to Gnd, $T_A = 25^\circ C$	4.5	5		mA_{CC}
	V_{OUT} Short to V_{CC} , $T_A = 25^\circ C$	9.0	16		mA_{CC}
POWER SUPPLY					
Supply Current (Includes Ladder Current)	$I_{CLK} = 640 kHz$ $V_{REF}/2 = NC, T_A = 25^\circ C$ and $CS = 5V$				
ADC0801/02/03/04LCJ/05			1.1	1.8	mA
ADC0804LCN/LCV/LCWM			1.9	2.5	mA

1. Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating outside specified operating conditions.

2. All voltages are measured with respect to Gnd, unless otherwise specified. The separate A Gnd point should always be wired to the D Gnd.

3. A Zener diode exists internally, from V_{CC} to Gnd and has a typical breakdown voltage of 7 V_{CC} .

4. For $V_{IN} = 1/2 V_{CC}$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see block diagram) which will forward bias analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. Be careful, during testing at low V_{CC} levels (4.5V), forward analog inputs (5V) can cause this input diode to conduct—especially at elevated temperatures, and cause errors for analog inputs near full-scale. The diodes 50 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 50 mV, the output will be correct. To achieve an absolute 0 V_{CC} to 5 V_{CC} input voltage range will therefore require a minimum supply voltage of 4.950 V_{CC} over temperature and loading.

5. Accuracy is guaranteed at $I_{CLK} = 640 kHz$. At higher clock frequencies accuracy can degrade. For lower clock frequencies, the duty cycle limits can be as long as the minimum clock high time interval or minimum clock low time interval is no less than 275 ns.

6. With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process. The request is internally latched; see Figure 2 and section 2.0.

7. The CS input is assumed to bracket the WH strobe input and therefore timing is dependent on the WH pulse width. An arbitrarily wide pulse width will hold converter in a reset mode and the start of conversion is initiated by the low to high transition of the WH pulse (see timing diagrams).

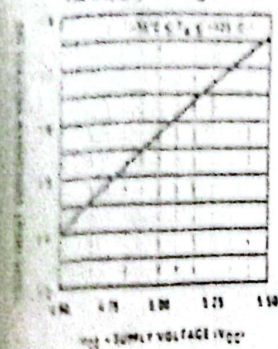
8. None of these A/Ds requires a zero adjust (see section 2.5.1). To obtain zero code at other analog input voltages see section 2.5 and Figure 5.

9. The $V_{REF}/2$ pin is the center point of a two resistor divider connected from V_{CC} to ground. Each resistor is 2.2k, except for the ADC0804LCJ where each resistor is 15k. Total ladder input resistance is the sum of the two equal resistors.

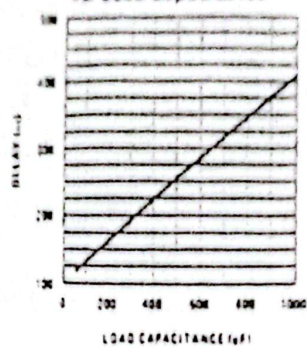
10. Human body model, 100 pF discharged through a 1.5 k Ω resistor.

Typical Performance Characteristics

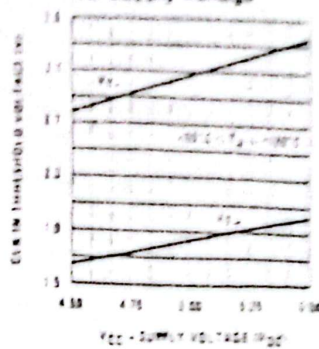
Logic Input Threshold Voltage vs. Supply Voltage



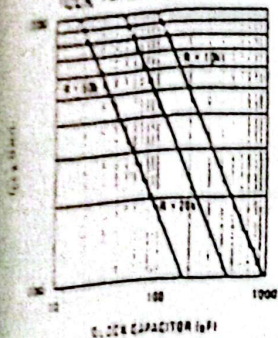
Delay From Falling Edge of RD to Output Data Valid vs. Load Capacitance



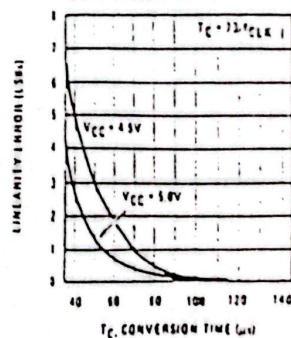
CLK IN Schmitt Trip Levels vs. Supply Voltage



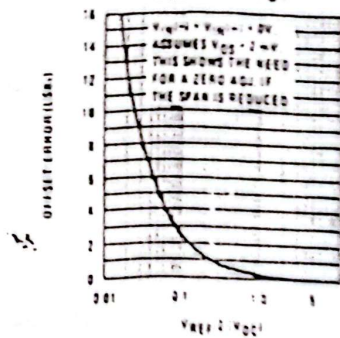
ICLK vs. Clock Capacitor



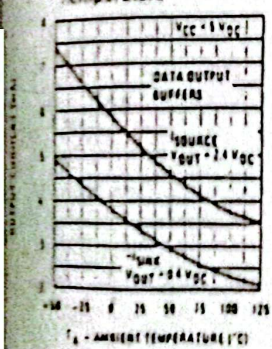
Full-Scale Error vs. Conversion Time



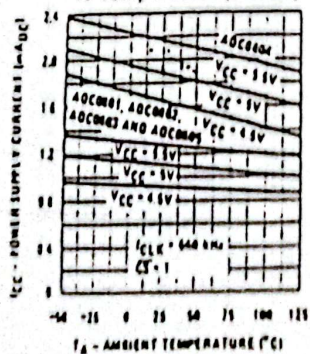
Effect of Unadjusted Offset Error vs. VREF/2 Voltage



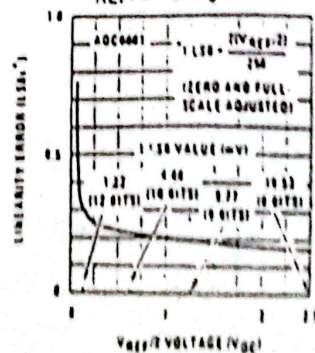
Output Current vs. Temperature



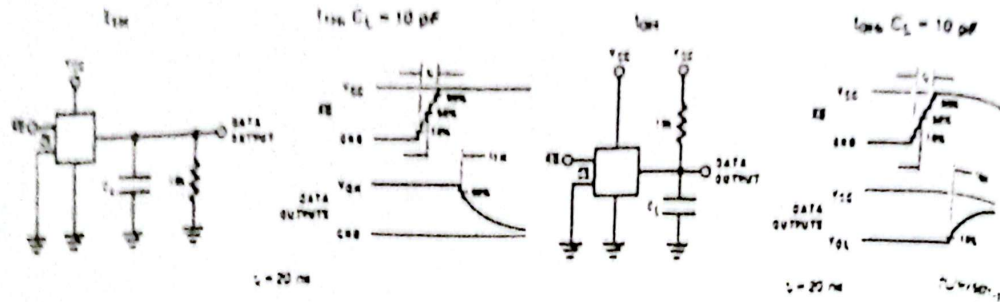
Power Supply Current vs. Temperature (Note 9)



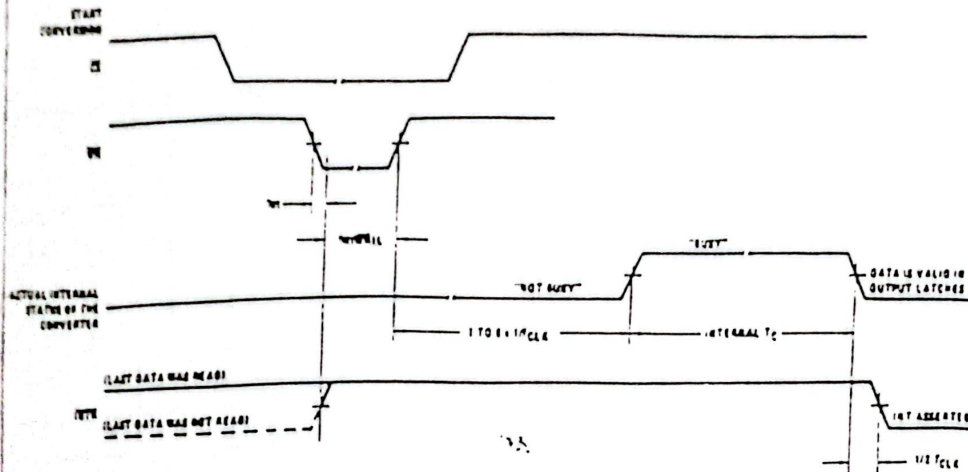
Linearity Error at Low VREF/2 Voltages



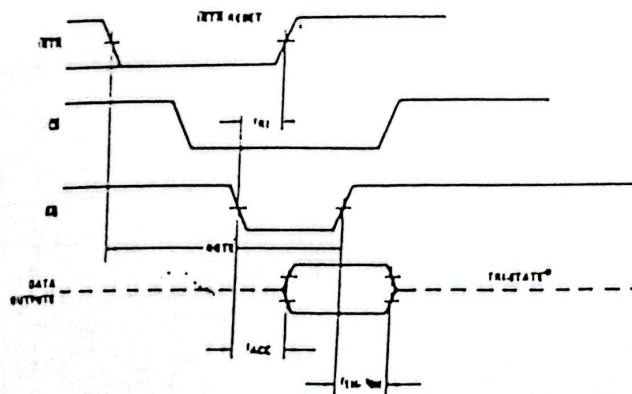
TRI-STATE Test Circuits and Waveforms



Timing Diagrams (All timing is measured from the 50% voltage points)



Output Enable and Reset (INT)

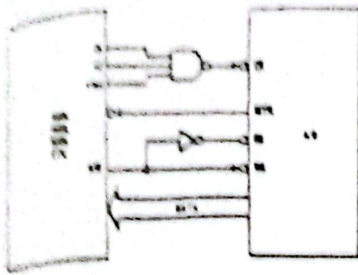


Note: Read strobe must occur 8 clock periods ($8/t_{CLK}$) after assertion of interrupt to guarantee reset of INT.

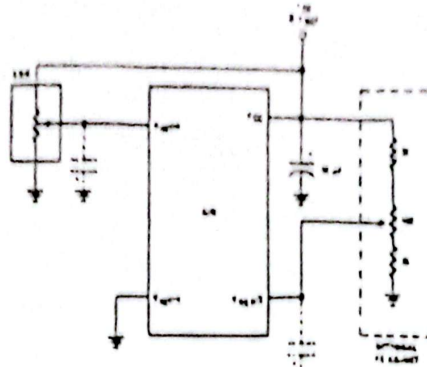
TL/M/5471-4

Typical Applications (Continued)

8000 Interface

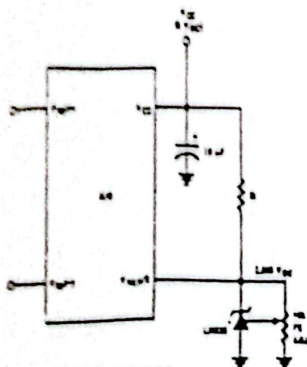


RatioMetric with Full-Scale Adjust



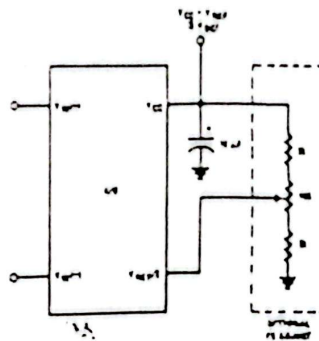
Note: before using code at V_{IN} or $V_{REF}/2$, see section 2.2.2 Input System Capacitors.

Absolute with a 2.500V Reference

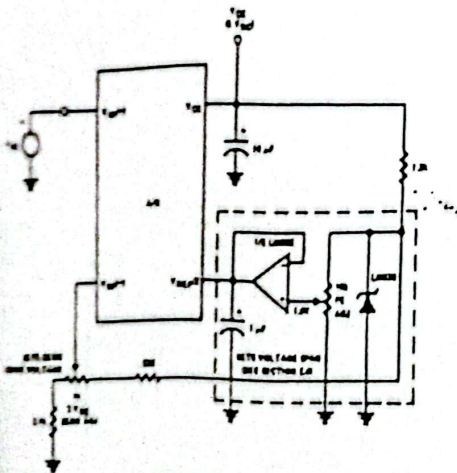


For the power, see also LM385-2.5

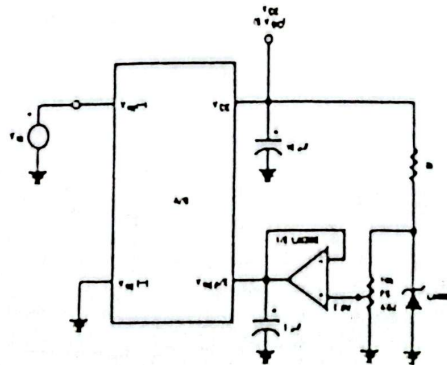
Absolute with a 5V Reference



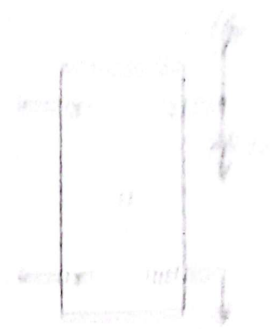
Zero-Shift and Span Adjust: $2V \leq V_{IN} \leq 5V$



