

SKRIPSI

PERANCANGAN DAN PEMBUATAN SISTEM PEREKAM TANGGAPAN FREKUENSI UNTUK MIKROFON AUDIO BERBASIS PC



Disusun Oleh :
Fauzy Chairul Alam
NIM : 98.17.047



**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2009**

LEMBAR PERSETUJUAN

**PERANCANGAN DAN PEMBUATAN SISTEM
PEREKAM TANGGAPAN FREKUENSI UNTUK
MIKROFON AUDIO BERBASIS PC**

SKRIPSI

*Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Elektro S-1 Konsentrasi Teknik Elektronika*

Disusun Oleh :

Fauzy Chairul Alam

NIM : 98.17.047

Diperiksa dan Disetujui

Mengetahui

Ketua Jurusan Teknik Elektro S-1



Ir. F. Yudi Limpraptono, MT

NIP.Y.1039500274

Dosen Pembimbing

Ir. F. Yudi Limpraptono, MT

NIP.Y.1039500274

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG
2009**



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Raya Karanglo Km.2
Malang

BERITA ACARA UJIAN SKRIPSI

Nama : Fauzy Chairul Alam
N.I.M : 98.17.047
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : **“ PERANCANGAN DAN PEMBUATAN SISTEM PEREKAM
TANGGAPAN FREKUENSI UNTUK MIKROFON AUDIO
BERBASIS PC “**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) Pada :

Hari : Senin
Tanggal : 23 Maret 2009
Dengan Nilai : 72.95 (B+) *By*

Panitia Majelis Penguji :



Ketua Majelis Penguji

(Ir. Sidik Noertjahdjono, MT)
NIP.Y.1028700163

Sekretaris Majelis Penguji

(Ir. F. Yudi Limpraptono, MT)
NIP.Y.1039500274

Anggota Penguji :

Penguji I

(Joseph Dedy Irawan, ST, MT)
NIP. 132315178

Penguji II

(Setvohadi, ST)
NIP. 1039700309

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat ALLAH SWT. karena berkat Rahmat dan Hidayah-Nyalah Skripsi dan Tugas Akhir ini dapat diselesaikan dengan baik.

Pada awal pelaksanaan sampai pada penyelesaian skripsi ini, penulis telah banyak memperoleh bimbingan, bantuan dan saran yang sangat bermanfaat, baik secara moril maupun materiil dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

1. Prof. Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
 2. Bapak Ir. Sidik Noertjahjono, MT, selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
 3. Bapak Ir. F Yudi Limpraptono MT, selaku Ketua Jurusan Teknik Elektro Program Studi Teknik Elektronika Institut Teknologi Nasional Malang.
 4. Bapak Ir. F Yudi Limpraptono MT, selaku Dosen Pembimbing, yang telah banyak meluangkan waktu, pikiran, arahan dan tuntunan yang progresif dalam penyusunan skripsi ini.
 5. Bapak Ir. Yusuf Ismail Nahkoda MT, selaku Sekretaris Jurusan Jurusan Teknik Elektro Program Studi Teknik Elektronika Institut Teknologi Nasional Malang.
 6. Seluruh rekan-rekan yang telah membantu penulis dalam menyelesaikan skripsi ini.
-

Akhirnya penulis menyadari bahwa penulisan skripsi ini masih ada kekurangan-kekurangan, namun demikian penulis mengharapkan pandangan-pandangan dan saran-saran untuk menyempurnakan dalam pembuatan alat serta dalam penulisan skripsi ini, baik dari pihak mahasiswa Teknik Elektronika khususnya, maupun dari dari pihak lain yang menggunakan skripsi ini sebagai bahan acuan dan referensi.

Semoga ALLAH SWT. Selalu melimpahkan Taufiq dan Hidayah-Nya kepada semua pihak yang telah membantu adalam penyusunan skripsi ini.

Malang, Mei 2009

Penulis

ABSTRAK

Penulis : FAUZI CHAIRUL ALAM

NIM : 98.17.047

PERANCANGAN DAN PEMBUATAN SISTEM PEREKAM TANGGAPAN FREKUENSI UNTUK MIKROFON AUDIO BERBASIS PC

Perkembangan teknologi komunikasi seperti telepon dan perkembangan di sektor hiburan seperti televisi, radio dan musik dewasa ini sangat pesat. Sektor komunikasi dan hiburan ini sangat memerlukan peralatan perekam dalam aplikasinya. Penyusunan skripsi ini bertujuan untuk mendapatkan hasil perancangan dan pembuatan sistem perekam tanggapan frekuensi untuk mikrofon audio berbasis PC.

Penyusunan skripsi dilakukan dengan merancang dan membuat sistem perekam tanggapan frekuensi. Perancangan dan pembuatan meliputi hardware dan software. Hardware meliputi komponen ADC 0804, amplifier, mikrofon jenis dinamik yang digerakkan dengan motor stepper. Perangkat ini dihubungkan ke computer melalui port serial dengan komponen ICL232 dan RS232. Bahasa pemrograman yang digunakan di computer adalah bahasa Delphi. Hasil pembuatan alat diuji coba untuk mengetahui kinerja system yang telah dirancang.

Hasil pengujian menunjukkan bahwa sistem dapat bekerja dengan baik. Mikrofon dapat menangkap sinyal audio yang diberikan. Kelemahannya adalah jika mikrofon berlawanan arah dengan sumber suara maka sensitivitasnya menjadi kecil.

DAFTAR ISI

LEMBAR PERSETUJUAN	
KATA PENGANTAR	
ABSTRAK	
DAFTAR ISI	
DAFTAR GAMBAR	
DAFTAR TABEL	
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	3
1.4. Tujuan	3
1.5. Manfaat	3
1.6. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	5
2.1. Pendahuluan	5
2.2. Gelombang/Sinyal	5
2.2.1. Gelombang Suara	7
2.2.2. Sinyal Analog	7
2.2.3. Sinyal Digital	9
2.3. Analog Digital Converter (ADC) 0804	10
2.4. Port Serial	13

2.4.1. Komunikasi Serial Asinkron	14
2.4.2. Komunikasi Serial Sinkron	15
2.4.3. Transmisi Data Serial	16
2.4.4. Transmisi Data Serial Asinkron	17
2.5. RS 232	19
2.6. Mikrokontroler AT89C51	20
2.6.1. Arsitektur AT89C51	21
2.6.2. Konfigurasi Pin	22
2.7. Mikروفon	25
2.8. Motor Stepper	26
2.9. Delphi	27
BAB III PERENCANAAN DAN PEMBUATAN ALAT	30
3.1. Pendahuluan	30
3.2. Blok Diagram Sistem	31
3.3. Perencanaan Perangkat Keras	33
3.3.1. Sistem Minimum ADC 0804	33
3.3.2. Koneksi ICL 232 dan RS 232	34
3.3.3. Perencanaan Timing Diagram	35
3.4. Perencanaan Motor Stepper	37
3.5. Perencanaan Perangkat Lunak	39
BAB IV PENGUJIAN SISTEM	42
4.1. Pendahuluan	42
4.2. Pengujian rangkaian ADC 0804	42

4.3. Pengujian Sistem	46
BAB V KESIMPULAN DAN SARAN	50
5.1. Kesimpulan	50
5.2. Saran	50
DAFTAR PUSTAKA	
LAMPIRAN LIST PROGRAM	

DAFTAR GAMBAR

Gambar 2.1 : Skema matematis sebuah gelombang	6
Gambar 2.2 : Macam gelombang	6
Gambar 2.3 : Sinyal analog	8
Gambar 2.4 : Tegangan pada sinyal analog	8
Gambar 2.5 : Sinyal digital	9
Gambar 2.6 : Tegangan pada sinyal digital	10
Gambar 2.7 : Skema dan foto ADC 0804	11
Gambar 2.8 : Sistem minimum ADC 0804	13
Gambar 2.9 : Sinkronisasi Awal Akhir	14
Gambar 2.10 : Aliran Data Sinkron	15
Gambar 2.11: Transmisi Data Serial	16
Gambar 2.12 : Transmisi Data Serial Asinkron	17
Gambar 2.13 : Komunikasi Half Duplex	18
Gambar 2.14 : Komunikasi Full Duplex	18
Gambar 2.15 : Konfigurasi Pin Konektor DB9	19
Gambar 2.16 : Mikrokontroler AT89C51	20
Gambar 2.17 : Arsitektur mikrokontroler AT89C51	21
Gambar 2.18 : Konfigurasi pin AT89C51	22
Gambar 2.19 : Rangkaian Driver Motor Stepper	27
Gambar 3.1: Blok diagram sistem	31
Gambar 3.2 : Diagram rangkaian sistem perekam	31

Gambar 3.3 : Sistem minimum ADC 0804	32
Gambar 3.4 : Diagram fungsi ICL 232	34
Gambar 3.5 : Koneksi ICL 232 dan RS 232	34
Gambar 3.6 : Timing diagram	35
Gambar 3.7 : Output enable dan Reset INTR	36
Gambar 3.8 : Skema motor stepper	37
Gambar 3.9 : Rangkaian driver motor stepper	39
Gambar 3.10 : Flow chart untuk mikrokontroler	40
Gambar 3.11 : Flow chart untuk PC	41
Gambar 4.1 : Pengujian Sub Sistem ADC	43
Gambar 4.2 : Foto alat	46
Gambar 4.3 : Skema pengujian	47
Gambar 4.4 : Hasil pengujian	48

DAFTAR TABEL

Tabel 2.1 : Pin RS232 pada Konektor DB9	20
Tabel 2.2 : Fungsi khusus port 3 mikrokontroler AT89C51	23
Tabel 4.1: Hasil Pengujian ADC	44

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dewasa ini, teknologi audio berkembang sangat pesat. Perkembangan ini dapat dilihat pada perkembangan teknologi komunikasi seperti telepon dan perkembangan di sektor hiburan seperti televisi, radio dan musik. Dengan banyaknya perkembangan ini maka peralatan perekam sangat dibutuhkan untuk keperluan dokumentasi suara yang ada. Sektor komunikasi dan hiburan sangat memerlukan peralatan perekam dalam aplikasinya.

Alat perekaman dapat terdiri dari berbagai bentuk dan ukuran, dari tape kaset sederhana hingga alat perekaman multitrack digital yang lebih maju. Hasil perekaman suara sangat tergantung pada peralatan yang digunakan. Misalnya untuk rekaman musik yang baik maka tergantung pada mikrofon, alat perekam yang digunakan serta kemampuan atau keandalan peralatan-peralatan tersebut. Kemampuan peralatan perekam salah satunya ditunjukkan oleh besaran frekuensi operasi peralatan tersebut.

Dalam rangkaian sound system, mikrofon merupakan komponen terdepan terutama dalam proses atau pekerjaan perekaman (*recording*). Pemilihan jenis microphone, disamping kualitas microphone itu sendiri, merupakan salah satu penentu kualitas rekaman. Untuk itu diperlukan perancangan sound sistem harus diawali dengan penentuan kemampuan mikrofonnya.

Berdasarkan uraian latar belakang di atas maka penulis menyusun skripsi berjudul Perancangan Dan Pembuatan Sistem Perekam Tanggapan Frekuensi Untuk Mikrofon Audio Berbasis PC ini.

1.2. Rumusan Masalah

Rumusan masalah skripsi ini adalah :

Merancang dan membuat sistem perekam tanggapan frekuensi untuk mikrofon audio berbasis PC.

1.3. Tujuan

Tujuan penulisan skripsi ini adalah :

Merancang dan membuat sistem perekam tanggapan frekuensi untuk mikrofon audio berbasis PC.

1.4. Batasan Masalah

Batasan masalah penulisan skripsi ini adalah :

1. Analog to Digital Converter yang digunakan adalah ADC 0820.
2. Perangkat lunak pada PC menggunakan compiler Borland Delphi 6.0.
3. Tidak membahas komponen input audio yang digunakan dalam pengujian alat.

1.5. Metodologi

Metodologi penyusunan skripsi ini adalah :

1. Studi literatur.

Yaitu aktivitas mengumpulkan referensi atau teori-teori yang sesuai dari buku-buku atau sumber internet.

2. Perancangan sistem.

Dalam hal ini dibahas tentang langkah-langkah perancangan sistem, blok diagram dan spesifikasi komponen sistem.

3. Pembuatan sistem.

Yaitu aktivitas pembuatan sistem yang telah dirancang.

4. Pengujian sistem.

Yaitu aktivitas pengujian sistem untuk mendapatkan kinerja sistem yang telah dibuat.

5. Kesimpulan.

Yaitu ringkasan seluruh hasil perancangan, pembuatan dan pengujian sistem.

1.6. Sistematika Penulisan

Sistematika penulisan skripsi ini adalah :

BAB I PENDAHULUAN

Membahas tentang latar belakang, rumusan masalah, tujuan, batasan masalah dan sistematika penulisan.

BAB II TEORI PENUNJANG

Membahas teori-teori yang bersesuaian dengan topik skripsi ini.

BAB III PERANCANGAN DAN PEMBUATAN SISTEM

Membahas proses perencanaan, pembuatan dan pengujian alat.

BAB IV PENGUJIAN SISTEM

Membahas kinerja alat yang telah dirancang dan dibuat serta dianalisa berdasarkan hasil yang diperoleh.

BAB V KESIMPULAN

Yaitu ringkasan hasil perancangan, pembuatan dan pengujian alat.

BAB II

TINJAUAN PUSTAKA

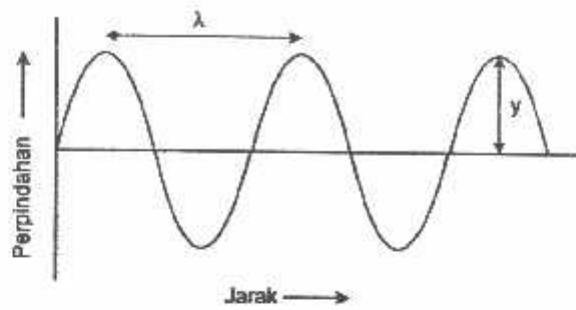
2.1. Pendahuluan

Dalam Bab II ini akan membahas teori-teori yang akan dipergunakan dalam “Perencanaan Dan Pembuatan Sistem Perekam Tanggapan Frekuensi Untuk Mikrofon audio Berbasis PC”, yang meliputi :

- Pembangkitan Sinyal
- Analog Digital Converter (ADC) 0804
- Port Serial
- Borland Delphi Sebagai Software Pemrograman

2.2. Gelombang/Sinyal

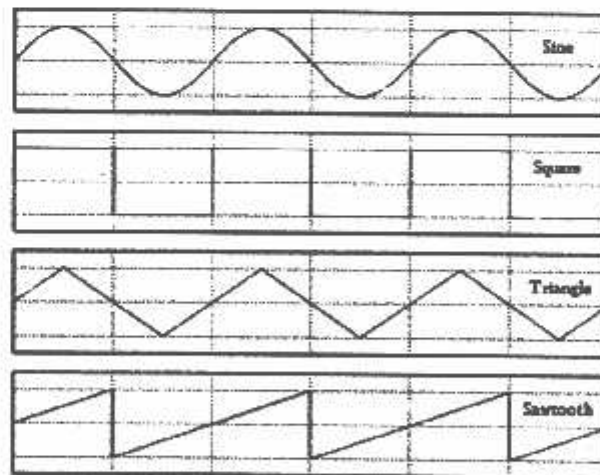
Gelombang didefinisikan sebagai rambatan dalam sebuah medium, umumnya dalam bentuk transfer energi. Energi ini pada umumnya berbentuk getaran sebuah benda. Gelombang terdiri dari berbagai macam bentuk fisik seperti gelombang air, suara, elektromagnet dan sebagainya. Secara matematis, gelombang digambarkan sebagai berikut :



Gambar 2.1 : Skema matematis sebuah gelombang [4]

Parameter gelombang adalah panjang gelombang, amplitudo, perioda dan frekuensi. Panjang gelombang adalah jarak antara dua puncak gelombang. Amplitudo adalah tinggi puncak gelombangnya. Perioda gelombang adalah waktu yang diperlukan untuk satu siklus gelombang. Frekuensi adalah jumlah perioda dalam satu satuan waktu.