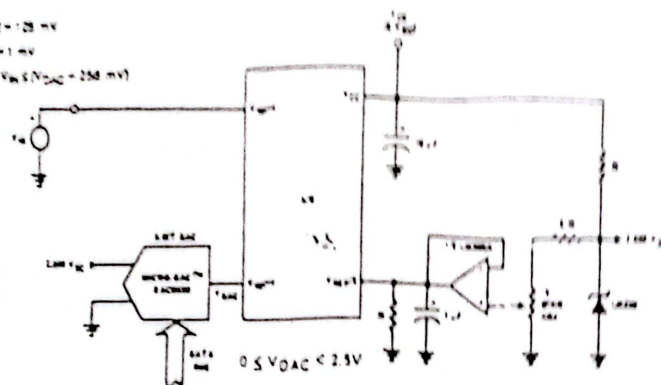
Span Adjust: $0V \leq V_{IN} \leq 3V$



TUW/5471-6

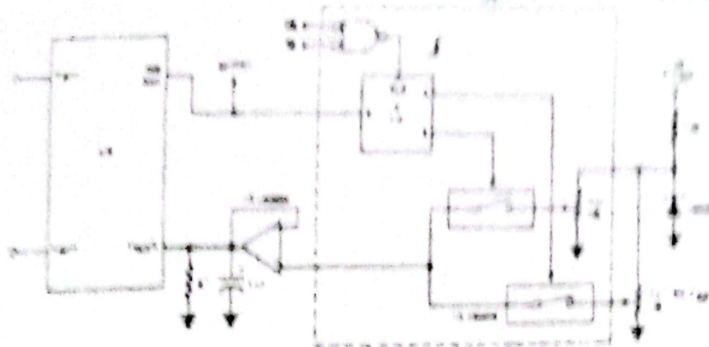
[illegible]
$$\begin{aligned} \text{E.g. } 1000 &= 10^3 \text{ g} \\ 1000000 &= 10^6 \text{ g} \\ \text{E.g. } 1000 &= 10^3 \text{ g} \\ 1000000 &= 10^6 \text{ g} \end{aligned}$$
$$V_{\text{gate}} = 2 = 1.58 \text{ mV}$$

$$1.58 = 1 \text{ mV}$$

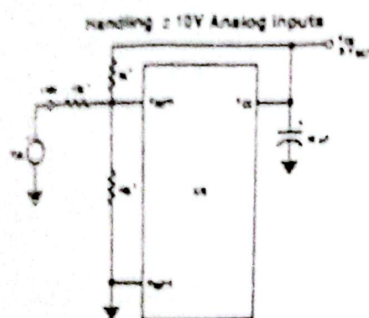
$$V_{\text{gate}} = 5 \text{ V} \Rightarrow V_{\text{gate}} = 2.58 \text{ mV}$$


Typical Applications ■ (Continued)

Competitive Differential-input Comparator with $R_{eq} = 50\text{ k}\Omega$ and $V_{th} = 0.7\text{ V}$ at $V_{DD} = 1.8\text{ V}$

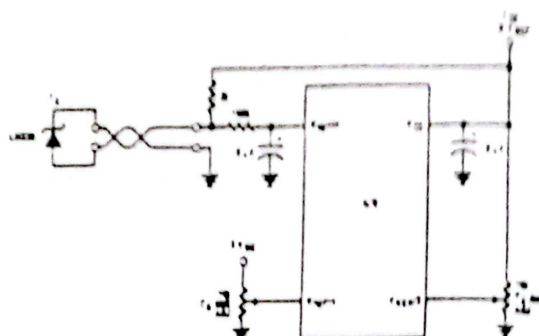


Use Figure 2 to select R value
 $Q(1) = 1.1$ for $N_{adj} = 1.8$ and $n = 1$ (Figure 2)
 Draw delivery within the dotted area if
 $N_{adj} = 1.8$ is not needed

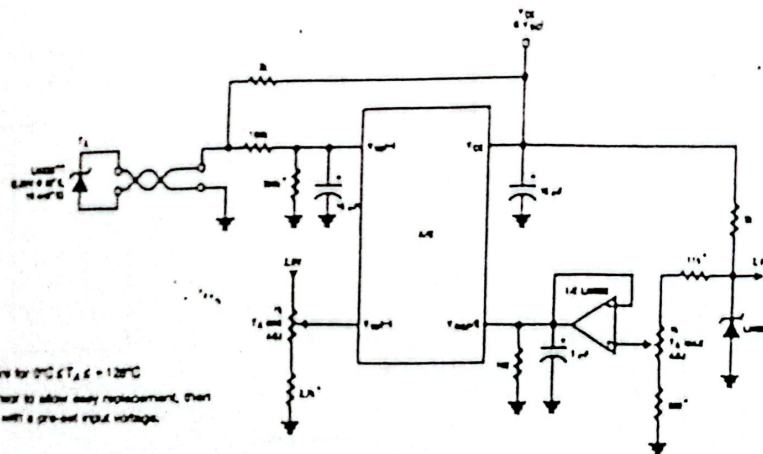


4-20-2018 10:00 AM 4-20-2018 10:00 AM 4-20-2018 10:00 AM

Low-Cost, μP Interfaced, Temperature-to-Digital Converter



μ P Interfaced Temperature-to-Digital Converter

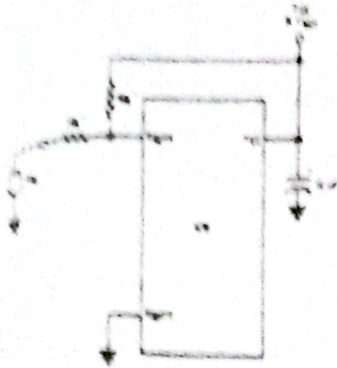


*All values shown are for $20^{\circ}\text{C} \leq T_A \leq 125^{\circ}\text{C}$

2000

Applications (Continued)

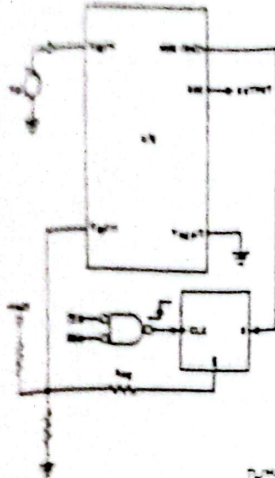
Handling 1 kV Analog Inputs



TU/H/S471-33

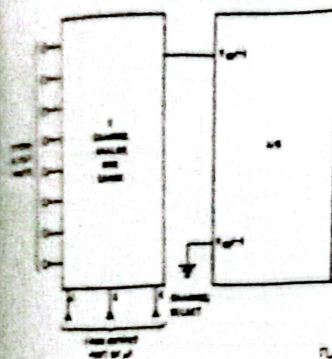
Buffering Input with Resolution Reduced by 1/2

Buffered Comparator with Hysteresis



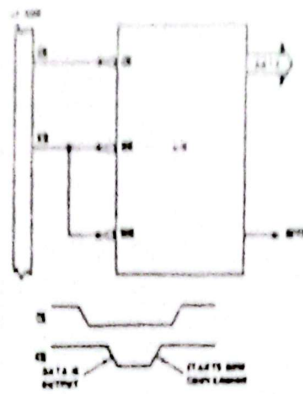
TU/H/S471-34

Analog Self-Test for a System



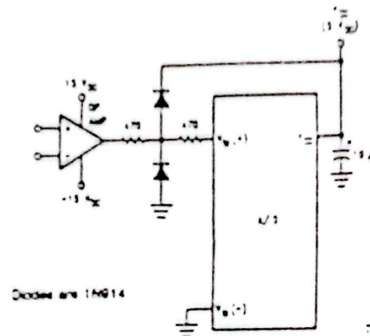
TU/H/S471-36

Reset Circuit for Failure



TU/H/S471-34

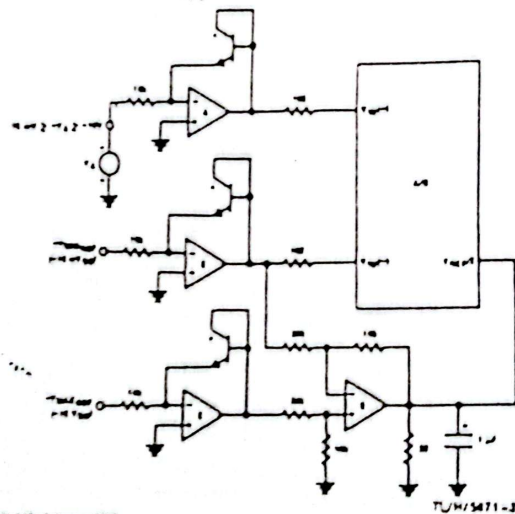
Protecting the Input



Diodes are 1N914

TU/H/S471-3

A Low-Cost, 2-Decade Logarithmic Converter



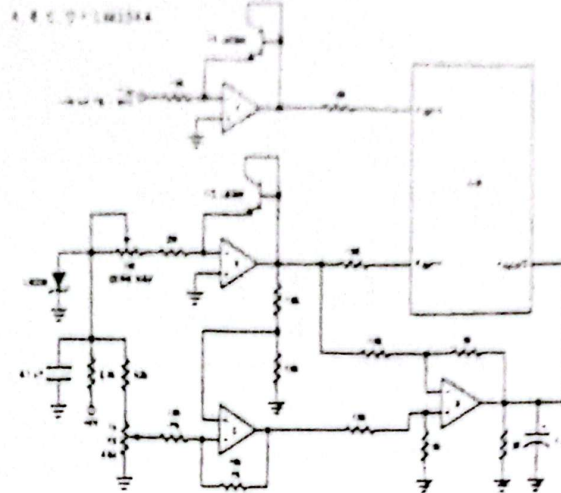
TU/H/S471-37

*LMC2240 transistors

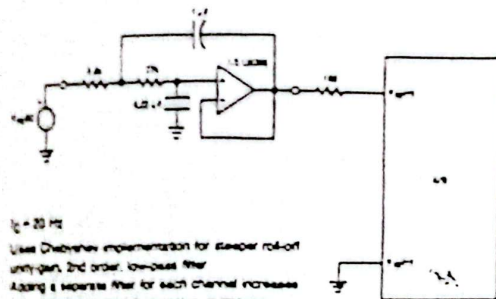
A, B, C, D - LMC224A quad op amp

Typical Applications (Continued)

3-Decade Logarithmic A/D Converter

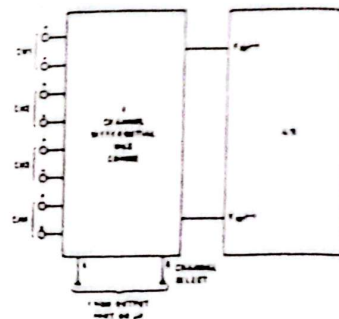


Noise Filtering the Analog Input

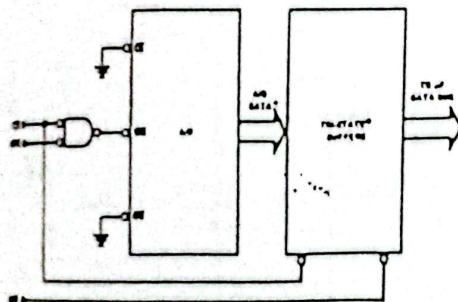


$f_c = 20 \text{ Hz}$
 Use Diodes for implementation for deeper roll-off
 unipolar, 2nd order, low-pass filter.
 Adding a separate filter for each channel increases
 system resolution time if an analog multiplexer
 is used.

Multiplexing Differential Inputs

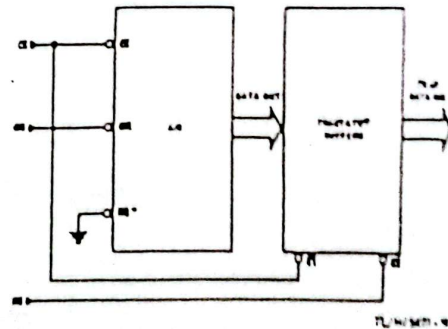


Output Buffers with A/D Data Enabled



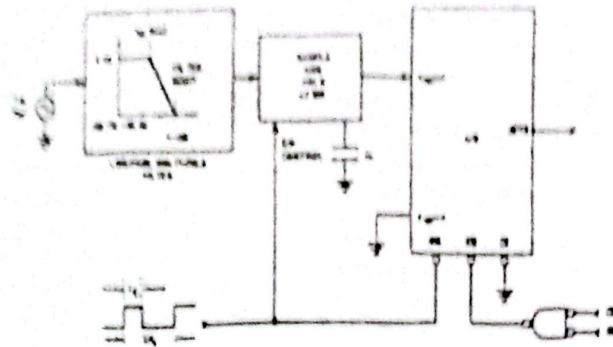
*A/D output data is updated 1 CLK period prior to assertion of $\overline{\text{INT}}$
 * $\overline{\text{INT}}$ is asserted to load

Increasing Bus Drive and/or Reducing Time on Bus



* $\overline{\text{INT}}$ is asserted to load output latch.

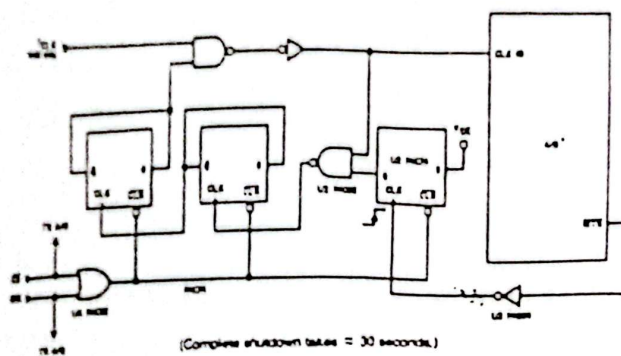
Sampling an A/D Input Signal



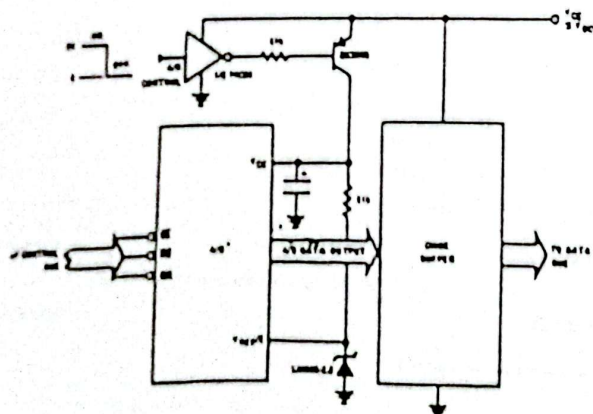
Note 1: Oversample whenever possible (times by 2, 25, or 50) to enhance input frequency tracking (averaging) and to allow for the start-up time of the filter.

Note 2: Consider the quantization error which are introduced when the bandwidth of the filter.

70% Power Savings by Clock Gating



Power Savings by A/D and V_{REF} Shutdown



*Use AD00001, 02, 03 or 04 for lowest power consumption.

Note: Logic inputs can be driven to V_{CC} with A/D supply at zero volts.

Buffer provides data bus from overshooting output of A/D when in shutdown mode.

TLV5611-11

Functional Description

UNDERSTANDING A/D ERROR SPECS

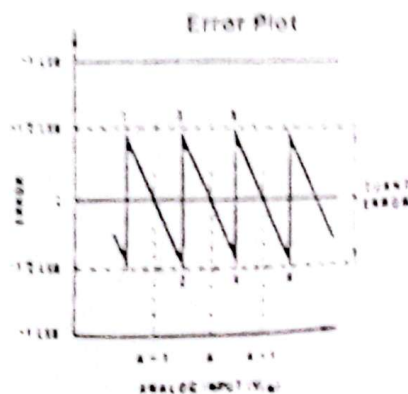
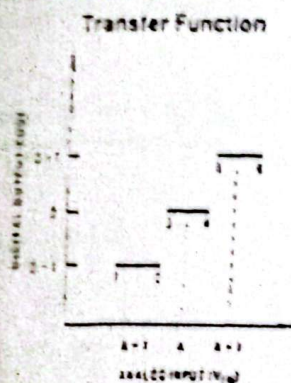
an A/D transfer characteristic (staircase waveform) is shown in Figure 1a. The horizontal scale is analog input voltage and the particular points labeled are in steps of $\pm 1/2$ LSB with 2.5V tied to the $V_{REF}/2$ pin. The digital codes that correspond to these inputs are shown as 0 and 0-1. For the correct A/D, not only will center- $(A-1)$, A , $A+1$ analog inputs produce the correct digital codes, but also each user (the positions are adjacent output codes) will be located $\pm 1/2$ LSB from each center-value. As shown, the users are ideal have no width. Correct digital output codes will be produced for a range of analog input voltages that extend $\pm 1/2$ from the ideal center-values. Each tread (the range of input voltage that provides the same digital output) is therefore, 1 LSB wide.

Figure 1b shows a worst case error plot for the ADC0801. Inputs are guaranteed to produce the correct codes and the adjacent users are guaranteed to be closer to the center-value points than $\pm 1/2$ LSB. In

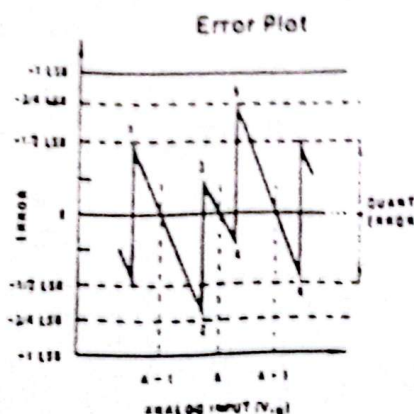
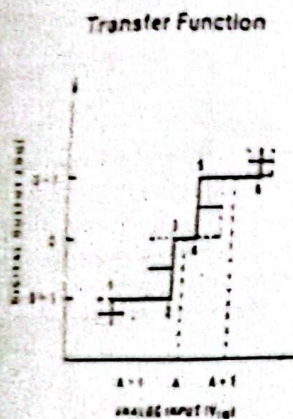
other words if we apply an analog input equal to the center value $\pm 1/2$ LSB, we guarantee that the A/D will produce the correct digital code. The maximum range of the deviation of the code boundaries is indicated by the horizontal arrows and it is guaranteed to be no more than $\pm 1/2$ LSB.

The error curve of Figure 1c shows a worst case error plot for the ADC0802. Here we guarantee that if we apply an analog input equal to the 1 LSB analog voltage center value the A/D will produce the correct digital code.

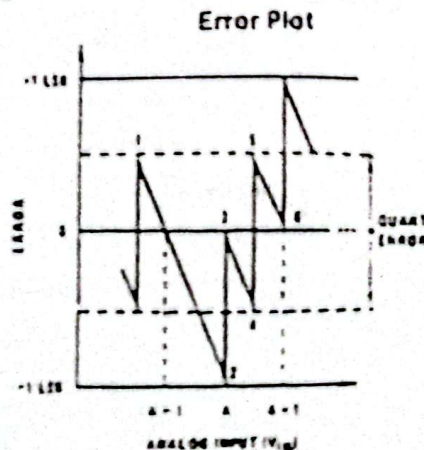
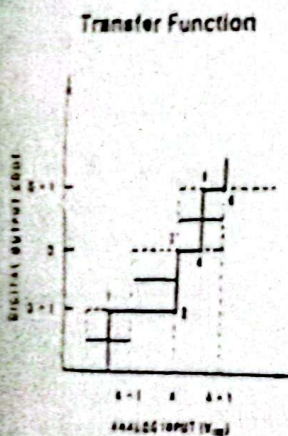
Next to each transfer function is shown the corresponding error plot. Many people may be more familiar with error plots than transfer functions. The analog input voltage to the A/D is provided by either a linear ramp or by the discrete output steps of a high resolution DAC. Notice that the error is continuously displayed and includes the quantization uncertainty of the A/D. For example the error at point 1 of Figure 1a is $-1/2$ LSB because the digital code appeared $1/2$ LSB in advance of the center-value of the tread. The error plots always have a constant negative slope and the adjacent code steps are always 1 LSB in magnitude.



a) Accuracy = ± 0 LSB: A Perfect A/D



b) Accuracy = $\pm 1/2$ LSB



c) Accuracy = $\pm 1/4$ LSB

72-115471-12

Functional Description (Continued)

When the $\overline{RD}$ is latched through the 8-bit shift register, which controls the SAR register, it appears as the input to the D-type latch, LATCH 1, as soon as this "1" is output from the shift register. The AND gate, G2, causes the new data value to transfer to the TRI-STATE output latches. When LATCH 1 is subsequently enabled, the Q output causes a high-to-low transition which causes the INTR F/F to set. An inverting buffer then supplies the $\overline{INTR}$ input sig-

nal. The $\overline{SET}$ control of the INTR F/F remains low for the entire clock period (as the internal clocks run at the frequency of the external clock). If the data output is continuously enabled ($\overline{CS}$ and $\overline{RD}$ both held low), the $\overline{INTR}$ signal will exit signal the end of conversion (by a high-to-low transition), because the $\overline{SET}$ input can control the Q output of the INTR F/F even though the RESET input is present at a "1" level in this operating mode. This $\overline{INTR}$ signal will therefore stay low for the duration of the $\overline{SET}$ signal, which is 3 periods of the external clock frequency during the A/D is not started during this interval).

When operating in the free-running or continuous conversion mode, the INTR pin tied to $\overline{WR}$ and $\overline{CS}$ wired low—see section 2.3), the START F/F is SET by the high-to-low transition of the $\overline{INTR}$ signal. This resets the SHIFT REGISTER, which causes the input to the D-type latch, LATCH 1, to be low. As the latch enable input is still present, the Q output will go high, which then allows the INTR F/F to be SET. This reduces the width of the resulting $\overline{INTR}$ output signal to only a few propagation delays (approximately 300

ns) as it is to be read, the combination of both $\overline{CS}$ and $\overline{RD}$ being low will cause the INTR F/F to be reset and the TRI-STATE output latches will be enabled to provide the 8-bit outputs.

Digital Control Inputs

Digital control inputs ($\overline{CS}$, $\overline{RD}$, and $\overline{WR}$) meet standard logic voltage levels. These signals have been renamed to comply to the standard A/D Start and Output Enable functions. In addition, these inputs are active low to allow for interface to microprocessor control busses. For microprocessor-based applications, the $\overline{CS}$ input (pin 1) is grounded and the standard A/D Start function is achieved by an active low pulse applied at the $\overline{WR}$ input (pin 10). The Output Enable function is caused by an active low pulse at the $\overline{RD}$ input (pin 2).

Using Differential Voltage Inputs and Common-Mode Rejection

The AD has additional applications flexibility due to the differential voltage input. The $V_{IN}(-)$ input (pin 7) is used to automatically subtract a fixed voltage value from the input reading (zero correction). This is also useful in 0-5mA current loop conversion. In addition, common-mode error can be reduced by use of the differential input. The interval between sampling $V_{IN}(+)$ and $V_{IN}(-)$ is 4 clock periods. The maximum error voltage due to this

slight time difference between the input voltage samples is given by:

$$\Delta V_e(\text{MAX}) = (V_a)(2\pi f_{cm}) \left(\frac{4.5}{f_{CLK}} \right)$$

where:

ΔV_e is the error voltage due to sampling delay

V_a is the peak value of the common-mode voltage

f_{cm} is the common-mode frequency

As an example, to keep this error to $1/2$ LSB (≈ 5 mV) when operating with a 50 Hz common-mode frequency, f_{cm} , and using a 640 kHz A/D clock, f_{CLK} , would allow a peak value of the common-mode voltage, V_a , which is given by:

$$V_p = \frac{[\Delta V_e(\text{MAX})](f_{CLK})}{(2\pi f_{cm})(4.5)}$$

or

$$V_p = \frac{(5 \times 10^{-3})(640 \times 10^3)}{(6.28)(50)(4.5)}$$

which gives

$$V_p \approx 1.9V$$

The allowed range of analog input voltages usually places more severe restrictions on input common-mode noise levels.

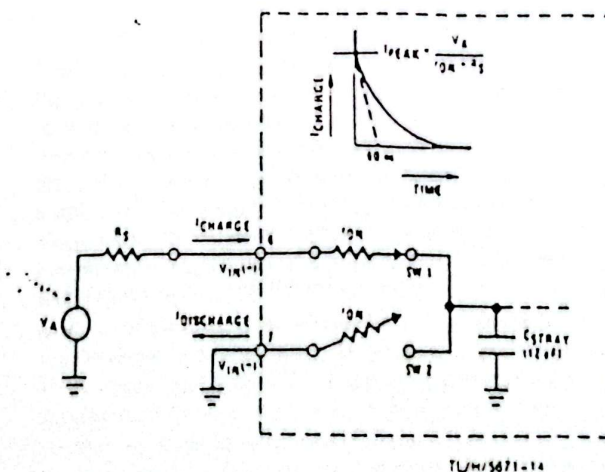
An analog input voltage with a reduced span and a relatively large zero offset can be handled easily by making use of the differential input (see section 2.4 Reference Voltage).

2.3 Analog Inputs

2.3.1 Input Current

Normal Mode

Due to the internal switching action, displacement currents will flow at the analog inputs. This is due to on-chip stray capacitance to ground as shown in Figure 3.



$$R_{ON} \text{ of SW 1 and SW 2} = 5 \text{ k}\Omega$$

$$C_{STRAY} = 5 \text{ k}\Omega \times 12 \text{ pF} = 60 \text{ ns}$$

FIGURE 3. Analog Input Impedance

Functional Description (Continued)

on this resistance is switched and will result in changing the $V_{IN}(-)$ input pin and leaving the output which will depend on the analog differential voltage levels. These current transients occur at the edges of the internal clocks. They rapidly decay and disappear as the on-chip comparator is strobed at the end of the clock period.

noise.

source applied to the $V_{IN}(-)$ or $V_{IN}(+)$ pin is the allowed operating range of $V_{CC} - 50$ mV. Large currents can flow through a parasitic diode to the V_{CC} pin. Currents can exceed the 1 mA max allowed by the diode (1N914) should be added to bypass the V_{CC} pin (with the current bypassed with the V_{CC} pin) (with the current bypassed with the V_{CC} pin) (with the current bypassed with the V_{CC} pin). The voltage at the $V_{IN}(-)$ pin can exceed the range of the forward voltage of this diode).

Input Bypass Capacitors

Capacitors at the inputs will average these charges and allow DC current to flow through the output resistors of the analog signal sources. This charge pumping is worse for continuous conversions with the $V_{IN}(-)$ input at full-scale. For continuous conversions with the clock frequency with the $V_{IN}(+)$ input at 5V, this current is at a maximum of approximately 5 μ A. Therefore, capacitors should not be used at the analog input of the $V_{REF}/2$ pin for high resistance sources (> 1 k Ω). Input bypass capacitors are necessary for noise filter. High source resistance is desirable to minimize crosstalk. The detrimental effects of the voltage drop on input resistance, which is due to the average input current, can be eliminated with a full-scale input while the given source resistor and input bypass capacitor are both in place. This is possible because the value of the input current is a precise linear function of the differential input voltage.

Input Source Resistance

Values of source resistance where an input bypass capacitor is not used, will not cause errors as the input current is out prior to the comparison time. If a low pass filter is used in the system, use a low valued series resistor for a passive RC section or add an op amp RC low pass filter. For low source resistance applications (e.g., a 0.1 μ F bypass capacitor at the inputs will cause pickup due to series lead inductance of a long PCB series resistor can be used to isolate this capacitor. The R and C are placed outside the feedback of the output of an op amp, if used.

are

to the analog inputs (pin 6 and 7) should be kept as possible to minimize input noise coupling. Both undesired digital clock coupling to these inputs can cause system errors. The source resistance for these inputs, in general, be kept below 5 k Ω . Larger values of resistance can cause undesired system noise but bypass capacitors, placed from the analog input, will eliminate system noise pickup but can cause scale errors as these capacitors will average the input switching currents of the A/D (see section 2.4.1). This scale error depends on both a large source

resistance and the use of an input bypass capacitor. This error can be eliminated by doing a full-scale adjustment of the A/D (adjust $V_{REF}/2$ for a proper full-scale reading—see section 2.4.2 on Full-Scale Adjustment) with the source resistance and input bypass capacitor in place.

2.4 Reference Voltage

2.4.1 Span Adjust

For maximum applications flexibility, these A/Ds have been designed to accommodate a 5 V_{CC} , 2.5 V_{CC} or an adjusted voltage reference. This has been achieved in the design of the IC as shown in Figure 4.