Gelombang juga dibedakan berdasarkan bentuknya yaitu gelombang sinus, gelombang kotak, gelombang segitiga dan gelombang gigi gergaji. Macam gelombang ini dapat dilihat pada gambar berikut :



Gambar 2.2 : Macam gelombang [4]

Gelombang sinus merupakan jenis gelombang yang banyak diaplikasikan. Jenis gelombang ini diaplikasikan untuk permasalahan matematika, fisika, pemrosesan sinyal, audio dan rekayasa elektronik. Gelombang lain yang juga banyak digunakan di dunia elektronik adalah gelombang kotak. Gelombang ini digunakan untuk switch sirkuit digital dan dibangkitkan dalam bentuk perangkat logika biner. Gelombang ini digunakan sebagai referensi waktu atau clock signal karena bentuk interval gelombangnya.

2.2.1. Gelombang Suara

Gelombang suara dihasilkan oleh getaran suatu benda di udara. Kecepatan rambat gelombang suara dinyatakan dalam frekuensi dan diukur dalam satuan Hertz (Hz). Batas pendengaran manusia rata-rata adalah 100-15.000 Hz. Parameter suara yang juga penting adalah level atau sering disebut sebagai volume. Kekuatan suara ini diukur dalam satuan decibel (dB).

Di dunia elektronika, gelombang suara banyak dimanfaatkan dalam media komunikasi. Pemanfaatan ini dilakukan dalam bentuk sinyal suara.

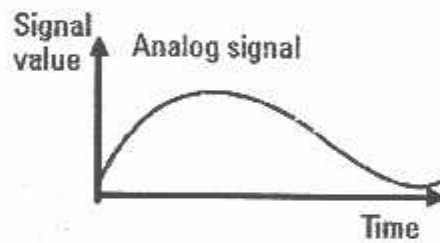
2.2.2. Sinyal Analog

Sinyal analog adalah sinyal data dalam bentuk gelombang yang kontinyu, yang membawa informasi dengan mengubah karakteristik gelombang. Dua parameter/karakteristik terpenting yang dimiliki oleh isyarat analog adalah amplitude dan frekuensi. Isyarat analog biasanya dinyatakan dengan gelombang sinus, mengingat gelombang sinus merupakan dasar untuk semua bentuk isyarat

analog. Hal ini didasarkan kenyataan bahwa berdasarkan analisis fourier, suatu sinyal analog dapat diperoleh dari perpaduan sejumlah gelombang sinus.

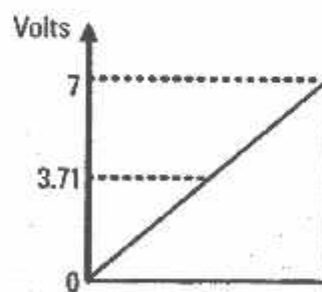
Sinyal analog memiliki jumlah kemungkinan nilai amplitudo yang tak terhingga. Dengan menggunakan sinyal analog, maka jangkauan transmisi data dapat mencapai jarak yang jauh, tetapi sinyal ini mudah terpengaruh oleh *noise*. Gelombang pada sinyal analog yang umumnya berbentuk gelombang sinus memiliki tiga variable dasar, yaitu amplitudo, frekuensi dan *phase*.

- *Amplitudo* merupakan ukuran tinggi rendahnya tegangan dari sinyal analog.
- *Frekuensi* adalah jumlah gelombang sinyal analog dalam satuan detik.
- *Phase* adalah besar sudut dari sinyal analog pada saat tertentu.



Gambar 2.3 : Sinyal analog [5]

Tegangan analog memiliki jumlah kemungkinan level tegangan yang tak hingga seperti dapat dilihat pada gambar berikut :

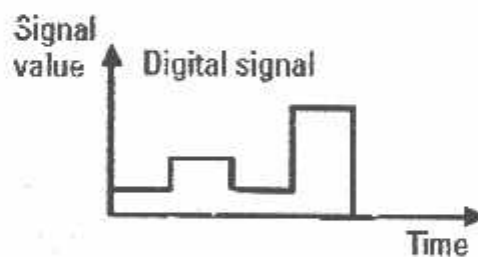


Gambar 2.4 : Tegangan pada sinyal analog [5]

2.2.3. Sinyal Digital

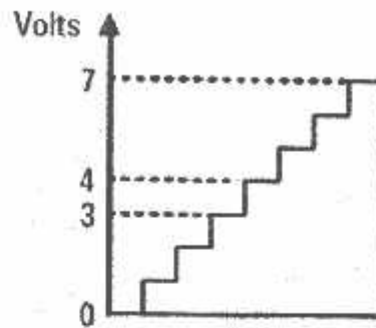
Sinyal digital merupakan sinyal data dalam bentuk pulsa yang dapat mengalami perubahan yang tiba-tiba dan mempunyai besaran 0 dan 1. Sinyal digital hanya memiliki dua keadaan, yaitu 0 dan 1, sehingga tidak mudah terpengaruh oleh derau, tetapi transmisi dengan sinyal digital hanya mencapai jarak jangkauan pengiriman data yang relatif dekat.

Sinyal digital memiliki jumlah kemungkinan nilai amplitudo yang terhingga. Biasanya sinyal ini juga dikenal dengan sinyal diskret. Sinyal yang mempunyai dua keadaan ini biasa disebut dengan bit. Bit merupakan istilah khas pada sinyal digital. Sebuah bit dapat berupa nol (0) atau satu (1). Kemungkinan nilai untuk sebuah bit adalah 2 buah (2^1). Kemungkinan nilai untuk 2 bit adalah sebanyak 4 (2^2), berupa 00, 01, 10, dan 11. Secara umum, jumlah kemungkinan nilai yang terbentuk oleh kombinasi n bit adalah sebesar 2^n buah.



Gambar 2.5 : Sinyal digital [5]

Pada suatu tegangan digital hanya ada beberapa kemungkinan level tegangan, seperti gambar berikut :



Gambar 2.6 : Tegangan pada sinyal digital [5]

Sinyal digital memiliki beberapa keunggulan antara lain :

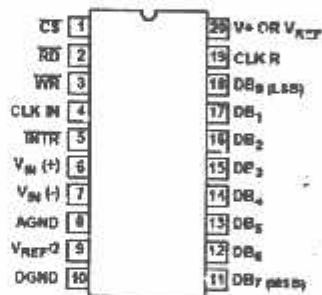
- Teknologi digital menawarkan biaya lebih rendah, keandalan (reliability) yang lebih baik, pemakaian ruang yang lebih kecil, dan konsumsi daya yang rendah
- Teknologi digital membuat kualitas komunikasi tidak tergantung pada jarak
- Teknologi digital lebih toleran terhadap noise
- Jaringan digital ideal untuk komunikasi data yang semakin berkembang
- Teknologi digital memungkinkan pengenalan layanan-layanan baru
- Teknologi digital menyediakan kapasitas transmisi yang besar
- Teknologi digital menawarkan fleksibilitas

2.3. Analog Digital Converter (ADC) 0804.

ADC adalah perangkat untuk mengkonversi sinyal masukan dalam bentuk analog (tegangan) menjadi sinyal keluaran dalam bentuk digital. ADC-08 adalah Modul Analog to Digital Converter 8-bit berkecepatan tinggi. Modul ini cocok untuk aplikasi pendeteksi tegangan dan mengubah sinyal sensor analog menjadi data digital.

Fitur umum ADC 0804 adalah sebagai berikut :

1. Dapat dihubungkan dengan konverter 8080 tanpa penurunan logika antarmuka dengan waktu akses 135 nano detik.
2. Dapat dihubungkan dengan semua mikroprocessor lain atau dioperasikan sendiri.
3. Input tegangan dalam bentuk analog differensial.
4. Input dan output logika sesuai dengan tingkat tegangan spesifik MOS dan TTL.
5. Bekerja dengan tegangan referensi 2,5 Volt.
6. On-chip clock generator.
7. Tegangan input analog antara 0 sampai 5 Volt dengan suplai 5 Volt tunggal/



Gambar 2.7 : Skema dan foto ADC 0804 [7]

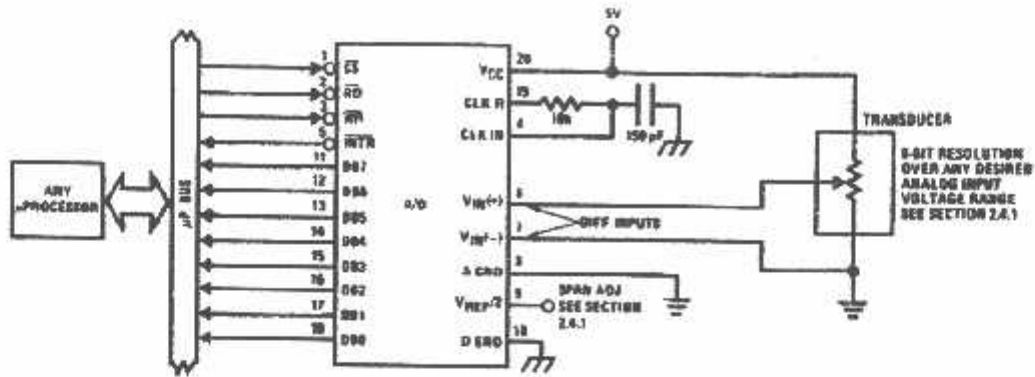
Pada ADC 0804, pin 11-18 merupakan pin keluaran digital yang dapat dihubungkan langsung dengan bus data-alamat. Apabila pin CS atau pin RD dalam keadaan tinggi, pin 11 sampai pin 18 akan mengambang. Apabila CS dan RD rendah keduanya, keluaran digital akan muncul pada saluran keluaran. Untuk memulai suatu konversi, CS harus rendah. Bilamana WR menjadi rendah,

konverter akan mengalami reset dan ketika WR kembali pada keadaan tinggi, konversi segera dimulai.

Pin 5 adalah saluran untuk INTR, sinyal selesai konversi. INTR akan menjadi tinggi pada saat memulai konversi, dan dibuat aktif rendah bilamana konversi telah selesai. Pin 6 dan 7 adalah masukan diferensial yang membandingkan dua masukan sinyal analog. Jenis masukan ini memungkinkan pemilihan bentuk masukan, yaitu pin 7 untuk masukan positif bersisi-tunggal (single-ended positif input), atau pin 6 untuk masukan negatif bersisi-tunggal (single-ended negatif input), atau mengaktifkan kedua pin untuk masukan diferensial.

Piranti ini mempunyai 2 ground, A GND dan D GND yang terletak pada pin 8 dan 10. Keduanya harus digroundkan. Pin 20 disambungkan dengan catu tegangan yang sebesar +5V. Dalam ADC 0804, V_{ref} merupakan tegangan masukan analog maksimum, yaitu tegangan yang menghasilkan suatu keluaran digital maksimum FFH. Bila pin 9 tidak dihubungkan (tidak dipakai), V_{REF} berharga sama dengan tegangan catu V_{CC} . Ini berarti bahwa catu tegangan +5V memberikan jangkauan masukan analog dari 0 sampai +5V bagi masukan positif yang bersisi-tunggal.

Komponen ADC 0804 ini berfungsi untuk merubah sinyal analog menjadi sinyal digital agar dapat diolah oleh PC. Sebagai unit konversi sinyal maka pin-pin ADC 0804 dihubungkan dengan PC dengan skema sebagai berikut :



Gambar 2.8 : Sistem minimum ADC 0804 [10]

ADC 0804 dapat dihubungkan dengan berbagai macam mikrokontroler. Yang dibutuhkan adalah 11 pin yaitu 8 pin untuk data dan 3 pin untuk kontrol.

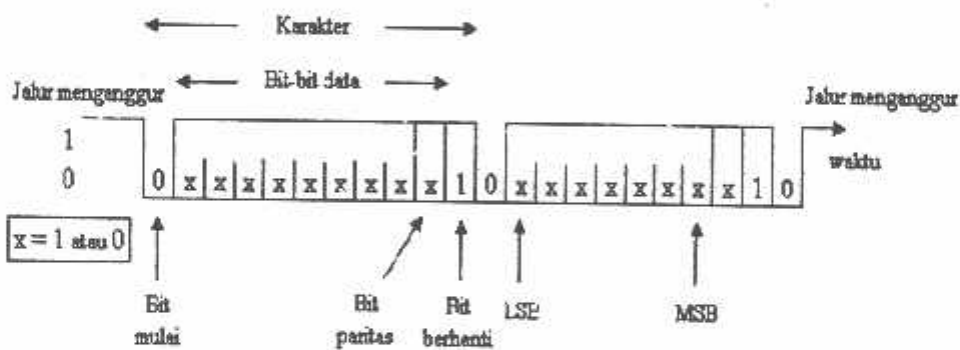
2.4. Port Serial

Akses data dengan menggunakan port serial memiliki beberapa kelebihan dibandingkan dengan menggunakan port paralel. Kelemahan pada komunikasi paralel adalah banyaknya kabel yang harus terhubung. Hal ini menjadi tidak efisien bila kita menggabungkan alat ke komputer (*interfacing*) untuk jarak yang jauh. Oleh karena itu digunakan komunikasi serial untuk mengatasinya.

Pada komunikasi serial, data yang dikirim per bit (data tersebut antri), walaupun mempunyai kelemahan pengiriman data yang lebih lambat dibanding dengan komunikasi paralel, komunikasi serial bisa digunakan untuk jarak yang jauh. Berdasarkan formatnya, pengiriman data komunikasi serial dibedakan menjadi dua bentuk yaitu :

2.4.1. Komunikasi Serial Asinkron

Pada pengiriman data asinkron (tak sinkron), setiap karakter dikirimkan sebagai satu kesatuan (*entity*) bebas, yang berarti bahwa waktu antara pengiriman bit terakhir dari sebuah karakter dan bit pertama dari karakter berikutnya tidak tetap. Pengiriman data tak sinkron lebih sederhana daripada pengiriman sinkron karena hanya isyarat data saja yang dikirimkan. Detak penerima dibangkitkan secara lokal di dalam penerima dan tetap dijaga agar sesuai dengan detak pengirim yang menggunakan bit awal (*start bit*) dan bit akhir (*stop bit*) yang dikirimkan dengan setiap karakter. Penyesuaian detak pengirim dan penerima terjadi per karakter.

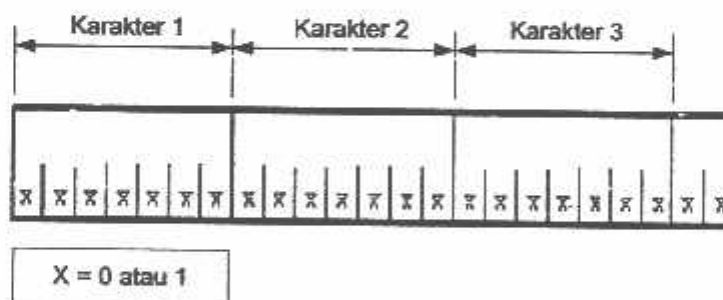


Gambar 2.9 : Sinkronisasi Awal Akhir [2]

Gambar 2.11 menunjukkan sinkronisasi awal-akhir, bit awal dan akhir tidak membawa informasi, tetapi hanya menunjukkan awal dan akhir setiap karakter. Dari gambar dapat dilihat bahwa bit kedelapan disebut bit paritas, diikutsertakan dalam bentuk gelombang tersebut. Bit ini akan dipasang pada 1 atau 0 untuk meyakinkan cacah bit pada setiap karakter adalah genap untuk paritas genap, atau ganjil untuk paritas ganjil.

2.4.2. Komunikasi Serial Sinkron

Pada pengiriman data sinkron sejumlah blok data dikirimkan secara kontinyu tanpa bit awal atau akhir. Detak atau *clock* pada penerima dioperasikan secara kontinyu dan dikunci agar sesuai dengan detak pada pengirim. Untuk mendapat keadaan yang sesuai, informasi pendetakan harus dikirimkan lewat jalur bersama-sama dengan data dan memanfaatkan metode penyandian tertentu sehingga informasi pendekatan dapat diikutsertakan. Data secara kontinyu menunjukkan akan dikirimkan terus menerus tanpa adanya pembatas. Interval waktu antara bit terakhir dari suatu karakter dengan bit pertama dari karakter berikutnya adalah nol atau kelipatan bulat dari periode waktu yang diperlukan untuk mengirimkan sebuah karakter. Jika pada data yang dikirimkan terdapat pembatas, pengirim akan menambah *byte* tambahan untuk mengganti pembatas tersebut. Sehingga tidak diperlukan adanya bit awal dan bit akhir.



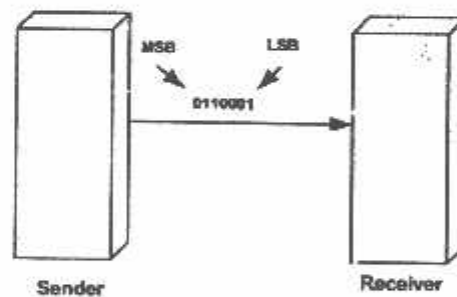
Gambar 2.10 : Aliran Data Sinkron [2]

Gambar 2.12 menunjukkan aliran bit sinkron. Penerima harus memulai pencacahan pada tengah-tengah bit pertama dari karakter pertama, jika tidak akan timbul kesalahan pada isyarat yang diterima. Setelah penyesuaian bit, penerima harus tahu pada kelompok mana bit tersebut akan membentuk karakter

(penyesuaian karakter). Penerima harus memantau data yang diterima setiap bit sampai penerima mengenali pola karakter sinkronisasi. Sehingga penerima dapat mengetahui himpunan bit mana yang membentuk karakter yang pertama dikirimkan. Karakter berikutnya dengan mudah dapat dikirimkan dan dikenali.

2.4.3. Transmisi Data Serial

Transmisi data serial adalah transmisi data yang pengiriman dan penerimaan datanya berurutan tiap bitnya. Jadi kita hanya membutuhkan satu saluran untuk mengirimkan data antar dua perangkat komunikasi. Keuntungan transmisi serial yaitu biaya lebih murah karena hanya membutuhkan satu saluran saja sehingga banyak digunakan untuk komunikasi jarak jauh. Sedangkan kerugiannya adalah kecepatan pengiriman data lebih rendah dibandingkan dengan transmisi paralel.

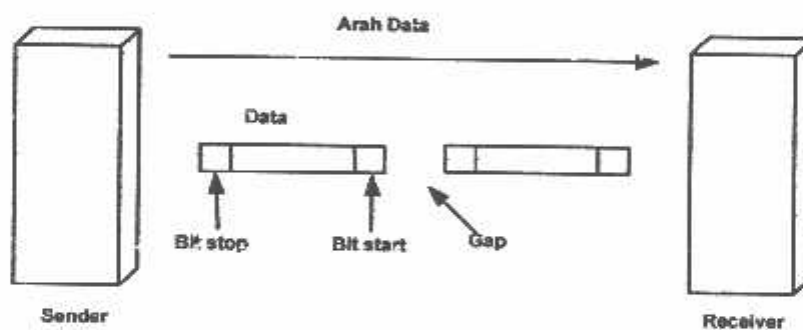


Gambar 2.11: Transmisi Data Serial [2]

Pengiriman data akan dimulai dari LSB (*Least Significant Bit*) dan diakhiri dengan MSB (*Most Significant Bit*). Setiap karakter yang dikirimkan, disusun sesuai dengan suatu urutan dengan bit tertentu. Berdasarkan formatnya, salah satu jenis komunikasi serial yaitu :

2.4.4. Transmisi Data Serial Asinkron

Pada pengiriman data asinkron (tidak sinkron), setiap karakter dikirimkan sebagai satu kesatuan (*entity*) bebas yang berarti bahwa waktu antara pengiriman bit terakhir dari sebuah karakter dan bit pertama dari sebuah karakter berikutnya tidak tetap. Pengiriman data asinkron lebih sederhana daripada pengiriman sinkron karena hanya isyarat data saja yang dikirimkan. Detak penerima dibangkitkan secara lokal di dalam penerima dan tetap dijaga agar sesuai dengan detak pengirim yang menggunakan bit awal (*start bit*) dan bit akhir (*stop bit*) yang dikirimkan setiap karakter dan data yang satu dengan data selanjutnya dipisahkan gap. Penyesuaian detak pengiriman dan penerima terjadi karakter per karakter.



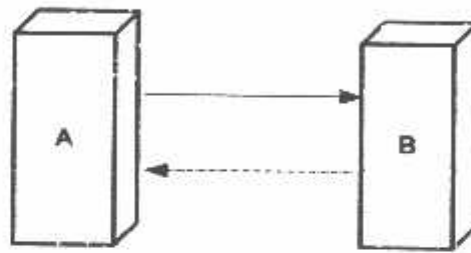
Gambar 2.12 : Transmisi Data Serial Asinkron [2]

Gambar di atas menunjukkan sinkronisasi awal-akhir, bit awal dan akhir tidak membawa informasi, tetapi hanya menunjukkan awal dan akhir setiap karakter. Dari gambar dapat dilihat bahwa bit kedelapan disebut bit paritas, diikutsertakan dalam bentuk gelombang tersebut. Bit ini akan dipasang pada 1 atau 0 untuk meyakinkan cacah bit pada setiap karakter adalah genap untuk paritas genap, atau ganjil untuk paritas ganjil.

Berdasarkan arah komunikasinya, pengiriman data serial dibedakan menjadi :

- **Half Duplex**

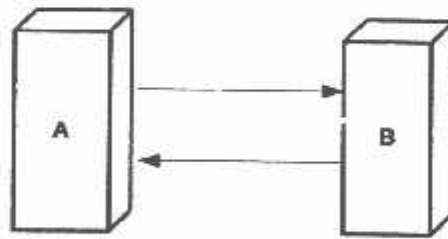
Merupakan sistem komunikasi yang mengirimkan data dalam satu arah. Sistem ini tidak dapat mengirimkan data secara bersamaan sehingga perlu saling menunggu secara bergantian untuk berkomunikasi. Pada gambar di bawah menunjukkan komunikasi *half duplex*.



Gambar 2.13 : Komunikasi Half Duplex [2]

- **Full Duplex**

Merupakan sistem komunikasi yang mengirimkan data dalam dua arah. Sistem ini dapat mengirimkan data secara bersamaan sehingga tidak perlu saling menunggu secara bergantian untuk komunikasi. Gambar di bawah menunjukkan komunikasi *full duplex*.

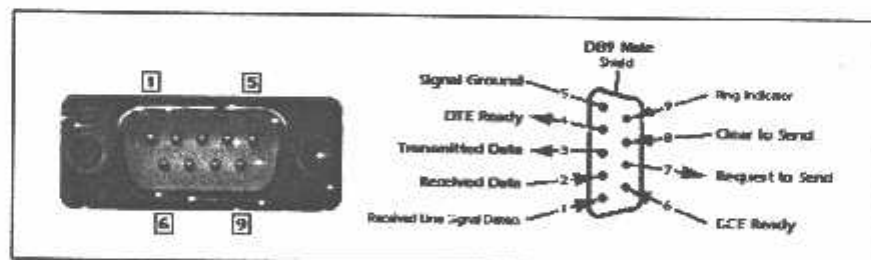


Gambar 2.14 : Komunikasi Full Duplex [2]

2.5. RS 232

RS-232 merupakan seperangkat alat yang berfungsi sebagai *interface* dalam proses transfer data secara serial. Pada RS 232, komunikasi didefinisikan sebagai sebuah metode komunikasi serial asinkron. Serial merupakan informasi yang dikirim satu bit dalam satu waktu. Sedangkan informasi asinkron tidak dikirim pada *time slot* yang sudah dikenal. Pengiriman data dapat dimulai pada beberapa waktu yang diberikan dan itu tugas *receiver* untuk mendeteksi kapan pesan dimulai dan berakhir.

RS 232 menggunakan cara interaksi secara asinkron, dimana sinyal *clock* tidak dikirim bersamaan dengan data. Setiap *word* diselaraskan dengan menggunakan start bit dari RS 232 dan sebuah *clock* internal pada setiap sisi menjaga *timing* yang diinginkan. Di dalam komputer terdapat fasilitas komunikasi serial yang menggunakan standar RS-232, yaitu terletak pada COM1 dan COM2. Kedua fasilitas ini menggunakan konektor DB9 atau DB25 sebagai penghubung dengan piranti luar. Gambar konektor DB9 seperti gambar berikut.



Gambar 2.15 : Konfigurasi Pin Konektor DB9 [11]

Konfigurasi Pin DB9 adalah sebagai berikut :

Tabel 2.1 : Pin RS232 pada Konektor DB9 [11]

Pin 1	Received Line Signal Detector (Data Carrier Detect)
Pin 2	Received Data
Pin 3	Transmit Data
Pin 4	Data Terminal Ready
Pin 5	Signal Ground
Pin 6	Data Set Ready
Pin 7	Request To Send
Pin 8	Clear To Send
Pin 9	Ring Indicator

2.6. Mikrokontroler AT89C51

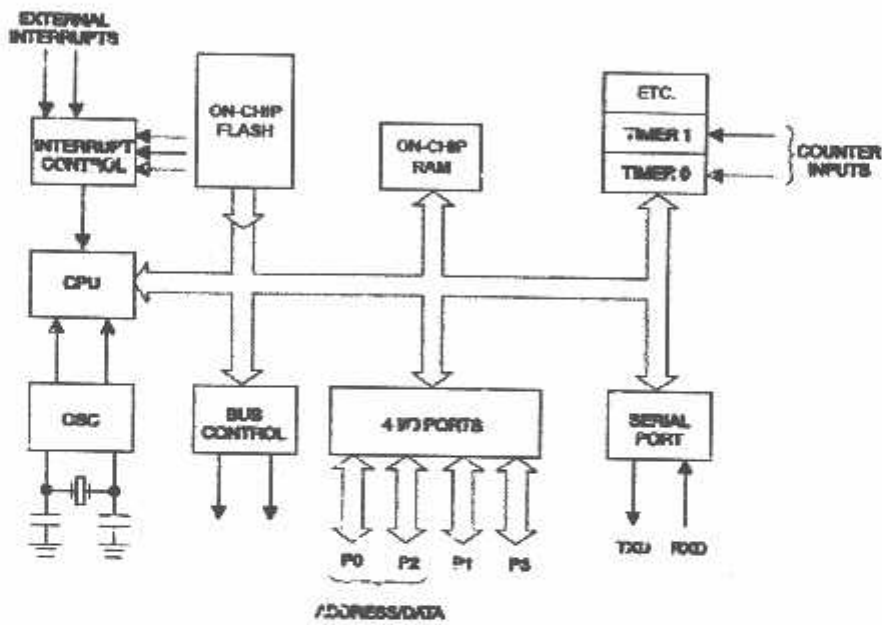
Mikrokontroler AT89C51 ialah mikrokomputer CMOS 8 bit dengan 4KB *Flash Programmable dan Erasable Read Only Memory* (PEROM). Mikrokontroler berteknologi memori *non volatile* kerapatan tinggi dari Atmel ini kompatibel dengan mikrokontroler standar industri MCS-51 (seperti mikrokontroler 8031 yang terkenal dan banyak digunakan beberapa waktu lalu) baik pin kaki IC maupun set instruksinya serta harganya yang cukup murah.



Gambar 2.16 : Mikrokontroler AT89C51 [12]

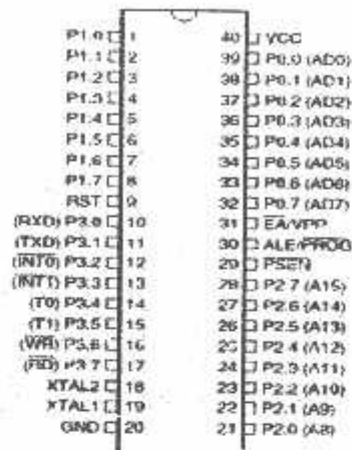
AT89C51 mempunyai memori yang terdiri dari RAM internal 128 byte dengan alamat 00H-7FH dapat diakses menggunakan RAM address register. RAM ini terdiri dari *Register Banks* dengan 8 buah register (R0-R7). Memori lain yaitu 21 buah *Special Function Register* dimulai dari alamat 80H-FFH. RAM ini beda lokasi dengan Flash PEROM dengan alamat 000H -7FFH.

2.6.1. Arsitektur AT89C51



Gambar 2.17 : Arsitektur mikrokontroler AT89C51 [12]

2.6.2. Konfigurasi Pin



Gambar 2.18 : Konfigurasi pin AT89C51 [12]

Deskripsi tiap pin AT89C51 :

1. VCC.

Suplai tegangan

2. GND.

Ground.