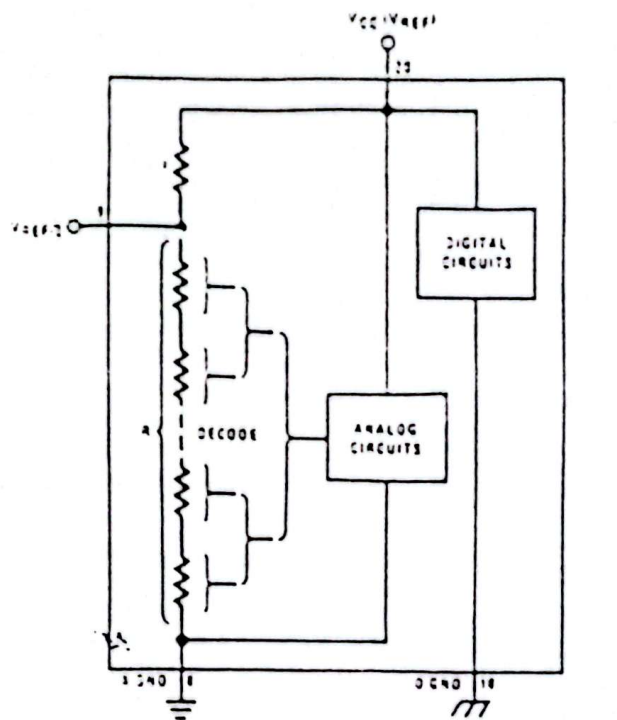
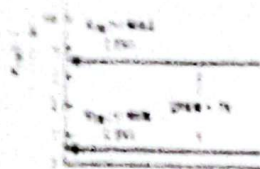


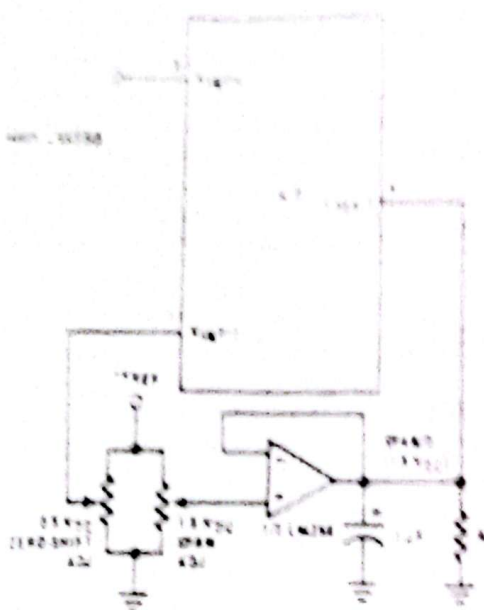
FIGURE 4. The $V_{REFERENCE}$ Design on the IC

Notice that the reference voltage for the IC is either $1/2$ of the voltage applied to the V_{CC} supply pin, or is equal to the voltage that is externally forced at the $V_{REF}/2$ pin. This allows for a ratiometric voltage reference using the V_{CC} supply, a 5 V_{CC} reference voltage can be used for the V_{CC} supply or a voltage less than 2.5 V_{CC} can be applied to the $V_{REF}/2$ input for increased application flexibility. The internal gain to the $V_{REF}/2$ input is 2, making the full-scale differential input voltage twice the voltage at pin 9.

An example of the use of an adjusted reference voltage is to accommodate a reduced span—or dynamic voltage range of the analog input voltage. If the analog input voltage were to range from 0.5 V_{CC} to 3.5 V_{CC} , instead of 0V to 5 V_{CC} , the span would be 3V as shown in Figure 5. With 0.5 V_{CC} applied to the $V_{IN}(-)$ pin to absorb the offset, the reference voltage can be made equal to $1/2$ of the 3V span or 1.5 V_{CC} . The A/D now will encode the $V_{IN}(+)$ signal from 0.5V to 3.5 V with the 0.5V input corresponding to zero and the 3.5 V_{CC} input corresponding to full-scale. The full 8 bits of resolution are therefore applied over this reduced analog input voltage range.



a) Analog Input Signal Example



b) Accommodating an Analog Input from 0.5V (Digital Out = 00hex) to 3.5V (Digital Out = FFhex)

FIGURE 5. Adapting the A/D Analog Input Voltages to Match an Arbitrary Input Signal Range

Reference Accuracy Requirements

The converter can be operated in a ratiometric mode or an absolute mode. In ratiometric converter applications, the accuracy of the reference voltage is a factor in both the accuracy of the source transducer and the output of the A/D converter and therefore cancels out in the final digital output. The ADC0805 is specified particularly for use in ratio-conversion applications with no adjustments required. In absolute-conversion applications, both the initial value and the long-term stability of the reference voltage are important factors in the accuracy of the A/D converter. For $V_{REF}/2$ (pin 9) at 2.5V nominal value, initial errors of ± 10 mV will cause conversion errors of ± 1 LSB due to the offset of the $V_{REF}/2$ input. In reduced span applications, the value and the stability of the $V_{REF}/2$ input voltage are even more important. For example, if the span is reduced to 2.5V, the analog input LSB voltage value is correspondingly reduced from 20 mV (5V span) to 10 mV and the $V_{REF}/2$ input becomes 5 mV. As can be seen, this reduces the allowed initial tolerance of the reference voltage and requires correspondingly less absolute change in temperature variations. Note that spans smaller than this place even tighter requirements on the initial accuracy and stability of the reference source.

Next, the magnitude of the reference voltage will require initial adjustment. Errors due to an improper value in the reference voltage appear as full-scale errors in the A/D conversion. IC voltage regulators may be used for reference voltage adjustment if the ambient temperature changes are not excessive. The LM339 2.5V IC reference diode (from National Semiconductor) has a temperature stability of 1.8 mV typ (± 1 max) over $0^\circ\text{C} \leq T_A \leq 70^\circ\text{C}$. Other temperature stable diodes are also available.

2.5 Errors and Reference Voltage Adjustments

2.5.1 Zero Error

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

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

2.5.2 Full-Scale

The full-scale adjustment can be made by applying a differential input voltage that is $1/2$ LSB less than the desired analog full-scale voltage range and then adjusting the magnitude of the $V_{REF}/2$ input (pin 9 or the V_{CC} supply if pin 9 is not used) for a digital output code that is just changing from 1111 1110 to 1111 1111.

Functional Description (Continued)

Following up on an automatic analog input voltage switch.

When the range of the A/D is shifted away from the previous range, the analog input signal must be set to ground. The new zero reference voltage is determined by the A/D's V_{REF} voltage that is applied to the reference pin. V_{REF} is V_{CC} (where the V_{REF} is applied to pin 3) and the zero reference voltage is V_{SS} . After the adjustment is just obtain the V_{REF} voltage once reference.

After adjustment should then be made with the V_{REF} voltage applied by forcing a voltage to the V_{REF} which is given by:

$$V_{REF} = V_{CC} + 1.5 \left(\frac{V_{MAX} - V_{MIN}}{256} \right)$$

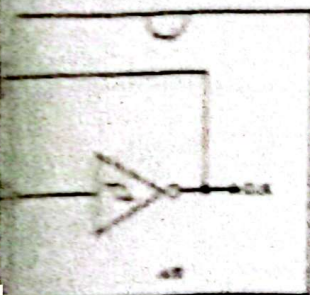
The left end of the analog input range

When the end (the other zero) of the analog range, V_{REF} is ground referenced.

After the V_{REF} voltage is then adjusted to provide a range from V_{REF} to V_{REF} . This completes the adjustment.

Timing Diagram

The A/D can be derived from the CPU clock or external AC can be added to provide self-clocking. The timing makes use of a Schmitt trigger as shown in



$$CLK = \frac{1}{1.1RC}$$

$$R \approx 10 \text{ k}\Omega$$

$V_{CC} = 5.0 \text{ V}$

FIGURE 1. Self-Clocking the A/D

capacitive or DC loading of the clock R pin should be avoided. This will disturb normal converter operation. A 10 pF capacitor, such as driving up to 7 A/D converters from a single clock R pin of 1 converter, are recommended. For larger clock line loading, a CMOS or low power input logic should be used to minimize loading on the clock R pin (do not use a standard TTL input).

Starting a Conversion

When the V_{REF} and V_{REF} go low and return high) conversion, the converter is reset and a new conversion. The output data latch is not updated if the

conversion in process is not allowed to be completed. Therefore the data of the previous conversion remains in the latch. The $INTF$ output simply remains at the V_{REF} level.

2.8 Continuous Conversions

For operation in the free-running mode an initializing pulse should be used, following power-up, to ensure circuit operation. In this application, the V_{REF} input is grounded and the V_{REF} input is tied to the $INTF$ output. This V_{REF} and $INTF$ node should be momentarily forced to logic low following a power-up cycle to guarantee operation.

2.9 Driving the Data Bus

This MOS A/D, like MOS microprocessors and memories, will require a bus driver when the total capacitance of the data bus gets large. Other circuitry, which is tied to the data bus, will add to the total capacitive loading, even in TRI-STATE (high impedance mode). Backplane bussing also greatly adds to the stray capacitance of the data bus.

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

At higher CPU clock frequencies time can be extended for I/O reads (and/or writes) by inserting wait states (8086) or using clock extending circuits (6800).

Finally, if time is short and capacitive loading is high, external bus drivers must be used. These can be TRI-STATE buffers (low power Schottky such as the DM74LS240 series is recommended) or special higher drive current products which are designed as bus drivers. High current bipolar bus drivers with PNP inputs are recommended.

2.10 Power Supplies

Noise spikes on the V_{CC} supply line can cause conversion errors as the comparator will respond to this noise. A low inductance tantalum filter capacitor should be used close to the converter V_{CC} pin and values of 1 μF or greater are recommended. If an unregulated voltage is available in the system, a separate LM340LAZ-5.0, TO-92, 5V voltage regulator for the converter (and other analog circuitry) will greatly reduce digital noise on the V_{CC} supply.

2.11 Wiring and Hook-Up Precautions

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

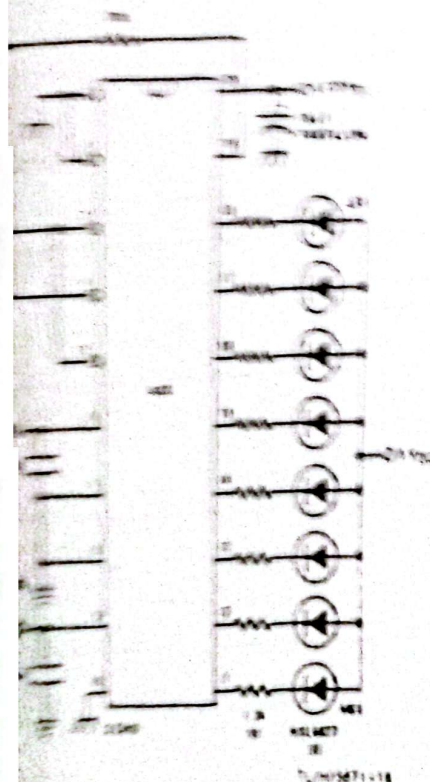


FIGURE 7 Basic A/D Tester

The first step in the design of the A/D converter is the selection of the number of bits to be converted. The number of bits is determined by the required resolution of the converter. For example, a 10-bit converter would provide a resolution of 1/1024 of the full-scale range.

The next step is to select the type of converter to be used. There are two basic types of A/D converters: the successive approximation converter and the dual-slope converter. The successive approximation converter is generally faster and more accurate than the dual-slope converter.

The third step is to select the reference voltage for the converter. The reference voltage is the voltage against which the input signal is compared. It must be a stable, precise voltage.

4.1. THE SUCCESSIVE APPROXIMATION CONVERTER

The successive approximation converter is a type of A/D converter that uses a successive approximation register (SAR) to convert the input signal. The SAR is a register that holds the digital value of the input signal. The SAR is initialized to zero and then the input signal is compared to the current value of the SAR. If the input signal is greater than the current value, the SAR is incremented by one. If the input signal is less than the current value, the SAR is decremented by one. This process is repeated until the SAR value is equal to the input signal.

4.2. THE DUAL-SLOPE CONVERTER

The dual-slope converter is a type of A/D converter that uses two different slopes to convert the input signal. The first slope is used to integrate the input signal for a fixed period of time. The second slope is used to de-integrate the signal until it reaches zero. The time it takes for the signal to reach zero is proportional to the input signal. This time is then converted into a digital value by a counter.