3. Port 0.

Port 0 adalah sebuah open-drain bi-directional I/O port 8 bit. Setiap pin dapat menerima 8 input TTL. Ketika nomor 1 menulis data ke pin port 0, pin-pin tersebut dapat digunakan sebagai input impedansi tinggi. Port 0 juga dapat dikonfigurasi menjadi multiplexed alamat atau data orde rendah saat mengakses program eksternal dan memori data. Pada mode ini, port 0 memiliki internal pull up. Port 0 juga menerima kode bytes selama pemrograman Flash, dan mengeluarkan output selama verifikasi program. Eksternal pull up diperlukan selama verifikasi ini.

4. Port 1.

Port 1 adalah sebuah port bi-directional I/O 8 bit dengan internal pull up. Port 1 mengeluarkan atau menerima 4 input TTL. Port 1 juga menerima alamat berorde bytes rendah selama pemrograman Flash dan verifikasi.

5. Port 2.

Port 2 adalah sebuah port bi-directional I/O 8 bit dengan internal pull up. Port 2 mengeluarkan alamat berorde byte tinggi dari memori program eksternal dan selama mengakses data memori menggunakan alamat 16 bit (MOVX @DPTR). Selama mengakses memori data eksternal yang menggunakan alamat 8 bit (MOVX.@RI), port 2 mengeluarkan content dari P2 Special Function Register. Port 2 juga menerima alamat berorde bit tinggi dan beberapa sinyal kontrol selama pemrograman Flash dan verifikasi.

6. Port 3.

Port 2 adalah sebuah port bi-directional I/O 8 bit dengan internal pull up. Port 3 juga menyediakan fungsi-fungsi khusus berikut :

Tabel 2.2 : Fungsi khusus port 3 mikrokontroler AT89C51 [12]

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

Port 3 juga menerima beberapa sinyal kontrol selama pemrograman Flash dan verifikasi.

7. RST.

Input reset.

8. $\overline{\text{ALE/PROG}}$

Address Latch Enable mengeluarkan pulsa untuk mengunci byte rendah dari alamat selama mengakses memori eksternal. Pin ini juga merupakan input program pulsa selama pemrograman Flash. Pada operasi normal, ALE mengeluarkan data pada laju tetap sebesar 1/6 frekuensi osilasi dan dapat digunakan untuk timing eksternal atau tujuan clocking.

9. $\overline{\text{PSEN}}$

Program Store Enable adalah unit pembaca untuk program memori eksternal. Ketika AT89C51 menerima kode dari program eksternal, PSEN diaktifkan dua kali tiap siklusnya, kecuali aktivasi ini dimatikan.

10. $\overline{\text{EA/VPP}}$.

External Access Enable. EA harus dihubungkan pada GND agar perangkat ini dapat menerima kode dari memori program eksternal yang lokasinya mulai 0000H sampai FFFFH. EA harus dihubungkan dengan Vcc untuk eksekusi program internal. Pin ini juga menerima tegangan 12 Volt (V_{pp}) selama pemrograman Flash untuk komponen yang membutuhkan tegangan tersebut.

11. XTAL1.

Berfungsi untuk input pada inverting oscillator amplifier dan internal clock operating circuit.

12. XTAL2

Berfungsi untuk output pada inverting oscillator amplifier.

2.7. Mikrofon

Mikrofon merujuk pada suatu alat yang mengubah energi akustik (suara) menjadi energi listrik (sinyal audio). Selanjutnya mikrofon dapat dikategorikan sebagai salah satu jenis transducer (alat yang mengubah energy dari satu bentuk ke bentuk lainnya, dalam kasus ini, energy akustik menjadi energy listrik).

Mikrofon merupakan awal dari sebuah rangkaian tata suara, sehingga membuat hasil kerja mikrofon menjadi kritis untuk keseluruhan performa dari sistem tata suara. Kejernihan sebuah mikrofon menghasilkan representasi listrik dari suara yang dicupliknya. Hal ini diperoleh dengan berbagai macam metoda yang digunakan. Dahulu, sejumlah metode dipergunakan untuk berbagai keperluan, dan sekarang begitu banyak jenis mikrofon yang ditemukan dalam berbagai keperluan.

Pemilihan mikrofon yang lebih hati-hati akan mencegah terjadinya pengurangan kemampuan mikrofon dari performa maksimal. Pemilihan mikrofon yang baik tergantung dari pengetahuan akan karakteristik dasar mikrofon dan pengertian dalam mengaplikasikannya. Supaya efektif, mikrofon yang digunakan harus seimbang antara sumber suara yang ingin dicuplik (suara manusia, alat musik, dan lain-lain) dan sistem tata suara yang digunakan (sound system untuk live music, alat perekaman, dan lain-lain). Lima karakteristik dasar dari sebuah mikrofon yang harus diperhatikan ketika memilih sebuah mikrofon untuk berbagai keperluan, yaitu:

1. Prinsip dari cara kerja mikrofon.
2. Daerah respon frekuensi suara yang dapat dicuplik mikrofon.

3. Arah/sudut pencuplikan mikrofon.
4. Keluaran sinyal listrik yang dihasilkan mikrofon.
5. Bentuk fisik dari mikrofon

2.8. Motor Stepper

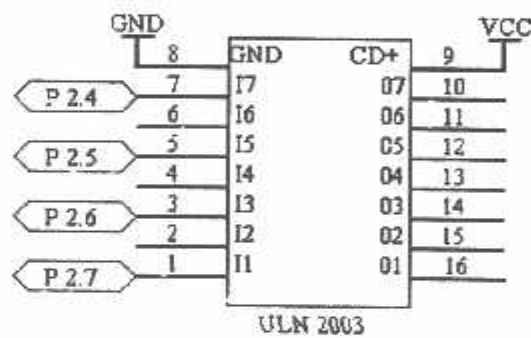
Motor stepper adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkan motor stepper diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik. Penggunaan motor stepper memiliki beberapa keunggulan dibandingkan dengan penggunaan motor DC biasa.

Keunggulannya antara lain adalah :

- Sudut rotasi motor proporsional dengan pulsa masukan sehingga lebih mudah diatur
- Motor dapat langsung memberikan torsi penuh pada saat mulai bergerak
- Posisi dan pergerakan repetisinya dapat ditentukan secara presisi
- Memiliki respon yang sangat baik terhadap mulai, stop dan berbalik (perputaran)
- Sangat realibel karena tidak adanya sikat yang bersentuhan dengan rotor seperti pada motor DC
- Dapat menghasilkan perputaran yang lambat sehingga beban dapat dikopel langsung ke porosnya

- Frekuensi perputaran dapat ditentukan secara bebas dan mudah pada *range* yang luas.

Secara teoritis, sebuah *motor stepper* dapat digerakkan langsung oleh mikrokontroler. Dalam kenyataannya, arus dan tegangan yang dikeluarkan oleh mikrokontroler terlalu kecil untuk menggerakkan sebuah *motor stepper*. Gerbang-gerbang TTL mikrokontroler hanya mampu mengeluarkan arus dalam orde mili-ampere dan tegangan antara 2 sampai 2,5 Volt. Sementara itu untuk menggerakkan *motor stepper* diperlukan arus yang lebih besar (dalam orde ampere) dan tegangan berkisar 5 sampai 24 Volt. Untuk mengatasi masalah tersebut, diutamakan suatu rangkaian *driver* pengontrol *motor stepper*.



Gambar 2.19 : Rangkaian Driver Motor Stepper [10]

2.9. Delphi

Borland Delphi merupakan paket perangkat-lunak pemrograman terpadu (*Integrated Development Environment, IDE*). Delphi dibuat oleh perusahaan perangkat-lunak bernama Borland Software Corporation. Bahasa Delphi berlandaskan pada bahasa pemrograman Pascal. Pascal sendiri diciptakan oleh seorang profesor bernama Niklaus Wirth pada tahun 1970. Bahasa Pascal

merupakan pengembangan dari bahasa pemrograman ALGOL (ALGOritmic Language), yaitu bahasa pemrograman yang dikembangkan dengan tujuan untuk mengatasi berbagai kekurangan yang terdapat pada Fortran.

Beberapa ciri khas bahasa Pascal antara lain:

- Kapitalisasi bebas. Penulisan kode program tidak perlu memperhatikan kapitalisasi huruf.
- Terstruktur. Bagian-bagian kode program dapat dikelompokkan dalam fungsi-fungsi dan prosedur-prosedur. Sedangkan fungsi dan prosedur dapat dikelompokkan dalam file terpisah yang disebut Unit.
- Memiliki tipe data standar. Bahasa Pascal memiliki beberapa tipe data standar, seperti Boolean, Char, Byte, ShortInt, Word, SmallInt, LongWord, Integer, String, dan Real.
- Tipe data dapat dikembangkan. Pemrogram dapat membuat tipe data sendiri yang diturunkan dari tipe-tipe data standar.
- Variable bersifat keras (strict). Variabel dengan tipe tertentu tidak dapat digunakan untuk menyimpan data dengan tipe lainnya.

Pada perkembangan selanjutnya, Delphi mengimplementasikan konsep Pemrograman Berorientasi Objek (*Object Oriented Programming*, OOP) melalui bahasa Pascal. Beberapa keunggulan Delphi antara lain:

- Menggunakan model Pengembangan Aplikasi Cepat (Rapid Application Development, RAD).

BAB III

PERENCANAAN DAN PEMBUATAN ALAT

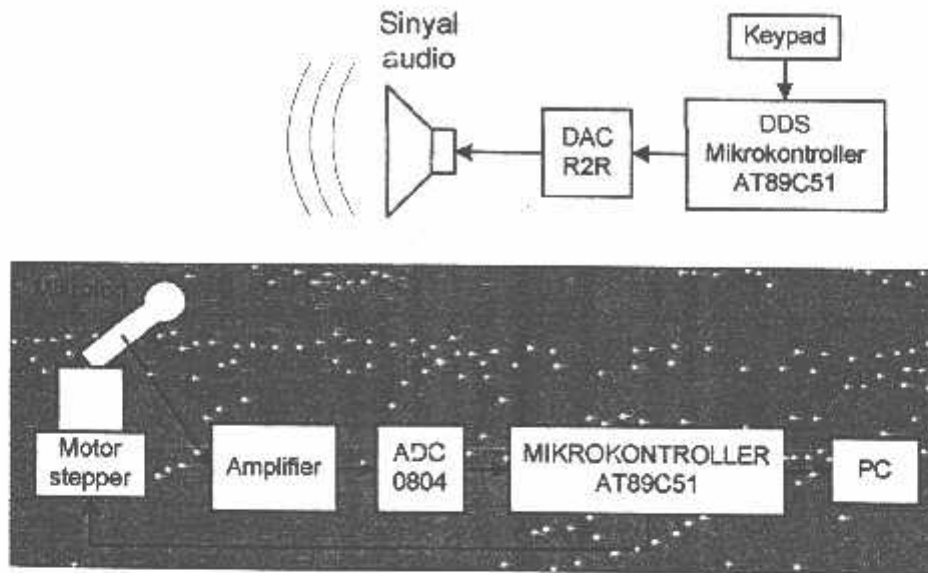
3.1. Pendahuluan

Pada Bab II telah dijelaskan berbagai teori penunjang yang digunakan dalam perancangan sistem dalam skripsi ini. Selanjutnya dilakukan perancangan secara garis besar untuk dijadikan acuan bagi tahap-tahap perencanaan selanjutnya. Adapun sistem yang direncanakan didasarkan pada beberapa spesifikasi sebagai berikut :

- a. ADC 0804 digunakan sebagai komponen utama untuk memproses data.
- b. ADC dihubungkan dengan ICL 232 yang selanjutnya dihubungkan dengan RS 232 yang ada di port serial PC.
- c. Motor stepper digunakan untuk memberikan gerak putar pada mikrofon.

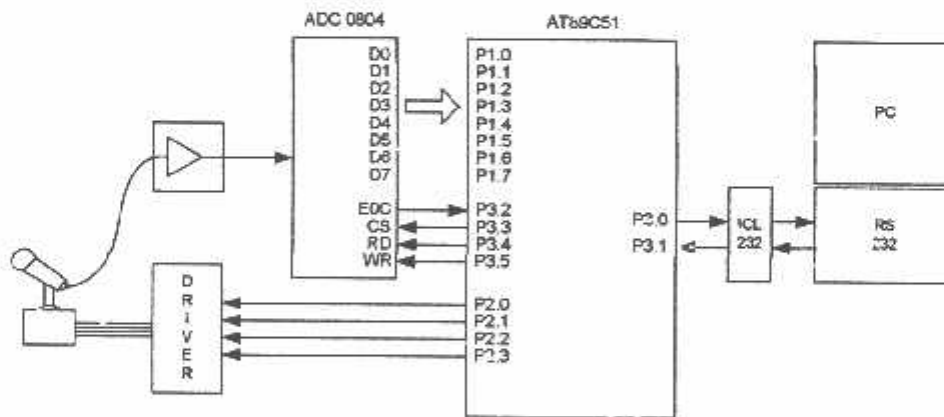
Selain itu juga digunakan komponen-komponen pendukung antara lain mikrokontroler AT89C51 dan DAC untuk komponen pembangkit sinyal audio. Tetapi komponen pelengkap ini tidak dibahas dalam perencanaan pada skripsi ini. Juga dilengkapi amplifier untuk menguatkan sinyal yang diterima oleh mikrofon.

3.2. Blok Diagram Sistem



Ruang lingkup pembahasan skripsi

Gambar 3.1: Blok diagram sistem



Gambar 3.2 : Diagram rangkaian sistem perekam

Fungsi dari masing-masing subsistem pada blok diagram di atas adalah sebagai berikut :

1. Mikrofon.

Berfungsi untuk menangkap sinyal audio. Digunakan mikrofon jenis dinamik. Mikrofon dinamik memiliki sensitivitas yang rendah tetapi harganya lebih murah sehingga banyak digunakan.

2. Amplifier.

Berfungsi sebagai penguat sinyal yang ditangkap oleh mikrofon. Amplifier yang digunakan adalah jenis operasional amplifier (op amp). Keuntungan dari penggunaan Op Amp adalah karena komponen ini memiliki penguatan (A) yang sangat besar, Impedansi input yang besar, ($Z_{in} \gg$) dan Impedansi Output yang kecil ($Z_{out} \ll$). Selain dari itu, kemampuan interval frekuensi dari komponen ini sangat lebar.

3. ADC 0804.

Berfungsi untuk merubah sinyal analog dari amplifier menjadi sinyal digital untuk diteruskan ke PC.

4. Motor stepper.

Untuk memutar mikrofon dari posisi awal (0°) sampai 360° .

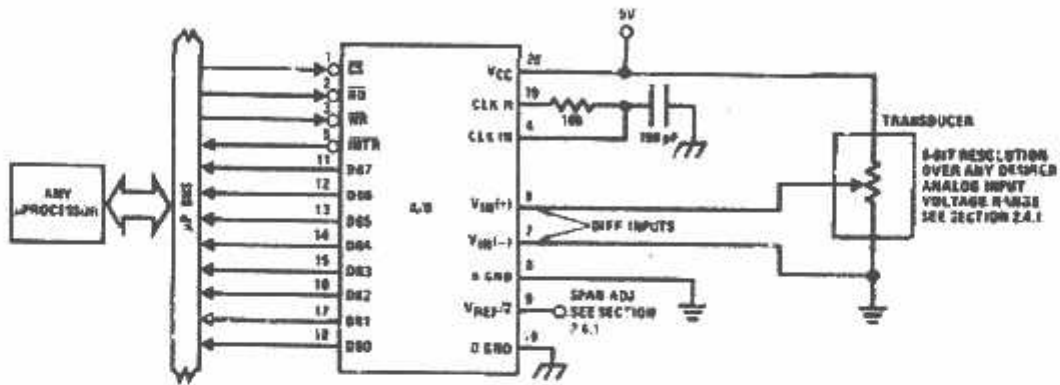
5. PC.

Berfungsi untuk merekam sinyal audio yang diterima. PC dilengkapi dengan software yang dirancang untuk melakukan proses perekaman.

3.3. Perencanaan Perangkat Keras

3.3.1. Sistem Minimum ADC 0804

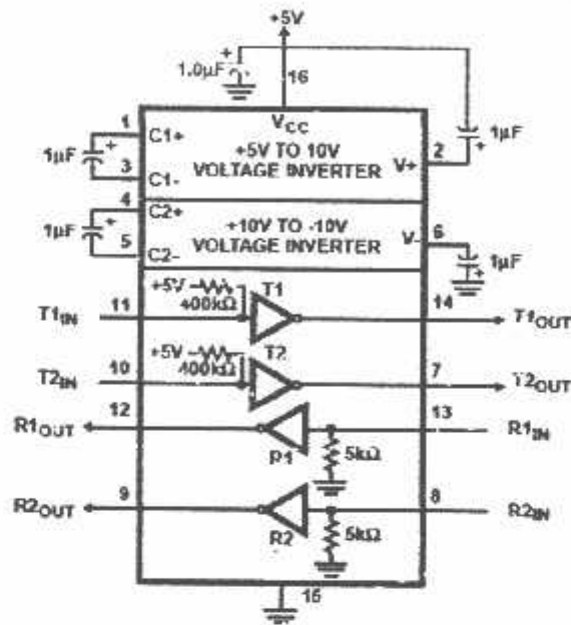
Komponen ADC 0804 ini berfungsi untuk merubah sinyal analog menjadi sinyal digital agar dapat diolah oleh PC. Sebagai unit konversi sinyal maka pin-pin ADC 0804 dihubungkan dengan PC dengan skema sebagai berikut :



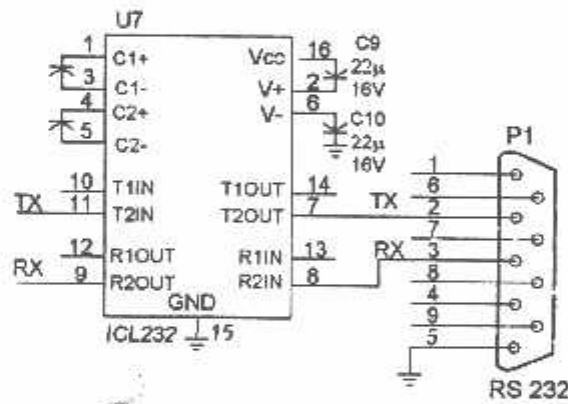
Gambar 3.3 : Sistem minimum ADC 0804 [10]

ADC 0804 dapat dihubungkan dengan berbagai macam mikrokontroler. Yang dibutuhkan adalah 11 pin yaitu 8 pin untuk data dan 3 pin untuk kontrol.

3.3.2. Koneksi ICL 232 dan RS 232



Gambar 3.4 : Diagram fungsi ICL 232 [12]

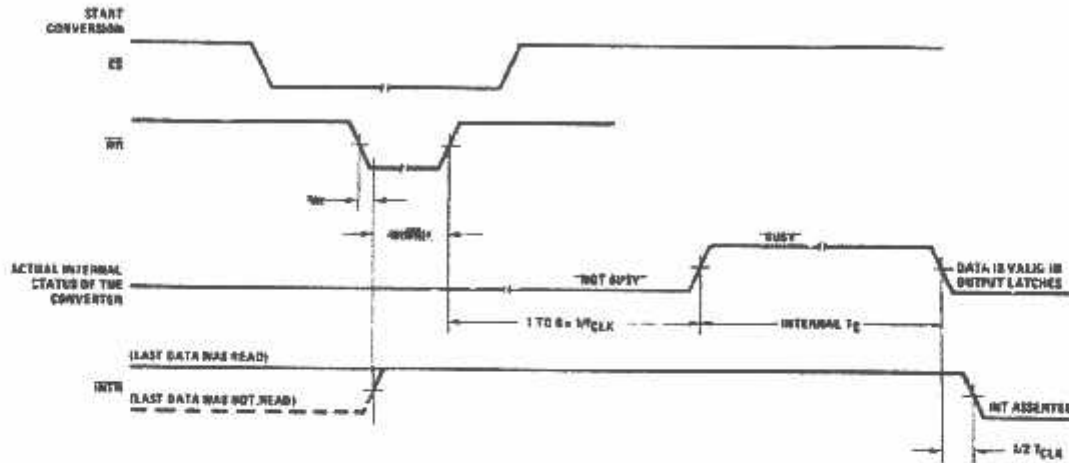


Gambar 3.5 : Koneksi ICL 232 dan RS 232 [11]

Koneksi dilakukan dengan menghubungkan pin-pin ADC 0804 dengan pin ICL 232. Pin-pin pada ICL.232 selanjutnya dihubungkan dengan pin-pin RS 232 di port serial komputer. Tegangan input yang digunakan di ADC 0820 sebesar +5 V.

3.3.3. Perencanaan Timing Diagram

Perencanaan timing diagram ADC 0804 dilakukan sesuai dengan referensi data sheet ADC 0804.

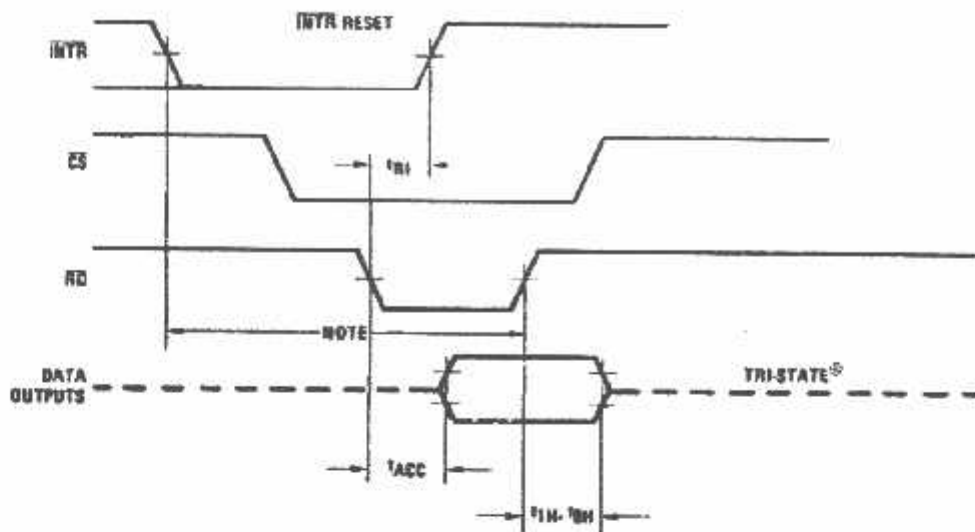


Gambar 3.6 : Timing diagram [10]

Timing diagram di atas menunjukkan bagaimana cara memulai konversi data.

Konversi data dilakukan dengan tahap-tahap sebagai berikut :

1. Chip select (CS) diatur pada posisi sinyal low.
2. Selanjutnya WR pada posisi low.
3. Chip select (CS) pada posisi high.
4. Konversi berakhir setelah pin INTR berubah menjadi low.



Gambar 3.7 : Output enable dan Reset INTR [10]

Setelah konversi telah selesai dilakukan oleh ADC, data telah tersimpan dalam pintu output dari ADC. Diagram di atas menunjukkan bagaimana membaca nilai konversi dari pintu output dari ADC. Data dari konversi yang baru dilakukan hanya tersedia untuk dibaca setelah ADC 0804 memposisikan pin INTR pada kondisi low atau ketika konversi telah selesai dilakukan. Langkah-langkah pembacaan output dari ADC 0804 adalah sebagai berikut :

1. Chips select (CS) diposisikan low.
2. Pin pembaca (RD) diposisikan low.
3. Data dibaca dari port dimana ADC tersebut dihubungkan.
4. Pin RD diposisikan high.
5. Chip select (CS) diposisikan high.

3.4. Perencanaan Motor Stepper

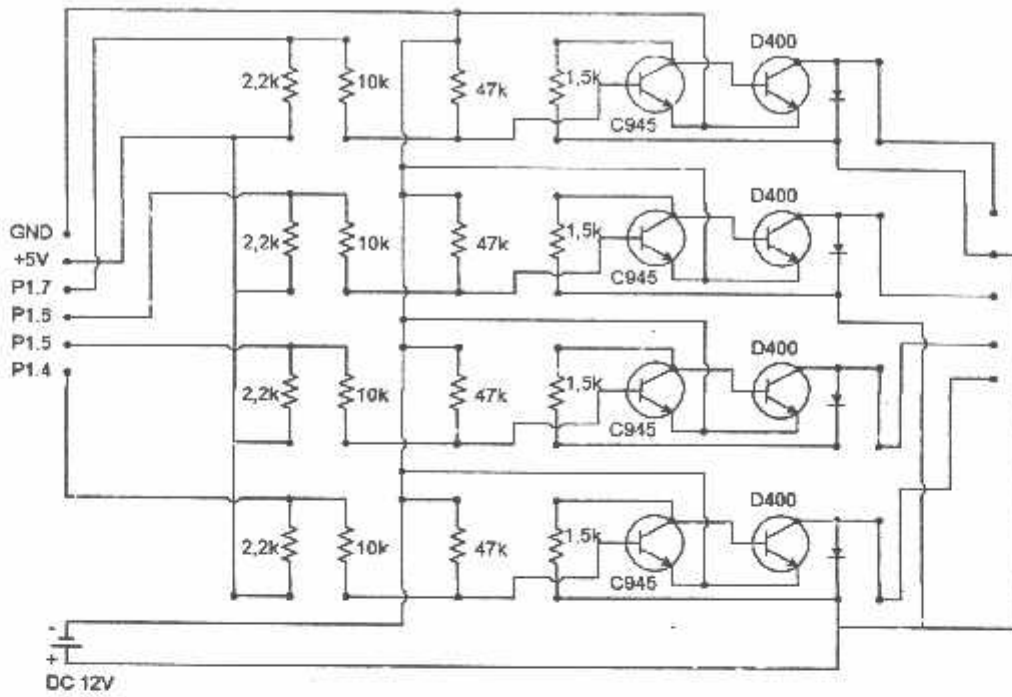


Gambar 3.8 : Skema motor stepper [12]

Gambar di atas menampilkan penampang motor stepper dengan empat koil. Setiap koil memiliki empat kondisi kutub. Bila kondisi satu yang aktif, posisi rotor akan tampak seperti gambar di atas. Bila kondisi bergeser ke kondisi dua, rotor akan berputar ke kiri dengan sudut putar sesuai jarak kondisi satu ke kondisi dua. Tetapi jika setelah kondisi satu, kondisi empat yang aktif, rotor akan menuju ke koil dengan pin empat paling dekat dengan pin satu kondisi sebelumnya. Hal ini menyebabkan rotor berputar ke kanan dan seterusnya.

Ketelitian sudut putar pada motor stepper sebanding dengan banyaknya koil dan kondisi masukannya. Pada kondisi di atas, motor stepper dengan empat koil dan empat kondisi kutub dengan metode full step akan mampu menghadap atau bergerak ke 16 sudut yang berbeda.

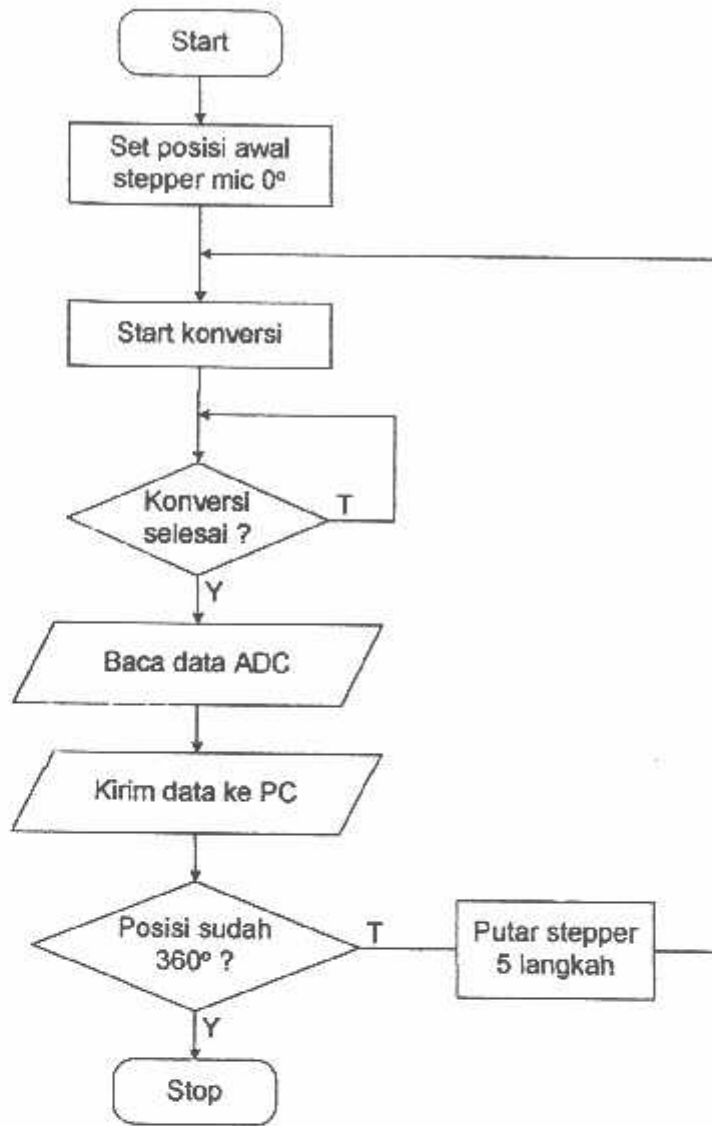
Secara teoritis, sebuah *motor stepper* dapat digerakkan langsung oleh mikrokontroller. Dalam kenyataannya, arus dan tegangan yang dikeluarkan oleh mikrokontroller terlalu kecil untuk menggerakkan sebuah *motor stepper*. Gerbang-gerbang TTL mikrokontroller hanya mampu mengeluarkan arus dalam orde mili-ampere dan tegangan antara 2 sampai 2,5 Volt. Sementara itu untuk menggerakkan *motor stepper* diperlukan arus yang lebih besar (dalam orde ampere) dan tegangan berkisar 5 sampai 24 Volt. Untuk mengatasi masalah tersebut, dibutuhkan suatu rangkaian *driver* pengontrol *motor stepper*.