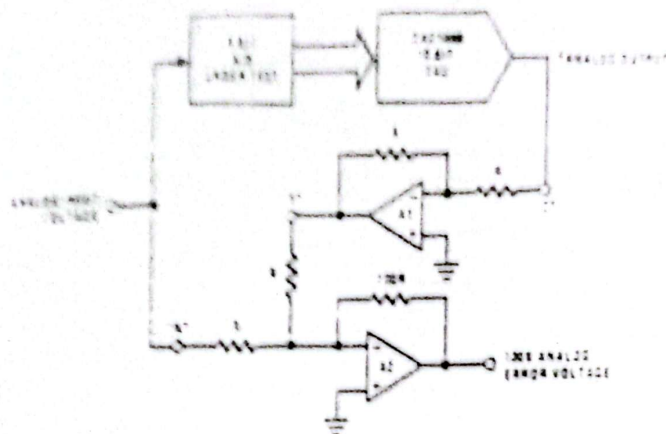


FIGURE 8. A/D Tester with Analog Error Output

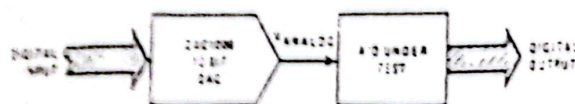
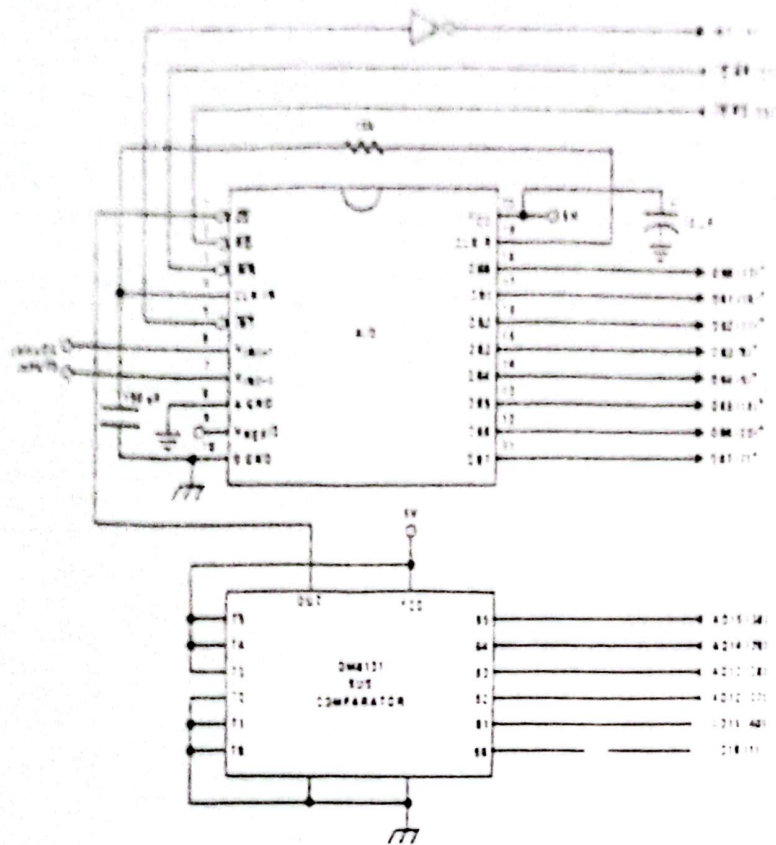


FIGURE 9. Basic "Digital" A/D Tester

TABLE I. DECODING THE DIGITAL OUTPUT LED_s

BINARY	FRACTIONAL BINARY VALUE FOR				OUTPUT VOLTAGE CENTER VALUES WITH $V_{REF}/2 = 2.560 V_{DC}$	
	MS GROUP		LS GROUP		VMS GROUP*	VLS GROUP*
1 1 1	7/8	15/16	7/128	15/256	4.800	0.300
1 1 0		13/16		13/256	4.480	0.280
1 0 1		11/16		11/256	4.160	0.260
1 0 0		9/16		9/256	3.840	0.240
0 1 1	5/8	11/16	5/128	11/256	3.520	0.220
0 1 0		9/16		9/256	3.200	0.200
0 0 1		7/16		7/256	2.880	0.180
0 0 0		5/16		5/256	2.560	0.160
1 1 1	3/8	7/16	3/128	7/256	2.240	0.140
1 1 0		5/16		5/256	1.920	0.120
1 0 1		3/16		3/256	1.600	0.100
1 0 0		1/16		1/256	1.280	0.080
0 1 1	1/8	3/16	1/128	3/256	0.960	0.060
0 1 0		1/8		1/256	0.640	0.040
0 0 1		1/16		1/256	0.320	0.020
0 0 0					0	0

* - VMS Group - VLS Group



TDH04871-02

Note 1: Pin numbers for the DP8229 system controller, others are INS8080A.

Note 2: Pin 23 of the INS8229 must be tied to -1.2V through a 1 k Ω resistor to generate the RST 7 instruction when an interrupt is acknowledged as required by the accompanying sample program.

FIGURE 10. ADC0801-INS8080A CPU Interface

SAMPLE PROGRAM FOR FIGURE 10 ADC0801-INS8080A CPU INTERFACE

```

00 00 03  RST 7:      JWP  LD DATA
.
.
21 00 02  START:     LXI H 0200H      ; HL pair will point to
.                        ; data storage locations
01 00 04  RETURN:    LXI SP 0400H    ; Initialize stack pointer (Note 1)
7D                MOV A, L           ; Test # of bytes entered
FE 0F          CPI 0FH              ; If # = 16, JWP to
0A 13 01      JZ CONT              ; user program
03 E0          OUT 20H              ; Start A/D
7E                EI                ; Enable interrupt
00                NOP                ; Loop until end of
01 0F 01      JWP LOOP              ; conversion
.
.      CONT:
.
.      (User program to
.      process data)
.
.
0B E0  LD DATA:    IN 20H           ; Load data into accumulator
77                MOV M, A           ; Store data
23                INX H              ; Increment storage pointer
03 03 01      JWP RETURN

```

Pointer must be dimensioned because a RST 7 instruction pushes the PC onto the stack.
 * used were arbitrarily chosen.

Functional Description (Continued)

digital control bus signals of the 8080 $\overline{CS}$, $\overline{RD}$ and $\overline{WR}$ are directly wired to the digital control inputs of the A/D. The timing requirements are met to allow data to be converted and outputting the data while the data is being converted. The A/D should be used for larger microprocessor systems where the data bus leaves the PC board and/or where the loads are larger than 100 pF.

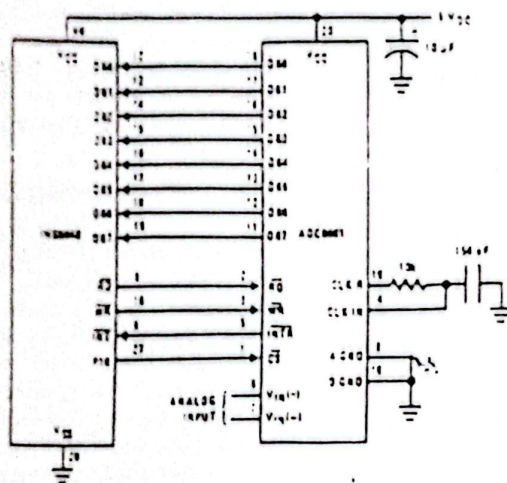
8080A CPU Interfacing Circuitry and Program

The sample program and associated hardware shown in Figure 11 may be used to input data from the 8080A CPU chip set (composed of the microprocessor, the INS8228 system controller and a clock generator). For simplicity, the A/D is used as an I/O device, specifically an 8-bit bi-directional device at an arbitrarily chosen port address, E0. The output capability of the A/D eliminates the need for an external interface device, however address decoding is required to generate the appropriate $\overline{CS}$ for the con-

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

4.1.2 INS8048 Interface

The INS8048 interface technique with the ADC0801 series (see Figure 11) is simpler than the 8080A CPU interface. There are 24 I/O lines and three test input lines in the 8048. With these extra I/O lines available, one of the I/O lines (bit 0 of port 1) is used as the chip select signal to the A/D, thus eliminating the use of an external address decoder. Bus control signals $\overline{RD}$, $\overline{WR}$ and $\overline{INT}$ of the 8048 are tied directly to the A/D. The 16 converted data words are stored at on-chip RAM locations from 20 to 2F (Hex). The $\overline{RD}$ and $\overline{WR}$ signals are generated by reading from and writing to a dummy address, respectively. A sample interface program is shown below.



70H/5671-21

FIGURE 11. INS8048 Interface

SAMPLE PROGRAM FOR FIGURE 11 INS8048 INTERFACE

```

JMP      10H          ; Program starts at addr 10
ORG      3H
JMP      50H          ; Interrupt jump vector
ORG      10H          ; Main program
ANL      P1, #0FEH    ; Chip select
MOVX     A, @R1        ; Read in the 1st data
; to reset the Intr
START:   ORL      P1, #1 ; Set port pin high
MOV      R0, #20H     ; Data address
MOV      R1, #0FFH    ; Dummy address
MOV      R2, #10H     ; Counter for 16 bytes
AGAIN:   MOV      A, #0FFH ; Set ACC for intr loop
ANL      P1, #0FEH    ; Send CS (bit 0 of P1)
MOVX     @R1, A       ; Send #R out
EN       I            ; Enable interrupt
LOOP:    JNZ      LOOP ; Wait for interrupt
DJNZ     R2, AGAIN   ; If 16 bytes are read
NOP      ; go to user's program
NOP
INDATA:  ORG      50H
MOVX     A, @R1        ; Input data, CS still low
MOV      @R0, A       ; Store in memory
INC      R0            ; Increment storage counter
ORL      P1, #1       ; Reset CS signal
CLR      A            ; Clear ACC to get out of
RETR     ; the interrupt loop

```


Functional Description (Continued)

Interfacing the Z-80

The Z-80 control bus is slightly different from that of the 8080. General RD and WR strobes are provided and separate RD and WR strobes are provided. RD and WR strobes are used when they have to be combined with the general RD and WR strobes to provide the equivalent 8080 signals. An advantage of combining the A/D in I/O space with the Z-80 is that the CPU will automatically insert one wait state (the RD and WR strobes are extended one clock period) to allow the time for the I/O devices to respond. Logic to map the A/D to I/O space is shown in Figure 13.

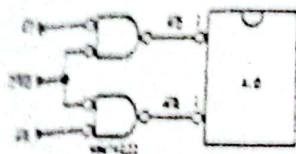


FIGURE 13 Mapping the A/D as an I/O Device for Use with the Z-80 CPU

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

Interfacing 6800 Microprocessor Derivatives (6802, etc.)

Control bus for the 6800 microprocessor derivatives (6802, etc.) use the RD and WR strobe signals. Instead it employs a single A/W line and additional timing, if needed, can be derived from the $\phi 2$ clock. All I/O devices are memory mapped in the 6800 system, and a special signal, VMA, indicates that the current address is valid. Figure 14 shows block schematic where the A/D is memory mapped in 6800 system. For simplicity, the CS decoding is shown in Figure 14. Note that in many 6800 systems, an all-

ready decoded $\overline{CS}$ line is brought out to the common bus at pin 21. This can be tied directly to the $\overline{CS}$ pin of the A/D provided that no other devices are addressed at HEX ADDR 1xxx or 5xxx.

The following subroutine performs essentially the same function as in the case of the 8080A interface and it can be called from anywhere in the user's program.

In Figure 15 the ADC0801 series is interfaced to the M6800 microprocessor through the commonly chosen Port B of the MC6820 or MC6821 Peripheral Interface Adapter (PIA). Here the $\overline{CS}$ pin of the A/D is grounded since the PIA is already memory mapped in the M6800 system and no $\overline{CS}$ decoding is necessary. Also notice that the A/D output data lines are connected to the microprocessor bus under program control through the PIA and therefore the A/D RD pin can be grounded.

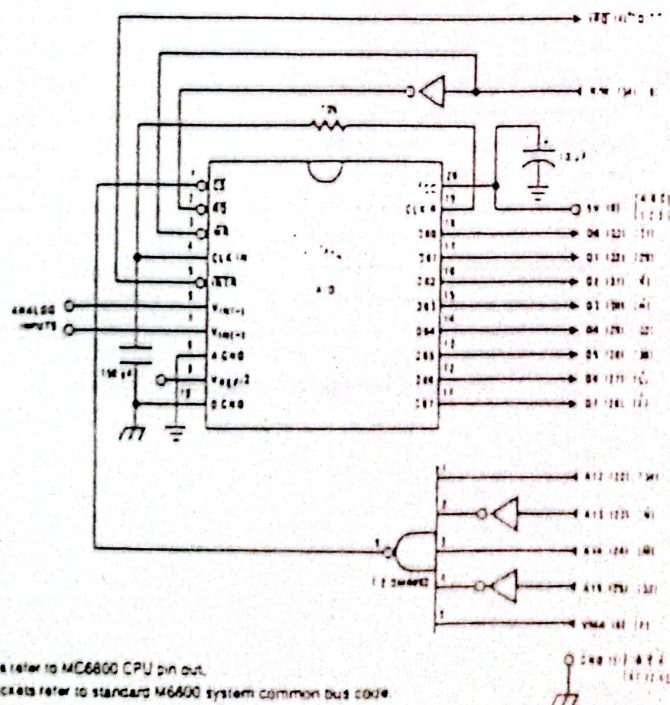
A sample interface program equivalent to the previous one is shown below Figure 15. The PIA Data and Control Registers of Port B are located at HEX addresses 8006 and 8007, respectively.

5.0 GENERAL APPLICATIONS

The following applications show some interesting uses for the A/D. The fact that one particular microprocessor is used is not meant to be restrictive. Each of these application circuits would have its counterpart using any microprocessor that is desired.

5.1 Multiple ADC0801 Series to MC6800 CPU Interface

To transfer analog data from several channels to a single microprocessor system, a multiple converter scheme presents several advantages over the conventional multiplexer single-converter approach. With the ADC0801 series, the differential inputs allow individual span adjustment for each channel. Furthermore, all analog input channels are sensed simultaneously, which essentially divides the microprocessor's total system servicing time by the number of channels, since all conversions occur simultaneously. This scheme is shown in Figure 16.



Pin numbers in parentheses refer to MC6800 CPU pin out.

Pin numbers in brackets refer to standard M6800 system common bus code.

FIGURE 14. ADC0801-MC6800 CPU Interface

SAMPLE PROGRAM FOR FIGURE 14 ADC0801 MC6800 CPU INTERFACE

Address	Instruction	Op Code	Comments
07 36	DATIN	STX	TEMP1
08 00 00		LDX	=30000
09 FF F8		STX	TEMP1
0A 50 00		STAA	30000
0B		CLI	
0C	CONVRT	HAL	(Wait for interrupt)
0D 34		LDX	TEMP1
0E 00 0F		CPX	=3020F
0F 14		BEQ	ENDP
10 50 00		STAA	30000
11		INX	
12 34		STX	TEMP1
13 70		BRA	CONVRT
14 34	INTRPT	LDX	TEMP1
15 50 00		LDAA	30000
16 00		STAA	X
17		RTI	
18 00	TEMP1	FDB	30200
19 00			
1A 00 00	TEMP2	FDB	30000
1B 00 00	ENDP	LDX	=30200
1C 34		STX	TEMP1
1D 36		LDX	TEMP2
1E		RTS	

For the microprocessor to service subroutines and interrupts, the stack pointer must be dimensioned in the user's program.