Gambar 3.10 : Rangkaian driver motor stepper

3.5. Perencanaan Perangkat Lunak

Perangkat lunak ini berdasarkan pengendali utama yaitu ADC 0804. Pembuatan perangkat lunak sistem aplikasi berdasarkan pada semua proses yang harus dikerjakan perangkat keras. Perangkat lunak dibutuhkan oleh PC untuk mengolah sinyal dari ADC 0804. Bahasa pemrograman yang digunakan adalah bahasa Delphi. Pembuatan perangkat lunak harus melalui proses-proses uji coba secara *software* maupun secara *hardware*.



Gambar 3.11 : Flow chart untuk mikrokontroller



Gambar 3.12 : Flow chart untuk PC

BAB IV

PENGUJIAN SISTEM

4.1. Pendahuluan

Bab ini membahas pengujian sistem. Sistem yang telah dirancang dan dibuat harus diuji kinerjanya. Dalam hal ini, sistem diberi input suara dengan frekuensi tertentu. Kinerja sistem yang diamati adalah kinerja motor stepper dan penangkapan sinyal oleh mikrofon. Kinerja motor stepper dalam hal ini adalah bagaimana gerakan putarnya. Kinerja mikrofon dalam hal ini adalah sensitivitas mikrofon tersebut.

Pengujian sistem ini meliputi pengujian bagian-bagian dari sistem perekam tanggapan frekuensi meliputi :

1. Pengujian rangkaian ADC 0804.
2. Pengujian motor stepper.
3. Pengujian mikrofon

4.2. Pengujian rangkaian ADC 0804

- Tujuan :

Tujuan pengujian ini adalah untuk mengetahui level tegangan keluaran ADC dan kelinieran hasil konversi ADC

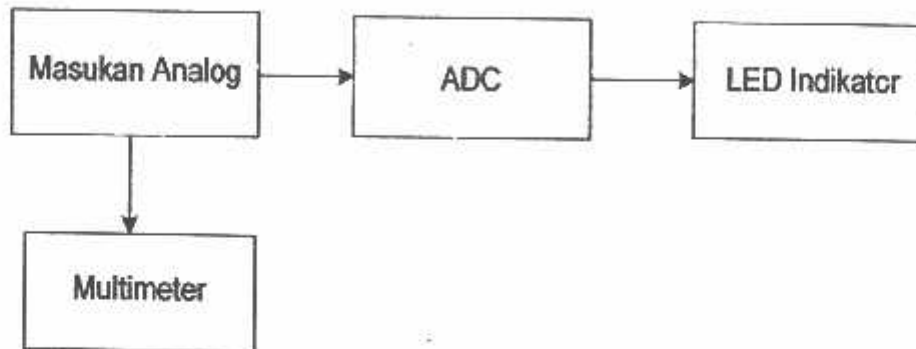
- Alat dan Bahan :

- Catu daya 5 volt

- Masukan analog

- Multimeter

- Blok diagram pengujian :



Gambar 4.1 : Pengujian Sub Sistem ADC

- Prosedur pengujian :

1. Menyusun rangkaian pengujian seperti yang ditunjukkan dalam gambar 4.1 di atas.
2. ADC diberikan masukan tegangan secara bertahap 0 – 5 volt dengan step 0.5 volt
3. Keluaran ADC dihubungkan dengan *Logic Probe* untuk mengetahui konversi masukan analog dengan keluaran digital berupa data biner.
4. Mengamati nyala *logic probe* untuk tiap-tiap masukan analog yang berbeda.

- Hasil pengujian dan analisa :

Hasil pengujian rangkaian ADC ditunjukkan dalam tabel berikut ini:

Tabel 4.1: Hasil Pengujian ADC

Vin	D7	D6	D5	D4	D3	D2	D1	D0	Hex	Des
0	0	0	0	0	0	0	0	0	00H	0
0.5	0	0	0	1	1	0	0	1	19H	25
1	0	0	1	1	0	0	1	1	33H	51
1.5	0	1	0	0	1	1	0	1	4DH	77
2	0	1	1	0	0	1	1	1	67H	103
2.5	1	0	0	0	0	0	0	1	81H	129
3	1	0	0	1	1	0	1	1	9BH	155
3.5	1	0	1	1	0	1	1	0	B6H	182
4	1	1	0	1	0	0	0	1	D1H	208
4.5	1	1	1	0	1	0	0	1	E9H	233
5	1	1	1	1	1	1	1	1	FFH	255

Untuk tegangan referensi (+) sebesar 5 volt dan referensi (-) sebesar 0 volt, maka diperoleh :

$$\begin{aligned}
 \text{Resolusi} &= \frac{V_{ref}}{2^n} \\
 &= \frac{5}{256} \\
 &= 0.0195 \text{ Volt}
 \end{aligned}$$

Apabila diperoleh keluaran 19H atau 25 maka sebenarnya tegangan masukan yang diberikan :

$$\begin{aligned}
 V_{in} &= \text{Resolusi} \times \text{nilai decimal} \\
 &= 0.0195 \times 25 \\
 &= 0.5 \text{ V}
 \end{aligned}$$

Dengan demikian ADC dapat bekerja dengan baik.

Dari hasil pengujian dan perhitungan, maka akan diketahui seberapa besar % kesalahan yang dapat diketahui :

$$\% \text{kesalahan} = \frac{(\text{perhitungan} - \text{pengukuran})}{\text{perhitungan}} \times 100\%$$

$V_{in} = 0,5$ Volt maka :

$$\begin{aligned}
 \% \text{ kesalahan} &= \frac{(26,31 - 25)}{26,31} \times 100\% \\
 &= 1,16\%
 \end{aligned}$$

$V_m = 1,5$ Volt maka :

$$\begin{aligned}
 \% \text{ kesalahan} &= \frac{(0,721 - 103)}{0,721} \times 100\% \\
 &= 1,4\%
 \end{aligned}$$

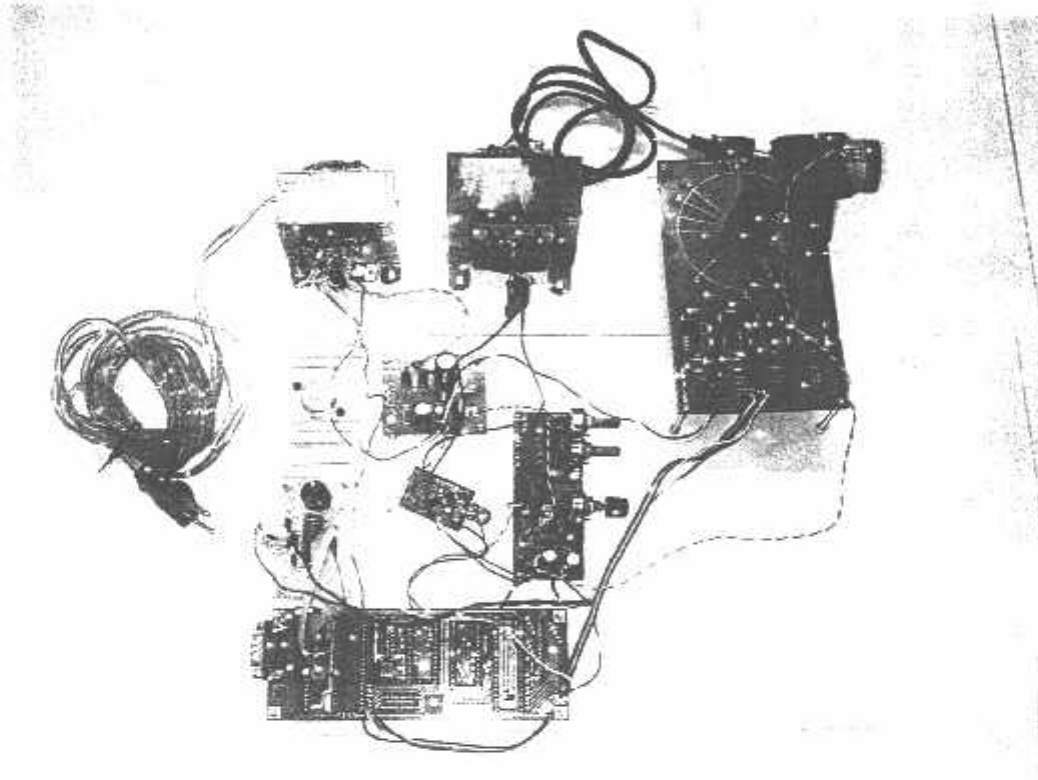
$V_{in} = 4$ Volt maka :

$$\begin{aligned}
 \% \text{ kesalahan} &= \frac{(32,5 - 208)}{32,5} \times 100\% \\
 &= 0,05\%
 \end{aligned}$$

Dari hasil yang diperoleh maka besar persentase rata-rata adalah 1% sampai dengan 2%. Dengan itu dapat kita ketahui bahwa persentase kesalahan sangat kecil sekali.

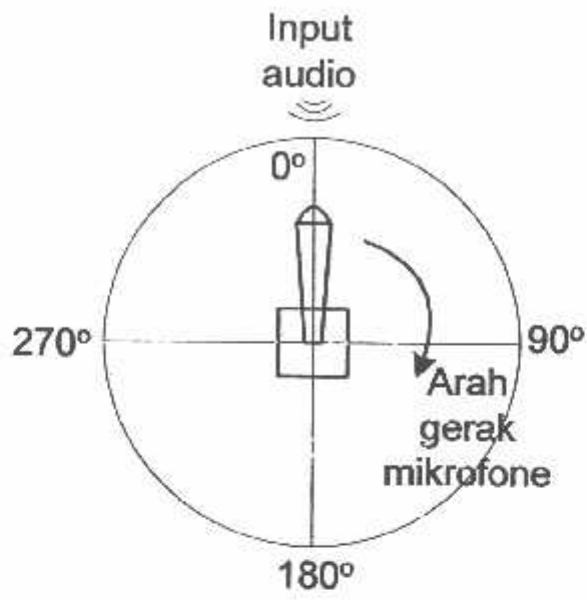
4.3. Pengujian Sistem

Sistem yang telah dirancang, dibuat dalam sebuah rangkaian alat, sebagai berikut :



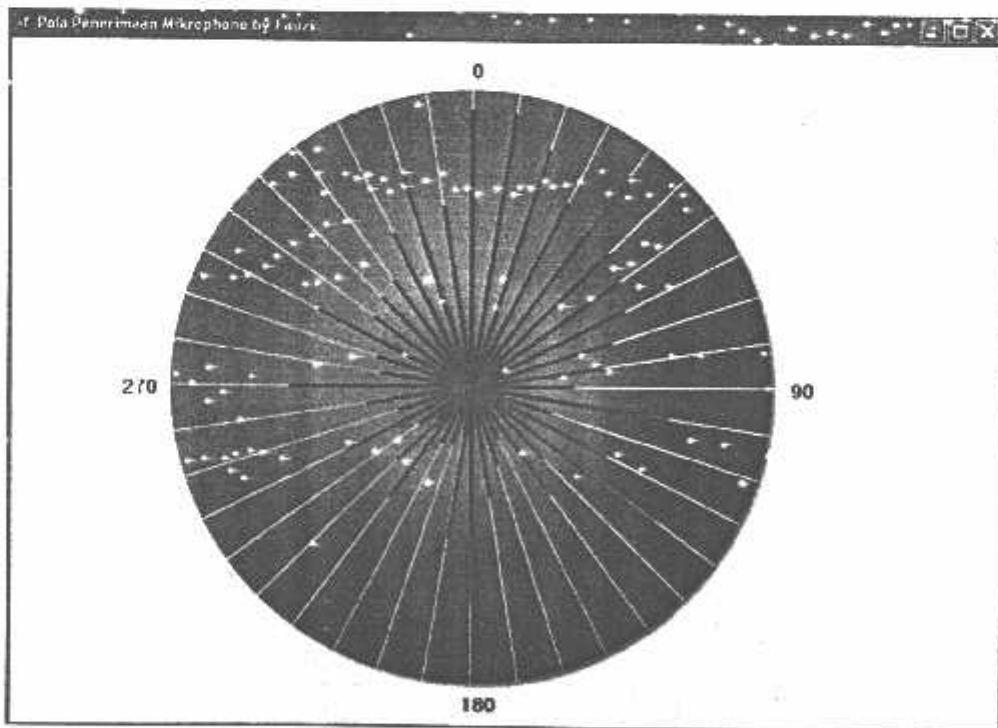
Gambar 4.2 : Foto alat

Pengujian dilakukan dengan memberikan input audio. Mikrofon diposisikan searah dengan arah sumber sinyal audio. Posisi awal mikrofon ini adalah posisi 0° . Setelah itu, motor stepper digerakkan searah dengan jarum jam. Setiap langkah motor stepper ini, mikrofon menangkap sinyal audio dan merekamnya di PC.



Gambar 4.3 : Skema pengujian

Hasil pengujiannya adalah :



Gambar 4.4 : Hasil pengujian

Hasil pengujian menunjukkan bahwa mikrofone dapat menangkap input audio. Tetapi hasil pengujian terhadap mikrofon di atas juga menunjukkan bahwa sistem belum bekerja dengan sempurna. Pola grafik dari penerimaan mikrofon masih menunjukkan pola yang tidak beraturan. Untuk itu diperlukan komponen pengolah input audio (filter) agar hasil yang didapat menjadi lebih baik.

Dengan sistem kendali mikrokontroller AT89C51, motor stepper dapat bergerak searah jarum jam sebesar $1,8^\circ$ per langkah (step). Motor stepper ini diprogram untuk bergerak 5 step setiap waktu tertentu. Jadi, untuk setiap waktu gerakan, motor bergerak sebesar 9° . Gerakan ini dapat dikendalikan sampai satu putaran penuh (360°). Pada posisi satu putaran penuh ini, motor stepper kembali berada pada posisi awal (0°). Dari ini maka dapat dikatakan bahwa tingkat kesalahan gerakan motor stepper adalah 0%.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil perencanaan dan pengujian dapat disimpulkan sebagai berikut :

1. Hasil pengujian ADC memiliki margin error 0,5 - 1%, sehingga ADC yang digunakan memiliki tingkat error yang kecil.
2. Motor stepper dapat bergerak (berputar) sebesar 9° per step, hingga satu putaran (360°) dan kembali pada posisi awal, sehingga tingkat error yang didapatkan adalah 0%.
3. Pola grafik dari penerimaan mikrofon tidak beraturan sehingga sistem masih belum dapat berfungsi dengan baik.
4. Sistem dapat diaplikasikan terutama untuk pembuatan sistem perekam tanggapan frekuensi yang murah dan sederhana.

5.2. Saran

Pengujian sebaiknya dilakukan di ruang kedap suara agar tidak ada gangguan suara dari sumber lain di sekitarnya.

DAFTAR PUSTAKA

- [1]. Boomkamp J, 2007, **Basic Signal Theory**, Sound & Music Software Development, Utrecht School of the Arts
 - [2]. <http://www.wikipedia.com>, 2008, **Porland Delphi**
 - [3]. <http://www.wikipedia.com>, 2008, **Wave**
 - [4]. Intersil Corporation, 1999, **ICL 232**
 - [5]. Juhana J, 2008, **Sinyal dan Sistem Digital**, Teknik Telekomunikasi, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung
 - [6]. National Semiconductor Corporation, 2008
 - [7]. www.logix4u.net, 2008, **A Tutorial On Parallel Port Interfacing**
 - [8]. Riyanto Z, 2004, **Komunikasi Data**, FMIPA Universitas Gadjah Mada Yogyakarta
 - [9]. Philips Semiconductor Corporation, 1994, **ADC0801/ADC0802/ADC0803/ADC0804/ADC0805 8-Bit Mp Compatible A/D Converters**
 - [10]. Wisnu H, 2006, **Pengendalian Motor Stepper Metode Half Step Dengan Rangkaian Digital**, Universitas Diponegoro, Semarang
 - [11]. http://www.camresearch.com/Data_Com_Basics/RS232_standard.html
 - [12]. Futurlec, 2009, **AT89C51 - 8 bit Microcontroller with 4k Bytes Flash**
-

LAMPIRAN

Lampiran Program

```
START BIT      P3.5
READ          BIT      P3.4
CS           BIT      P3.3
EOC          BIT      P3.2

;INISIALISASI SERIAL
      MOV      TMOD, #20H
      MOV      TH1, #0FDH
      SETB     TR1
      MOV      SCON, #50H
      MOV      R6, #01H

MULAI:      MOV      R3, #40

ULANG:      CALL     ADC
            MOV      SBUF, R7
            JNB     T1, $
            CLR     T1
            CALL     TUNDA
            CALL     TUNDA
            MOV      R2, #5
            CALL     PUTAR_KANAN
            CALL     TUNDA
            CALL     TUNDA
            DJNZ    R3, ULANG
            MOV      R4, #50

LAGI:       CALL     BALIK
            DJNZ    R4, LAGI
            JMP     $

PUTAR_KANAN:
      MOV      A, R6
      RL
      CJN     A, #10H, LANJUT1
      MOV      R6, #01H
      MOV      A, #01H

LANJUT1:    MOV      P2, A
            MOV      R6, A
            CALL     TUNDA2
            DJNZ    R2, PUTAR_KANAN
            RET
```

BALIK:	MOV	P2, #08H
	CALL	TUNDA2
	MOV	P2, #04H
	CALL	TUNDA2
	MOV	P2, #04H
	CALL	TUNDA2
	MOV	P2, #02H
	CALL	TUNDA2
	MOV	P2, #01H
	CALL	TUNDA2
	RET	
ADC:	MOV	P3, #OFFH
	CLR	CS
	CLR	START
	NOF	
	SETB	START
	JB	EOC, \$
	CLR	READ
	NOF	
	MOV	R7, P1
	SETB	READ
	SETB	CS
	RET	
TUNDA2:	MOV	R0, #100
TUNDA3:	MOV	R1, #100
	DJNZ	R1, \$
	DJNZ	R0, TUNDA3
	RET	
TUNDA:	MOV	R0, #255
TUNDA1:	MOV	R1, #255
	DJNZ	R1, \$
	DJNZ	R0, TUNDA1
	RET	
	END	



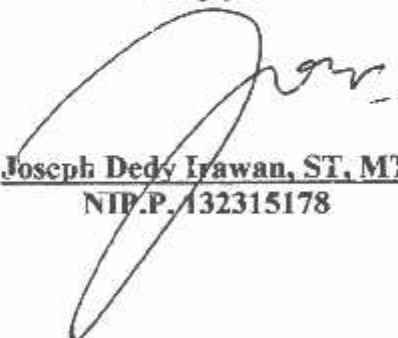
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Raya Karanglo Km.2
Malang

FOMULIR PERBAIKAN SKRIPSI

Nama : Fauzi. Chairul. Alam
NIM : 98.17.047
Masa Bimbingan : 20 Desember 2008 -- 20 Juni 2009
Judul Skripsi : PERANCANGAN DAN PEMBUATAN SISTEM PEREKAM
TANGGAPAN FREKUENSI UNTUK MIKROFON AUDIO
BERBASIS PC

NO.	TANGGAL	URAIAN	PARAF
1	23 MARET 2009	PERANCANGAN MOTOR STEPPER	
2	23 MARET 2009	HASIL SESUNGGUHNYA	
3	23 MARET 2009	KESIMPULAN DAN SARAN	

Disetujui,
Penguji I


Joseph Dedy Irawan, ST, MT
NIP.P. 132315178

Mengetahui,

Dosen Pembimbing


Ir. F. Yudi Lampraptono, MT
NIP.Y. 1039500274



ITN
JL. Raya Karanglo, Km 2
MALANG

Formulir Bimbingan Skripsi

Nama : FAUZI CHAIRUL ALAM
Nim : 98.17.047
Masa Bimbingan : 20 Desember 2008 – 20 Juni 2009
Judul : PERANCANGAN DAN PEMBUATAN SISTEM PEREKAM
TANGGAPAN FREKUENSI UNTUK MIKROFON AUDIO
BERBASIS PC

No	Tanggal	Uraian	Paraf
1		Bab I	
2		Bab II	
3		Bab III	
4		Bab IV	
5		Selesaian hasil	
6		Bab V	

Malang, 2009
Dosen Pembimbing

Ir.F.Yudi Limpraptono, MT
NIP Y. 1039500274

FORM S-4B

D A T A
S H E E T

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

DESCRIPTION

By using a half-flash conversion technique, the 8-bit ADC0820 CMOS A/D offers a 1.5 μ s conversion time while dissipating a maximum 75mW of power. The half-flash technique consists of 31 comparators, a most significant 4-bit ADC and a least significant 4-bit ADC.

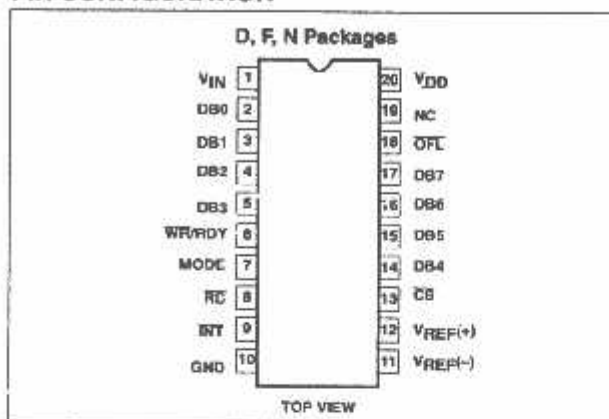
The input to the ADC0820 is tracked and held by the input sampling circuitry, eliminating the need for an external sample-and-hold for signals slewing at less than 100mV/ μ s.

For ease of interface to microprocessors, the ADC0820 has been designed to appear as a memory location or I/O port without the need for external interfacing logic.

FEATURES

- Built-in track-and-hold function
- No missing codes
- No external clocking
- Single supply—5V_{DC}
- Easy interface to all microprocessors, or operates stand-alone
- Latched 3-State outputs
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Operates ratiometrically or with any reference value equal to or less than V_{DD}
- 0V to 5V analog input voltage range with single 5V supply
- No zero- or full-scale adjust required
- Overflow output available for cascading
- 0.3" standard width 20-pin DIP

PIN CONFIGURATION



APPLICATIONS

- Microprocessor-based monitoring and control systems
- Transducer/ μ P interface
- Process control
- Logic analyzers
- Test and measurement

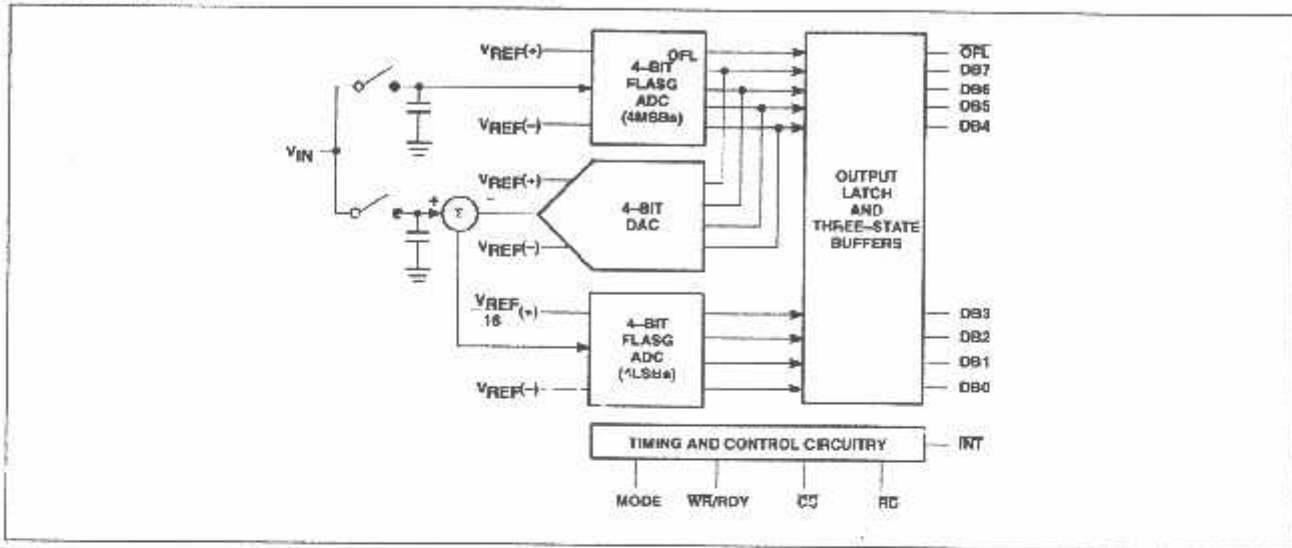
ORDERING INFORMATION

DESCRIPTION	TEMPERATURE RANGE	ORDER CODE	DWG #
20-Pin Plastic Dual In-Line Package (DIP)	0 to +70°C	ADC0820CNEN	D408B
20-Pin Plastic Small Outline (SO) package	0 to +70°C	ADC0820CNED	1021B

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

BLOCK DIAGRAM



PIN DESCRIPTION

PIN NO	SYMBOL	DESCRIPTION
1	V_{IN}	Analog input; range= $GND \leq V_{IN} \leq V_{DD}$
2	DB0	3-state data output—Bit 0 (LSB)
3	DB1	3-state data output—Bit 1
4	DB2	3-state data output—Bit 2
5	DB3	3-state data output—Bit 3
6	WR/RDY	WR-RD Mode WR: With \overline{CS} Low, the conversion is started on the falling edge of WR. Approximately 800ns (the preset internal time out, t_i) after the WR rising edge, the result of the conversion will be strobed into the output latch, provided that \overline{RD} does not occur prior to this time out (see Figures 3a and 3b). RD Mode RDY: This is an open-drain output (no internal pull-up device). RDY will go Low after the falling edge of \overline{CS} ; RDY will go 3-State when the result of the conversion is strobed into the output latch. It is used to simplify the interface to a microprocessor system (see Figure 1).
7	Mode	Mode: Mode selection input—it is internally tied to GND through a 30 μ A current source. RD Mode: When mode is Low. WR-RD Mode: When mode is High.
8	\overline{RD}	WR-RD Mode With \overline{CS} Low, the 3-State data outputs (DB0-DB7) will be activated when \overline{RD} goes Low. \overline{RD} can also be used to increase the speed of the converter by reading data prior to the preset internal time out ($T_i \sim 800$ ns). If this is done, the data result transferred to output latch is latched after the falling edge of the \overline{RD} (see Figures 3a and 3b). RD Mode With \overline{CS} Low, the conversion will start with \overline{RD} going Low; also, \overline{RD} will enable the 3-State data outputs at the completion of the conversion. RDY going 3-State and INT going Low indicate the completion of the conversion (see Figure 1).
9	INT	WR-RD Mode INT going Low indicates that the conversion is completed and the data result is in the output latch. INT will go Low ~ 800 ns (the preset internal time out, t_i) after the rising edge of WR (see Figure 3a); or INT will go Low after the falling edge of \overline{RD} , if \overline{RD} goes Low prior to the 800ns time out (see Figure 3b). INT is reset by the rising edge of RD or CS (see Figures 3a and 3b).

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

PIN DESCRIPTION (Continued)

PIN NO	SYMBOL	DESCRIPTION
		RD Mode
		INT going Low indicates that the conversion is completed and the data result is in the output latch. INT is reset by the rising edge of RD or CS (see Figure 1).
10	GND	Ground
11	V _{REF(-)}	The bottom of resistor ladder, voltage range: GND \leq V _{REF(-)} \leq V _{REF(+)}
12	V _{REF(+)}	The top of resistor ladder, voltage range: V _{REF(-)} \leq V _{REF(+)} \leq V _{DD}
13	CS	CS must be Low in order for the RD or WR to be recognized by the converter.
14	DB4	3-State data output—Bit 4
15	DB5	3-State data output—Bit 5
16	DB6	3-State data output—Bit 6
17	DB7	3-State data output—Bit 7 (MSB)
18	OFL	Overflow output—If the analog input is higher than the V _{REF(+)} LSB, OFL will be low at the end of conversion. It can be used to cascade 2 or more devices to have more resolution (9, 10-bit). It is always active and never becomes 3-state.
19	NC	No connection
20	V _{DD}	Power supply voltage

ABSOLUTE MAXIMUM RATINGS^{1, 2}

SYMBOL	PARAMETER	RATING	UNIT
V _{DD}	Supply voltage	7	V
	Logic control inputs	-0.2 to V _{DD} +0.2	V
	Voltage at other inputs and output	-0.2 to V _{DD} +0.2	V
T _{STG}	Storage temperature range	-65 to +150	°C
P _D	Maximum power dissipation ³ T _A =25°C(still-air) N package D package	1890 1390	mW mW
T _{SOLD}	Lead temperature (soldering, 10sec)	300	°C
T _A	Operating ambient temperature range ADC0820CNEN/CNED	T _{MIN} \leq T _A \leq T _{MAX} 0 to +70	°C

NOTES:

1. Absolute Maximum Ratings are those values beyond which the life of the device may be impaired.
2. All voltages are measured with respect to GND, unless otherwise specified.
3. Derate above 25°C at the following rates:
N package at 13.5mW/°C
D package at 11.1mW/°C

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

DC ELECTRICAL CHARACTERISTICS

RD mode (Pin 7=0), $V_{DD}=5V$, $V_{REF(+)}=5V$, and $V_{REF(-)}=GND$, unless otherwise specified. Limits apply from T_{MIN} to T_{MAX} .