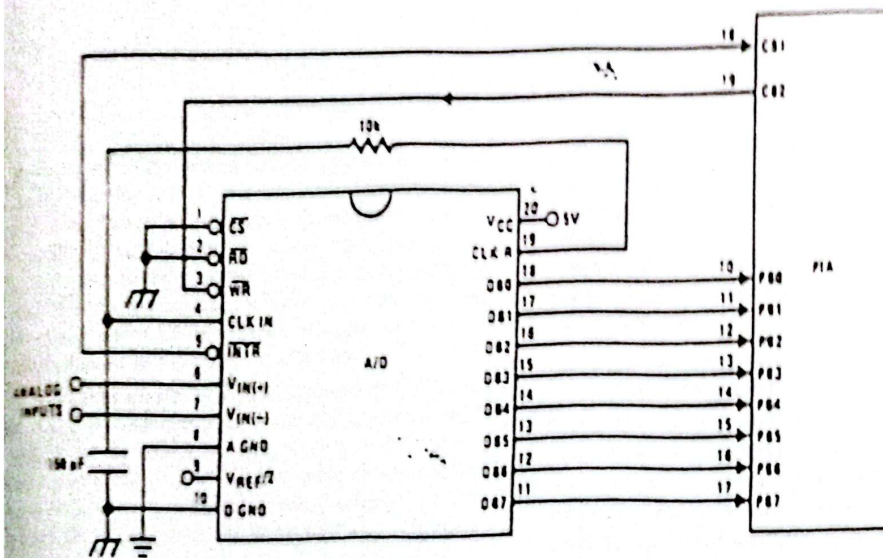


FIGURE 15. ADC0801-MC6820 PIA Interface

9471-23

General Description

SAMPLE PROGRAM FOR PRODUCE 1X ADC0801 MICRO20 PIA INTERFACE

0000	0000	DATA IN	LXI	H000H	: Load 0000 into PC
0001	0001		STX	0000H	: Store PC into 0000
0002	0002		LDAA	PIA0RB	: Load parallel PIA flag
0003	0003		STRA		
0004	0004		STLA	PIA0RB	
0005	0005		STAA	PIA0RB	: Set Port B as input
0006	0006		CLI		
0007	0007		CLAB	#024	
0008	0008		CLAA	#000	
0009	0009	CONVRT	STAB	PIA0RB	: Starts ADC0801
0010	0010		STAA	PIA0RB	
0011	0011		HAL		: Wait for interrupt
0012	0012		LDR	TEMP1	
0013	0013		STX	#0200	: Is final data stored?
0014	0014		BEQ	ENDP	
0015	0015		INX		
0016	0016		STX	TEMP1	
0017	0017		STRA	CONVRT	
0018	0018	INTERR	LDR	TEMP1	
0019	0019		LDAA	PIA0RB	: Read data in
0020	0020		STAA	X	: Store it at X
0021	0021		FTI		
0022	0022	TEMP1	DOB	00200	: Starting address for
0023	0023				: data storage
0024	0024	ENDP	LDR	#0200	: Reinitialize TEMP1
0025	0025		STX	TEMP1	
0026	0026		RTS		: Return from subroutine
0027	0027	PIA0RB	EQU	\$8006	: To user's program
0028	0028	PIA0RB	EQU	\$8007	

using schematic and sample subroutine (DATA IN) to interface (up to) 8 ADC0801's directly to the CPU. This scheme can easily be extended to allow use of more converters. In this configuration the converters (arbitrarily) located at HEX address 5000 in 68000 memory space. To save components, the signal is derived from just one RC pair on the first one. This output drives the other A/Ds.

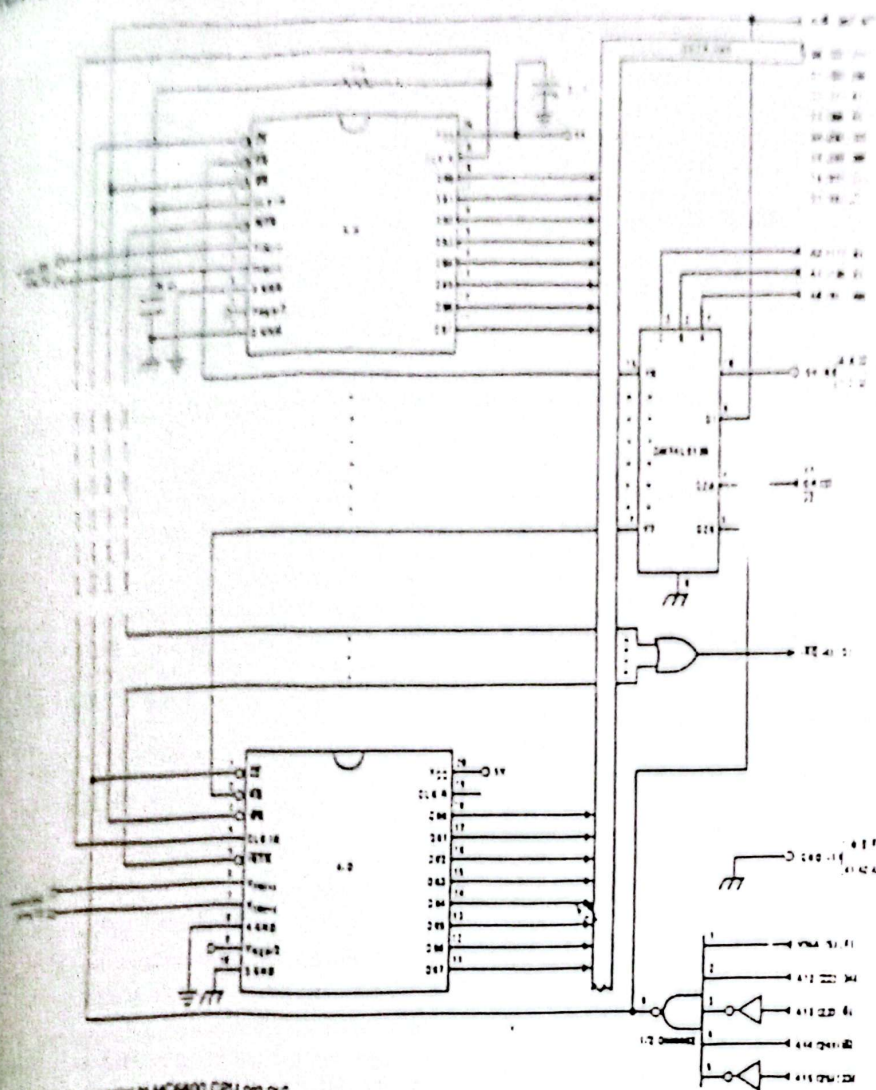
converters are started simultaneously with a STORE command at HEX address 5000. Note that any other HEX address in the form 500X will be decoded by the circuit, as the CS inputs low. This can easily be avoided by a more selective address decoding scheme. All the pins are OR'ed together to insure that all A/Ds have started their conversion before the microprocessor is in.

Assume, DATA IN, may be called from anywhere in the program. Once called, this routine initializes the

CPU, starts all the converters simultaneously and waits for the interrupt signal. Upon receiving the interrupt, it reads the converters (from HEX addresses 5000 through 5007) and stores the data successively at (arbitrarily chosen) HEX addresses 0200 to 0207, before returning to the user's program. All CPU registers then recover the original data they had before servicing DATA IN.

5.2 Auto-Zeroed Differential Transducer Amplifier and A/D Converter

The differential inputs of the ADC0801 series eliminate the need to perform a differential to single ended conversion for a differential transducer. Thus, one op amp can be eliminated since the differential to single ended conversion is provided by the differential input of the ADC0801 series. In general, a transducer preamp is required to take advantage of the full A/D converter input dynamic range.



Pinout references refer to MC6800 CPU pin out.

Pinout references in brackets refer to standard M6800 system common bus code.

70-115571-25

FIGURE 16. Interfacing Multiple A/Ds in an MC6800 System

SAMPLE PROGRAM FOR FIGURE 16 INTERFACING MULTIPLE A/Ds IN AN MC6800 SYSTEM

ADDRESS	HEX CODE	MNEMONICS	COMMENTS
00	0F 44	DATAIN STX TEMP	; Save Contents of X
01	CE 00 2A	LDX #002A	; Upon $\overline{IRQ}$ LOW CPU
02	FF FF 7B	STX \$FFF8	; Jumps to 002A
03	87 50 00	STAA \$5000	; Starts all A/D's
04	0E	CLI	
05	3E	WAI	; Wait for interrupt
06	CE 50 00	LDX #5000	
07	0F 40	STX INDEX1	; Reset both INDEX
08	CE 02 00	LDX #0200	; land 2 to starting
09	0F 42	STX INDEX2	; addresses
0A	CE 44	LDX TEMP	
0B	39	RTS	; Return from subroutine
0C	DE 40	INTRPT LDX INDEX1	; INDEX1 $\rightarrow$ X
0D	A6 00	LDAA X	; Read data in from A/D at X
0E	0B	INX	; Increment X by one
0F	0F 40	STX INDEX1	; X $\rightarrow$ INDEX1
10	DE 42	LDX INDEX2	; INDEX2 $\rightarrow$ X

Functional Description

SAMPLE PROGRAM FOR FIGURE 18: INTERFACING MULTIPLE A/D IN AN MICRO SYSTEM

ADDRESS	HEX CODE	ASSEMBLY	COMMENTS
0000	1100	LDI R1, 0	DATA START
0001	1100	LDI R1, 0	DATA START
0002	1100	LDI R1, 0	DATA START
0003	1100	LDI R1, 0	DATA START
0004	1100	LDI R1, 0	DATA START
0005	1100	LDI R1, 0	DATA START
0006	1100	LDI R1, 0	DATA START
0007	1100	LDI R1, 0	DATA START
0008	1100	LDI R1, 0	DATA START
0009	1100	LDI R1, 0	DATA START
000A	1100	LDI R1, 0	DATA START
000B	1100	LDI R1, 0	DATA START
000C	1100	LDI R1, 0	DATA START
000D	1100	LDI R1, 0	DATA START
000E	1100	LDI R1, 0	DATA START
000F	1100	LDI R1, 0	DATA START

Figure 18. The microcontroller is shown subroutines and macros. The stack pointer must be initialized in the user's program.

When the ADC input signals, a major system error is the offset voltage of the amplifiers used for the ADC. Figure 17 is a gain of 100 differential preamp. Offset voltage errors will be cancelled by a zeroing routine which is performed by the INS8080A microprocessor. The total allowable input offset voltage error is $\pm 50 \mu V$ for $1/2$ LSB error. This would require very precise amplifiers. The expression for differential output voltage of the preamp is:

$$V_{out} = \underbrace{(V_{in1} - V_{in2})}_{\text{SIGNAL}} \underbrace{\left(1 - \frac{R_2}{R_1}\right)}_{\text{GAIN}}$$

$$\underbrace{(V_{OS1} - V_{OS2})}_{\text{DC ERROR TERM}} \underbrace{\left(1 - \frac{R_2}{R_1}\right)}_{\text{GAIN}}$$

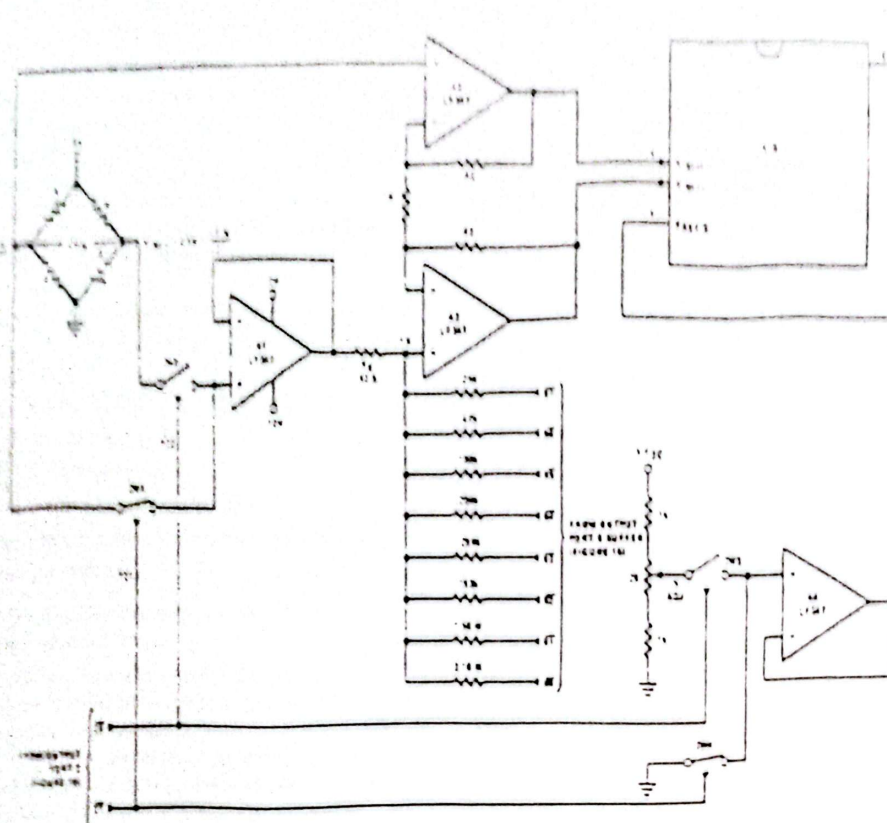
is the current through resistor R_x . All of the offset can be cancelled by making $\pm I_x R_x = V_{OS1} - V_{OS2}$. This is the principle of this auto-zeroing.

INS8080A uses the 2 I/O ports of an INS8255 Programmable Peripheral Interface (PPI) to control the auto-zero routine from the ADC0801 as shown in Figure 18.

INS8080A is programmed for basic I/O operation (mode 0) Port A being an input port and Ports B and C being output ports. Two bits of Port C are used to alternately open or close the 2 switches at the input of the preamp. Switch

SW1 is closed to force the preamp's differential input to be zero during the zeroing subroutine and then opened and SW2 is then closed for conversion of the actual differential input signal. Using 2 switches in this manner eliminates concern for the ON resistance of the switches as they must conduct only the input bias current of the input amplifiers.

Output Port B is used as a successive approximation register by the 8080 and the binary scaled resistors in series with each output bit create a D/A converter. During the zeroing subroutine, the voltage at V_x increases or decreases as required to make the differential output voltage equal to zero. This is accomplished by ensuring that the voltage at the output of A1 is approximately 2.5V so that a logic "1" (5V) on any output of Port B will source current into node V_x thus raising the voltage at V_x and making the output differential more negative. Conversely, a logic "0" (0V) will pull current out of node V_x and decrease the voltage, causing the differential output to become more positive. For the resistor values shown, V_x can move ± 12 mV with a resolution of $50 \mu V$, which will null the offset error term to $1/2$ LSB of full-scale for the ADC0801. It is important that the voltage levels that drive the auto-zero resistors be constant. Also, for symmetry, a logic swing of 0V to 5V is convenient. To achieve this, a CMOS buffer is used for the logic output signals of Port B and this CMOS package is powered with a stable 5V source. Buffer amplifier A1 is necessary so that it can source or sink the D/A output current.



U1, U2, U3

Use 14C13034 CMOS analog switches.

Resistors used in the auto-zero section can be $\pm 5\%$ tolerance.

FIGURE 17. Gain of 100 Differential Transducer Preamp

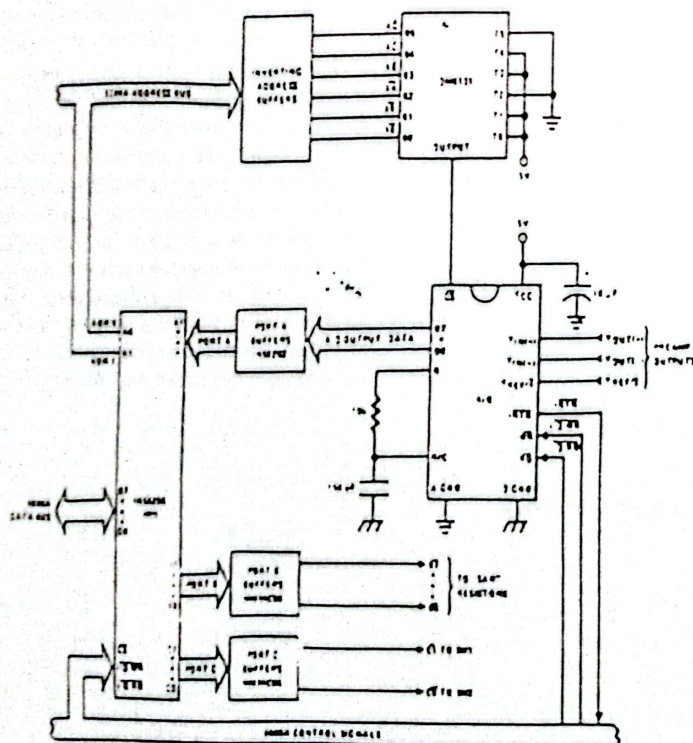


FIGURE 18. Microprocessor Interface Circuitry for Differential Preamp

A new chart for the zeroing subroutine is shown in Figure 19. It must be noted that the ADC0801 series will output an all zero code when it converts a negative input ($V_X(-) = V_{X(0)} - 1$). Also, a logic inversion exists as all of the I/O ports are buffered with inverting gates.

Basically, if the data read is zero, the differential output voltage is negative, so a bit in Port B is cleared to pull V_X more negative which will make the output more positive for the next conversion. If the data read is not zero, the output voltage is positive so a bit in Port B is set to make V_X more positive and the output more negative. This continues for 8 approximations and the differential output eventually converges to within 5 mV of zero.

The actual program is given in Figure 20. All addresses used are compatible with the BLC 80/10 microcomputer system, in particular:

Port A and the ADC0801 are at port address E4

Port B is at port address E5

Port C is at port address E6

PPI control word port is at port address E7

Program Counter automatically goes to ADDR:3C3D upon acknowledgement of an interrupt from the ADC0801

5.3 Multiple A/D Converters in a Z-80 Interrupt Driven Mode

In data acquisition systems where more than one A/D converter (or other peripheral device) will be interrupting program execution of a microprocessor, there is obviously a need for the CPU to determine which device requires servicing. Figure 21 and the accompanying software is a method of determining which of 7 ADC0801 converters has completed a conversion (INT₀ asserted) and is requesting an interrupt. This circuit allows starting the A/D converters in any sequence, but will input and store valid data from the converters with a priority sequence of A/D 1 being read first, A/D 2 second, etc., through A/D 7 which would have the lowest priority for data being read. Only the converters whose INT is asserted will be read.

The key to decoding circuitry is the DM74LS373, 8-bit D type flip-flop. When the Z-80 acknowledges the interrupt, the program is vectored to a data input Z-80 subroutine. This subroutine will read a peripheral status word from the DM74LS373 which contains the logic state of the INT₀ outputs of all the converters. Each converter which initiates an interrupt will place a logic "0" in a unique bit position in the status word and the subroutine will determine the identity of the converter and execute a data read. An identifier word¹ (which indicates which A/D the data came from) is stored in the next sequential memory location above the location of the data so the program can keep track of the identity of the data entered.

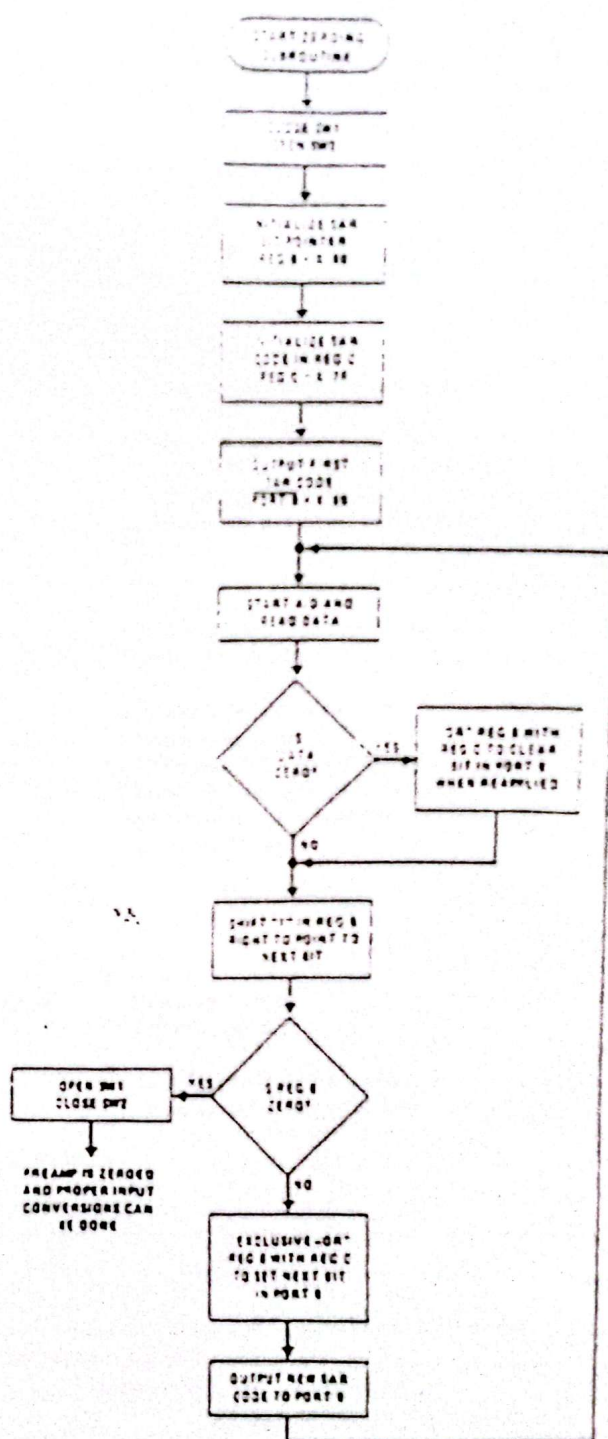


FIGURE 19. Flow Chart for Auto-Zero Routine


```

0000 1E90  MVI B0
0001 01E7  Out Control Port
0002 1E01  MVI H 01
0003 1E04  MVI A, H
0004 1E06  OUT A
0005 1E0C  MVI B 00
0006 1E0F  MVI A 7F
0007 4F  MOV C, A
0008 1E08  OUT B
0009 1E1A1D  LTI SP, DAA
0010 1E04  OUT A
0011 7B  IE
0012 00  NOP
0013 031E3D  JMP Loop
0014 7A  MOV A, D
0015 0600  ADI 00
0016 1A003D  JI Set C
0017 7B  MOV A, B
0018 7600  ORI 00
0019 1F  RAR
0020 7E00  CFI 00
0021 0A373D  JI Done
0022 4F  MOV B, A
0023 03333D  JMP New C
0024 7B  MOV A, C
0025 30  ORA B
0026 4F  MOV C, A
0027 03003D  JMP Shift B
0028 49  XRA C
0029 03003D  JMP Return
0030 4F  MOV B, A
0031 7C  MOV A, H
0032 0E03  XRI 03
0033 03E6  OUT C
0034  *
0035  *
0036  *
0037  *
0038  *
0039  *
0040  *
0041  *
0042  *
0043  *
0044  *
0045  *
0046  *
0047  *
0048  *
0049  *
0050  *
0051  *
0052  *
0053  *
0054  *
0055  *
0056  *
0057  *
0058  *
0059  *
0060  *
0061  *
0062  *
0063  *
0064  *
0065  *
0066  *
0067  *
0068  *
0069  *
0070  *
0071  *
0072  *
0073  *
0074  *
0075  *
0076  *
0077  *
0078  *
0079  *
0080  *
0081  *
0082  *
0083  *
0084  *
0085  *
0086  *
0087  *
0088  *
0089  *
0090  *
0091  *
0092  *
0093  *
0094  *
0095  *
0096  *
0097  *
0098  *
0099  *
0100  *
0101  *
0102  *
0103  *
0104  *
0105  *
0106  *
0107  *
0108  *
0109  *
0110  *
0111  *
0112  *
0113  *
0114  *
0115  *
0116  *
0117  *
0118  *
0119  *
0120  *
0121  *
0122  *
0123  *
0124  *
0125  *
0126  *
0127  *
0128  *
0129  *
0130  *
0131  *
0132  *
0133  *
0134  *
0135  *
0136  *
0137  *
0138  *
0139  *
0140  *
0141  *
0142  *
0143  *
0144  *
0145  *
0146  *
0147  *
0148  *
0149  *
0150  *
0151  *
0152  *
0153  *
0154  *
0155  *
0156  *
0157  *
0158  *
0159  *
0160  *
0161  *
0162  *
0163  *
0164  *
0165  *
0166  *
0167  *
0168  *
0169  *
0170  *
0171  *
0172  *
0173  *
0174  *
0175  *
0176  *
0177  *
0178  *
0179  *
0180  *
0181  *
0182  *
0183  *
0184  *
0185  *
0186  *
0187  *
0188  *
0189  *
0190  *
0191  *
0192  *
0193  *
0194  *
0195  *
0196  *
0197  *
0198  *
0199  *
0200  *
0201  *
0202  *
0203  *
0204  *
0205  *
0206  *
0207  *
0208  *
0209  *
0210  *
0211  *
0212  *
0213  *
0214  *
0215  *
0216  *
0217  *
0218  *
0219  *
0220  *
0221  *
0222  *
0223  *
0224  *
0225  *
0226  *
0227  *
0228  *
0229  *
0230  *
0231  *
0232  *
0233  *
0234  *
0235  *
0236  *
0237  *
0238  *
0239  *
0240  *
0241  *
0242  *
0243  *
0244  *
0245  *
0246  *
0247  *
0248  *
0249  *
0250  *
0251  *
0252  *
0253  *
0254  *
0255  *
0256  *
0257  *
0258  *
0259  *
0260  *
0261  *
0262  *
0263  *
0264  *
0265  *
0266  *
0267  *
0268  *
0269  *
0270  *
0271  *
0272  *
0273  *
0274  *
0275  *
0276  *
0277  *
0278  *
0279  *
0280  *
0281  *
0282  *
0283  *
0284  *
0285  *
0286  *
0287  *
0288  *
0289  *
0290  *
0291  *
0292  *
0293  *
0294  *
0295  *
0296  *
0297  *
0298  *
0299  *
0300  *
0301  *
0302  *
0303  *
0304  *
0305  *
0306  *
0307  *
0308  *
0309  *
0310  *
0311  *
0312  *
0313  *
0314  *
0315  *
0316  *
0317  *
0318  *
0319  *
0320  *
0321  *
0322  *
0323  *
0324  *
0325  *
0326  *
0327  *
0328  *
0329  *
0330  *
0331  *
0332  *
0333  *
0334  *
0335  *
0336  *
0337  *
0338  *
0339  *
0340  *
0341  *
0342  *
0343  *
0344  *
0345  *
0346  *
0347  *
0348  *
0349  *
0350  *
0351  *
0352  *
0353  *
0354  *
0355  *
0356  *
0357  *
0358  *
0359  *
0360  *
0361  *
0362  *
0363  *
0364  *
0365  *
0366  *
0367  *
0368  *
0369  *
0370  *
0371  *
0372  *
0373  *
0374  *
0375  *
0376  *
0377  *
0378  *
0379  *
0380  *
0381  *
0382  *
0383  *
0384  *
0385  *
0386  *
0387  *
0388  *
0389  *
0390  *
0391  *
0392  *
0393  *
0394  *
0395  *
0396  *
0397  *
0398  *
0399  *
0400  *
0401  *
0402  *
0403  *
0404  *
0405  *
0406  *
0407  *
0408  *
0409  *
0410  *
0411  *
0412  *
0413  *
0414  *
0415  *
0416  *
0417  *
0418  *
0419  *
0420  *
0421  *
0422  *
0423  *
0424  *
0425  *
0426  *
0427  *
0428  *
0429  *
0430  *
0431  *
0432  *
0433  *
0434  *
0435  *
0436  *
0437  *
0438  *
0439  *
0440  *
0441  *
0442  *
0443  *
0444  *
0445  *
0446  *
0447  *
0448  *
0449  *
0450  *
0451  *
0452  *
0453  *
0454  *
0455  *
0456  *
0457  *
0458  *
0459  *
0460  *
0461  *
0462  *
0463  *
0464  *
0465  *
0466  *
0467  *
0468  *
0469  *
0470  *
0471  *
0472  *
0473  *
0474  *
0475  *
0476  *
0477  *
0478  *
0479  *
0480  *
0481  *
0482  *
0483  *
0484  *
0485  *
0486  *
0487  *
0488  *
0489  *
0490  *
0491  *
0492  *
0493  *
0494  *
0495  *
0496  *
0497  *
0498  *
0499  *
0500  *
0501  *
0502  *
0503  *
0504  *
0505  *
0506  *
0507  *
0508  *
0509  *
0510  *
0511  *
0512  *
0513  *
0514  *
0515  *
0516  *
0517  *
0518  *
0519  *
0520  *
0521  *
0522  *
0523  *
0524  *
0525  *
0526  *
0527  *
0528  *
0529  *
0530  *
0531  *
0532  *
0533  *
0534  *
0535  *
0536  *
0537  *
0538  *
0539  *
0540  *
0541  *
0542  *
0543  *
0544  *
0545  *
0546  *
0547  *
0548  *
0549  *
0550  *
0551  *
0552  *
0553  *
0554  *
0555  *
0556  *
0557  *
0558  *
0559  *
0560  *
0561  *
0562  *
0563  *
0564  *
0565  *
0566  *
0567  *
0568  *
0569  *
0570  *
0571  *
0572  *
0573  *
0574  *
0575  *
0576  *
0577  *
0578  *
0579  *
0580  *
0581  *
0582  *
0583  *
0584  *
0585  *
0586  *
0587  *
0588  *
0589  *
0590  *
0591  *
0592  *
0593  *
0594  *
0595  *
0596  *
0597  *
0598  *
0599  *
0600  *
0601  *
0602  *
0603  *
0604  *
0605  *
0606  *
0607  *
0608  *
0609  *
0610  *
0611  *
0612  *
0613  *
0614  *
0615  *
0616  *
0617  *
0618  *
0619  *
0620  *
0621  *
0622  *
0623  *
0624  *
0625  *
0626  *
0627  *
0628  *
0629  *
0630  *
0631  *
0632  *
0633  *
0634  *
0635  *
0636  *
0637  *
0638  *
0639  *
0640  *
0641  *
0642  *
0643  *
0644  *
0645  *
0646  *
0647  *
0648  *
0649  *
0650  *
0651  *
0652  *
0653  *
0654  *
0655  *
0656  *
0657  *
0658  *
0659  *
0660  *
0661  *
0662  *
0663  *
0664  *
0665  *
0666  *
0667  *
0668  *
0669  *
0670  *
0671  *
0672  *
0673  *
0674  *
0675  *
0676  *
0677  *
0678  *
0679  *
0680  *
0681  *
0682  *
0683  *
0684  *
0685  *
0686  *
0687  *
0688  *
0689  *
0690  *
0691  *
0692  *
0693  *
0694  *
0695  *
0696  *
0697  *
0698  *
0699  *
0700  *
0701  *
0702  *
0703  *
0704  *
0705  *
0706  *
0707  *
0708  *
0709  *
0710  *
0711  *
0712  *
0713  *
0714  *
0715  *
0716  *
0717  *
0718  *
0719  *
0720  *
0721  *
0722  *
0723  *
0724  *
0725  *
0726  *
0727  *
0728  *
0729  *
0730  *
0731  *
0732  *
0733  *
0734  *
0735  *
0736  *
0737  *
0738  *
0739  *
0740  *
0741  *
0742  *
0743  *
0744  *
0745  *
0746  *
0747  *
0748  *
0749  *
0750  *
0751  *
0752  *
0753  *
0754  *
0755  *
0756  *
0757  *
0758  *
0759  *
0760  *
0761  *
0762  *
0763  *
0764  *
0765  *
0766  *
0767  *
0768  *
0769  *
0770  *
0771  *
0772  *
0773  *
0774  *
0775  *
0776  *
0777  *
0778  *
0779  *
0780  *
0781  *
0782  *
0783  *
0784  *
0785  *
0786  *
0787  *
0788  *
0789  *
0790  *
0791  *
0792  *
0793  *
0794  *
0795  *
0796  *
0797  *
0798  *
0799  *
0800  *
0801  *
0802  *
0803  *
0804  *
0805  *
0806  *
0807  *
0808  *
0809  *
0810  *
0811  *
0812  *
0813  *
0814  *
0815  *
0816  *
0817  *
0818  *
0819  *
0820  *
0821  *
0822  *
0823  *
0824  *
0825  *
0826  *
0827  *
0828  *
0829  *
0830  *
0831  *
0832  *
0833  *
0834  *
0835  *
0836  *
0837  *
0838  *
0839  *
0840  *
0841  *
0842  *
0843  *
0844  *
0845  *
0846  *
0847  *
0848  *
0849  *
0850  *
0851  *
0852  *
0853  *
0854  *
0855  *
0856  *
0857  *
0858  *
0859  *
0860  *
0861  *
0862  *
0863  *
0864  *
0865  *
0866  *
0867  *
0868  *
0869  *
0870  *
0871  *
0872  *
0873  *
0874  *
0875  *
0876  *
0877  *
0878  *
0879  *
0880  *
0881  *
0882  *
0883  *
0884  *
0885  *
0886  *
0887  *
0888  *
0889  *
0890  *
0891  *
0892  *
0893  *
0894  *
0895  *
0896  *
0897  *
0898  *
0899  *
0900  *
0901  *
0902  *
0903  *
0904  *
0905  *
0906  *
0907  *
0908  *
0909  *
0910  *
0911  *
0912  *
0913  *
0914  *
0915  *
0916  *
0917  *
0918  *
0919  *
0920  *
0921  *
0922  *
0923  *
0924  *
0925  *
0926  *
0927  *
0928  *
0929  *
0930  *
0931  *
0932  *
0933  *
0934  *
0935  *
0936  *
0937  *
0938  *
0939  *
0940  *
0941  *
0942  *
0943  *
0944  *
0945  *
0946  *
0947  *
0948  *
0949  *
0950  *
0951  *
0952  *
0953  *
0954  *
0955  *
0956  *
0957  *
0958  *
0959  *
0960  *
0961  *
0962  *
0963  *
0964  *
0965  *
0966  *
0967  *
0968  *
0969  *
0970  *
0971  *
0972  *
0973  *
0974  *
0975  *
0976  *
0977  *
0978  *
0979  *
0980  *
0981  *
0982  *
0983  *
0984  *
0985  *
0986  *
0987  *
0988  *
0989  *
0990  *
0991  *
0992  *
0993  *
0994  *
0995  *
0996  *
0997  *
0998  *
0999  *

```

FIGURE 20. Software for Auto-Zeroed Differential A/D

3

Multiple A/D Converters in a Z-80® Interrupt Driven Mode (Continued)

Following notes apply:

1. It is assumed that the CPU automatically performs a RST instruction when a valid interrupt is acknowledged (CPU in interrupt mode 1). Hence, the subroutine starting address of X0038.

2. Address bus from the Z-80 and the data bus to the Z-80 are assumed to be inverted by bus drivers.

3. Data and identifying words will be stored in sequential memory locations starting at the arbitrarily chosen address X 0E00.

4. Stack pointer must be dimensioned in the main program as the RST 7 instruction automatically pushes the P onto the stack and the subroutine uses an additional stack addresses.

5) The peripherals of concern are mapped into I/O space with the following port assignments:

HEX PORT ADDRESS	PERIPHERAL
00	MM74C374 8-bit flip-flop
01	A/D 1
02	A/D 2
03	A/D 3
04	A/D 4
05	A/D 5
06	A/D 6
07	A/D 7

This port address also serves as the A/D identifying word in the program.

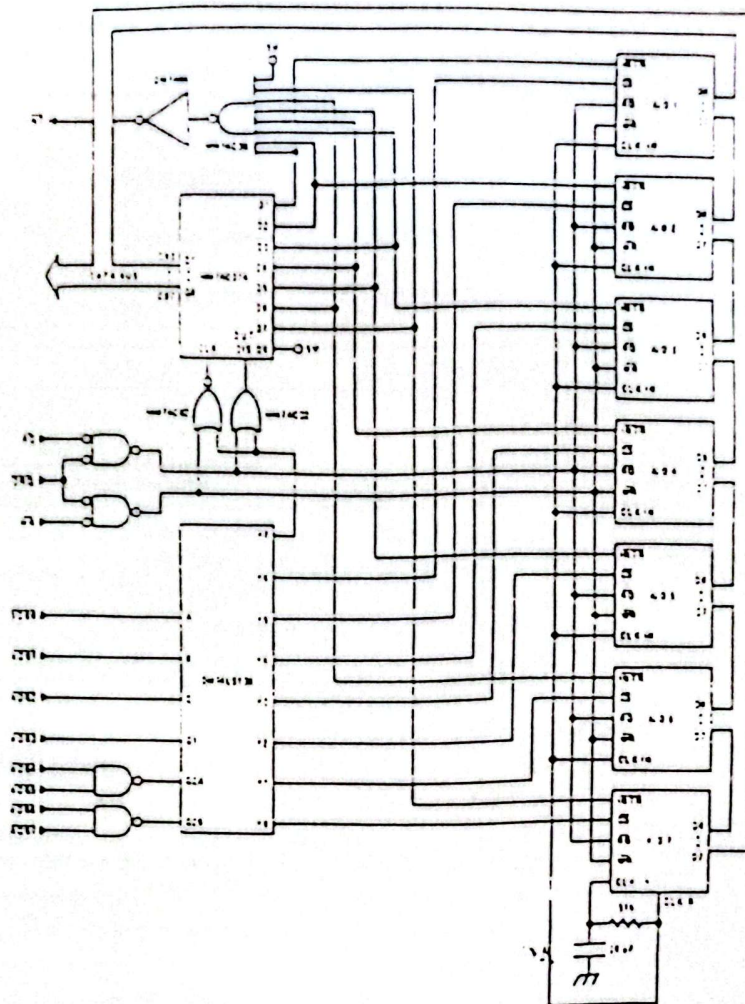


FIGURE 21. Multiple A/Ds with Z-80 Type Microprocessor

INTERRUPT SERVICING SUBROUTINE

LOC	OBJ CODE	SOURCE STATEMENT	COMMENT
0038	E5	PUSH HL	: Save contents of all registers affected by
0039	C5	PUSH BC	: this subroutine.
003A	F5	PUSH AF	: Assumed INT mode 1 earlier set.
003B	21 00 3E	LD (HL), X3E00	: Initialize memory pointer where data will be stored.
003E	0E 01	LD C, X01	: C register will be port ADDR of A/D converters.
0040	D3 00	OUT X00, A	: Load peripheral status word into 8-bit latch.
0042	D3 00	IN A, X00	: Load status word into accumulator.
0044	47	LD B, A	: Save the status word.
0045	79	TEST LD A, C	: Test to see if the status of all A/D's have
0046	7E 08	CP, X08	: been checked. If so, exit subroutine
0048	CA 60 00	JPL, DONE	
0049	78	LD A, B	: Test a single bit in status word by looking for
004C	1F	RRA	: a '1' to be rotated into the CARRY (an INT
004D	47	LD B, A	: is loaded as a '1'). If CARRY is set then load
004E	0A 55 00	JPC, LOAD	: contents of A/D at port ADDR in C register.
0051	0C	INC C	: If CARRY is not set, increment C register to point
0052	C3 45 00	JP, TEST	: to next A/D, then test next bit in status word.
0055	ED 78	IN A, (C)	: Read data from interrupting A/D and invert
0057	EE FF	XOR FF	: the data.
0059	77	LD (HL), A	: Store the data
005A	2C	INCL	
005B	71	LD (HL), C	: Store A/D identifier (A/D port ADDR).
005C	2C	INCL	
005D	C3 51 00	JP, NEXT	: Test next bit in status word.
0060	F1	POP AF	: Re-establish all registers as they were
0061	C1	POP BC	: before the interrupt.
0062	E1	POP HL	
0063	C9	RET	: Return to original program

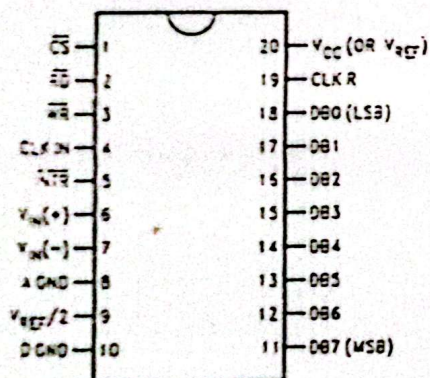
Ordering Information

TEMP RANGE	0°C TO 70°C	0°C TO 70°C	0°C TO 70°C	-40°C TO -85°C
ERROR	$\pm 1/4$ Bit Adjusted			ADC0801LCN
	$\pm 1/2$ Bit Unadjusted	ADC0802LCWM	ADC0802LCV	ADC0802LCN
	$\pm 1/2$ Bit Adjusted	ADC0803LCWM	ADC0803LCV	ADC0803LCN
	± 1 Bit Unadjusted	ADC0804LCWM	ADC0804LCV	ADC0804LCN
				ADC0805LCN
PACKAGE OUTLINE	M20B—Small Outline	J20A—Chip Carrier	N20A—Molded DIP	

TEMP RANGE	-40°C TO -85°C	-55°C TO -125°C
ERROR	$\pm 1/4$ Bit Adjusted	ADC0801LCJ
	$\pm 1/2$ Bit Unadjusted	ADC0802LCJ
	$\pm 1/2$ Bit Adjusted	ADC0803LCJ
	± 1 Bit Unadjusted	ADC0804LCJ
PACKAGE OUTLINE	J20A—Cavity DIP	J20A—Cavity DIP

Connection Diagrams

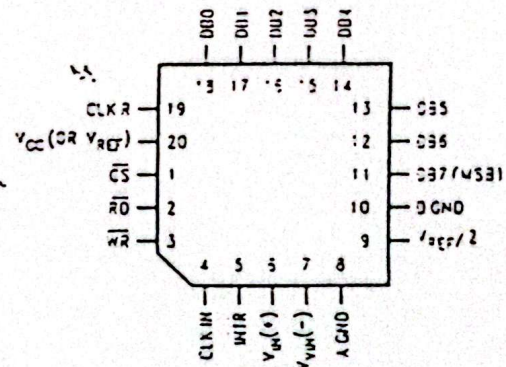
ADC080X
Dual-In-Line and Small Outline (SO) Packages



TL/M/5671-30

See Ordering Information

ADC080X
Molded Chip Carrier (PCC) Package



TL/M/5671-32