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT	
			Min	Typ ³	Max		
	Resolution		8	8	8	bits	
	Unadjusted error ¹	ADC0820C			± 1	LSB	
R_{REF}	Reference resistance		1	1.6	4	k Ω	
$V_{REF(+)}$	Input voltage ⁵		$V_{REF(-)}$		V_{DD}	V	
$V_{REF(-)}$	Input voltage		GND		$V_{REF(+)}$	V	
V_{IN}	Input voltage ⁵		GND-0.1		$V_{DD}+0.1$	V	
	Maximum analog input leakage current	$C_S=V_{DD}$ $V_{IN}=V_{DD}$ $V_{IN}=GND$	-3		3	μ A	
	Power supply sensitivity	$V_{DD}=5V \pm 5\%$		$\pm 1/16$	$\pm 1/4$	LSB	
$V_{IN(1)}$	Logical "1" input voltage	$V_{DD}=5.25V$	CS, WR, RD Mode	2.0 3.5	V_{DD} V_{DD}	V	
$V_{IN(0)}$	Logical "0" input voltage	$V_{DD}=4.75V$	CS, WR, RD Mode	GND GND	0.8 1.5	V	
$I_{IN(1)}$	Logical "1" input current	$V_{IN(1)}=5V$; CS, RD $V_{IN(1)}=5V$; WR $V_{IN(1)}=5V$; Mode			1 3 200	μ A	
$I_{IN(0)}$	Logical "0" input current	$V_{IN(0)}=0V$; CS, RD, WR, Mode		-1		μ A	
$V_{OUT(1)}$	Logical "1" output voltage	$V_{DD}=4.75V$, $I_{OUT}=-380\mu A$; DB0-DB7, OFL, INT		2.4	4.6	V	
		$V_{DD}=4.75V$, $I_{OUT}=-10\mu A$; DB0-DB7, OFL, INT		4.5	4.74		
$V_{OUT(0)}$	Logical "0" output voltage	$V_{DD}=4.75V$, $I_{OUT}=1.6mA$; DB0-DB7, OFL, INT, RDY			0.2	0.4	V
I_{OZ}	3-state output current	$V_{OUT}=5V$; DB0-DB7, RDY				3	μ A
		$V_{OUT}=0V$; DB0-DB7, RDY		-3			μ A
I_{SOURCE}	Output source current	$V_{OUT}=0V$; DB0-DB7, OFL		6	12		μ A
		INT		4.5	8		μ A
I_{SINK}	Output sink current	$V_{OUT}=5V$; DB0-DB7, OFL, INT, RDY		7	20		μ A
I_{DD}	Supply current	$C_S=WR=RD=0$			6	15	μ A
V_{DD}	Range			4.5		5.5	V

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

AC ELECTRICAL CHARACTERISTICS

 $V_{DD} = 5V$, $t_R = t_F = 20ns$, $V_{REF(+)} = 5V$, $V_{REF(-)} = 0V$, and $T_A = 25^\circ C$, unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS ⁴			UNIT
			Min	Typ ³	Max	
t_{CRD}	Conversion time for RD mode	Mode=0, Figure 1		1.6	2.5	μs
t_{ACCO}	Access time (delay from falling edge of RD to output valid)	Mode=0, Figure 1		$t_{CRD}+20$	$t_{CRD}+50$	ns
t_{CWR-RD}	Conversion time for WR-RD mode	Mode= V_{DD} , $t_{WR}=600ns$, $t_{RD}=600ns$; Figures 3a and 3b			1.52	μs
t_{WR}	Writes time	Min	Mode= V_{DD} , Figures 3a and 3b ²	600		ns
		Max				
t_{RD}	Read time	Min	Mode= V_{DD} , Figures 3a and 3b ²	600		ns
t_{ACC1}	Access time (delay from falling edge of RD to output valid)	Mode= V_{DD} , $t_{RD} < t_i$; Figure 3b, $C_L = 15pF$		190	280	ns
		$C_L = 100pF$		210	320	
t_{ACC2}	Access time (delay from falling edge of RD to output valid)	Mode= V_{DD} , $t_{RD} > t_i$; Figure 3a, $C_L = 15pF$		70	120	ns
		$C_L = 100pF$		90	150	
t_i	Internal comparison time	Mode= V_{DD} ; Figures 2 and 3a, $C_L = 50pF$		800	1300	ns
t_{1H} , t_{0H}	Three-state control (delay from rising edge of RD to Hi-Z state)	$R_L = 1k\Omega$, $C_L = 10pF$		100	200	ns
t_{INTL}	Delay from rising edge of WR to falling edge of INT	Mode= V_{DD} , $C_L = 50pF$ $t_{RD} > t_i$; Figure 3a $t_{RD} < t_i$; Figure 3b		$t_{RD}+200$	t_i $t_{RD}+290$	ns ns
t_{INTH}	Delay from rising edge of RD to rising edge of INT	Figures 1, 3a, and 3b, $C_L = 50pF$		125	225	ns
t_{INTHWR}	Delay from rising edge of WR to rising edge of INT	Figure 2, $C_L = 50pF$		175	270	ns
t_{RDY}	Delay from CS to RDY	Figure 1, $C_L = 50pF$, Mode=0		50	100	ns
t_{ID}	Delay from INT to output valid	Figure 2		20	50	ns
t_{RI}	Delay from RD to INT	Mode= V_{DD} , $t_{RD} < t_i$; Figure 3b		200	290	ns
t_P	Delay from end of conversion to next conversion	Figures 1, 2, 3a, and 3b ²		500		ns
SR	Slew rate, tracking			0.1		V/ μs
C_{VIN}	Analog input capacitance			45		pF
C_{OUT}	Logic output capacitance			5		pF
C_{IN}	Logic input capacitance			5		pF

NOTES:

- Unadjusted error includes offset, full-scale, and linearity errors.
- Accuracy may degrade if t_{WR} or t_{RD} is shorter than the minimum value specified.
- Typical values are at $25^\circ C$ and represent most likely parametric norm.
- Guaranteed but not 100% production tested. These limits are not used to calculate outgoing quality levels.
- V_{REF} and V_{IN} must be applied after V_{CC} has been turned on to prevent possibility of latching.

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

3-STATE TEST CIRCUITS AND WAVEFORMS

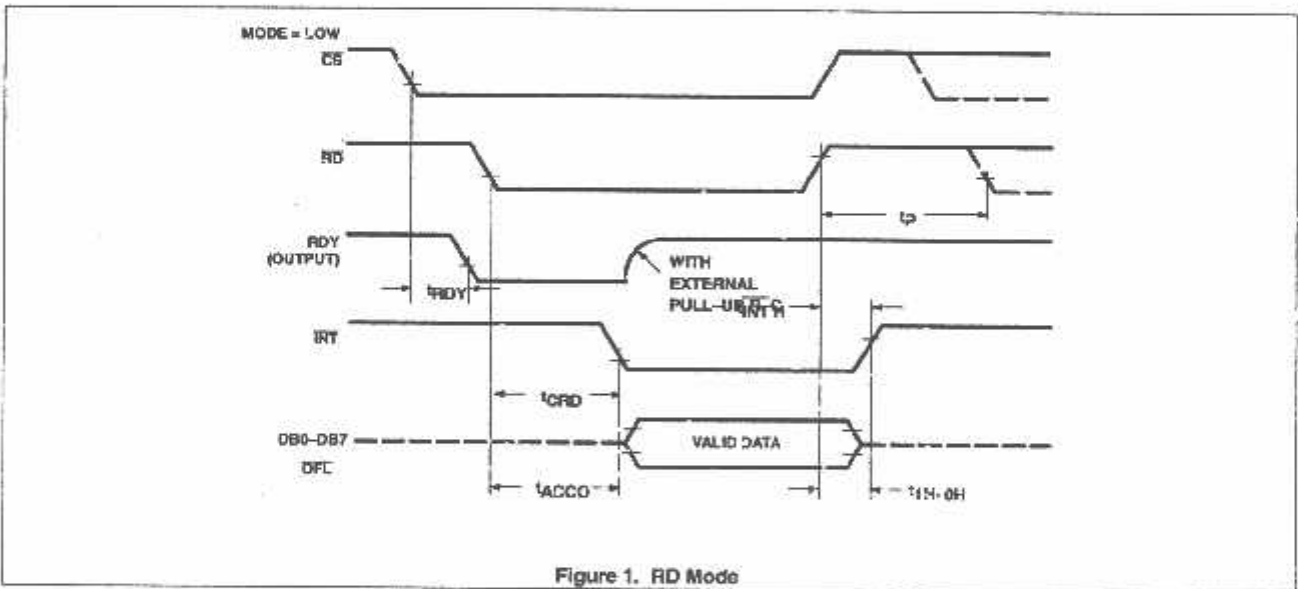
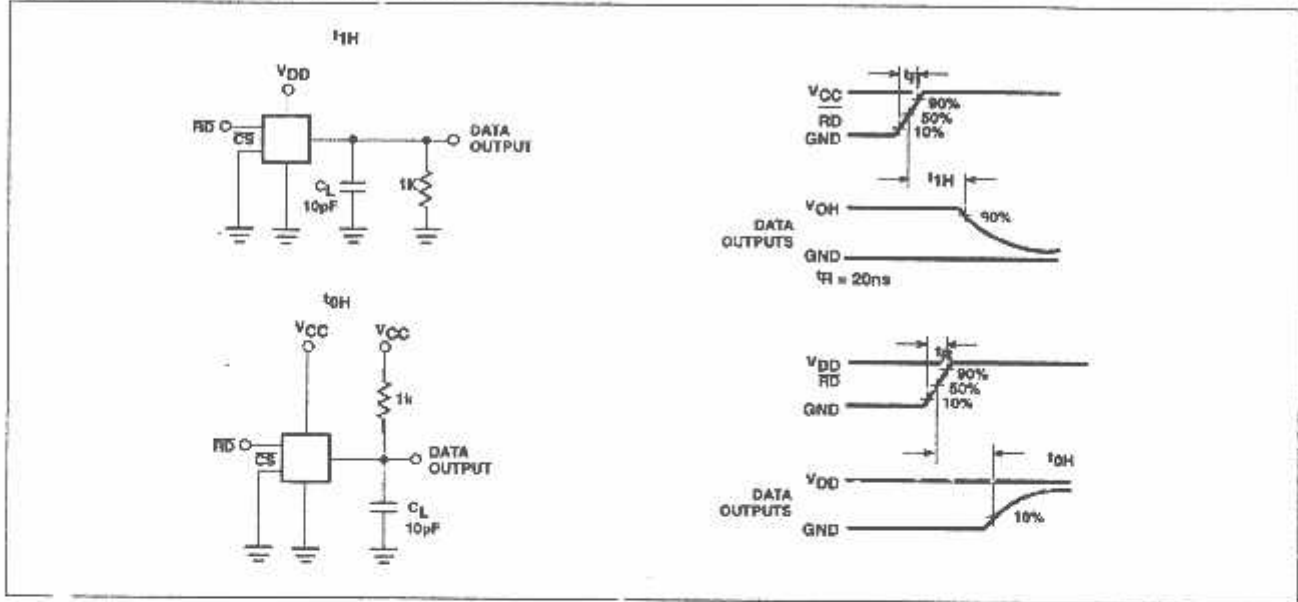


Figure 1. RD Mode

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

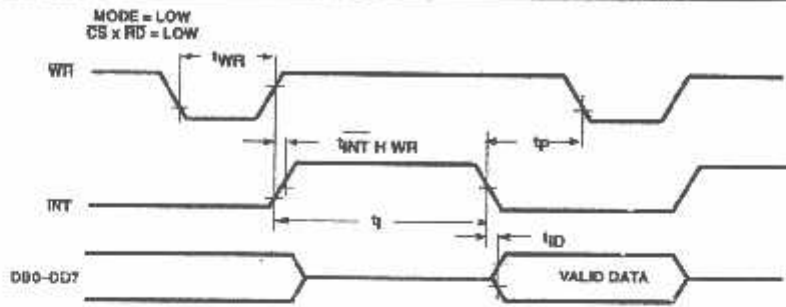
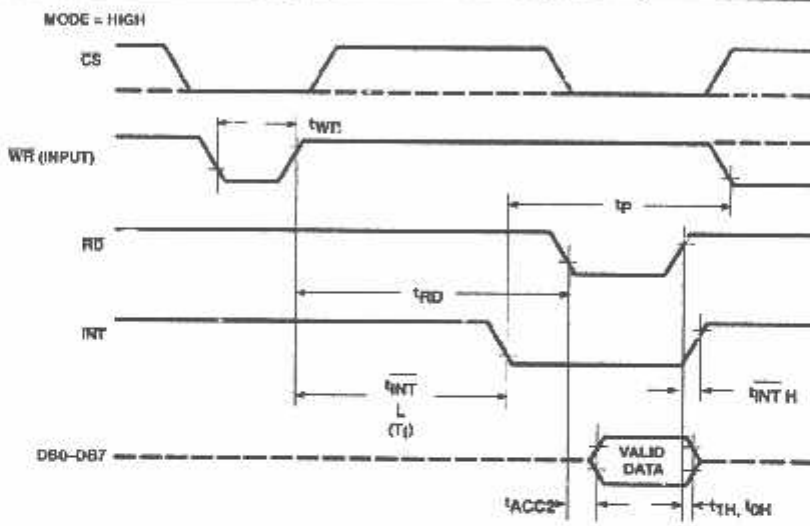
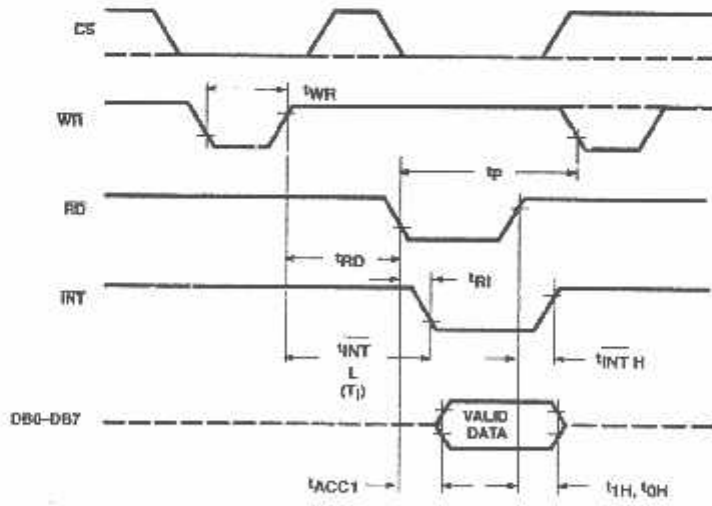


Figure 2. Stand-Alone Mode



a. WR-RD Mode ($t_{RD} > t_q$)



b. WR-RD Mode ($t_{RD} < t_q$)

Figure 3. WR-RD Mode

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

FUNCTIONAL DESCRIPTION

General Operation

The ADC0820 uses two 4-bit flash A/D converters to make an 8-bit measurement (Block Diagram). Each flash ADC is made up of 15 comparators which compare the unknown input to a reference ladder to get a 4-bit result. To take a full 8-bit reading, one flash conversion is done to provide the 4 most significant data bits (via the MS flash ADC). Driven by the 4 MSBs, an internal DAC recreates an analog approximation of the input voltage. This analog signal is then subtracted from the input, and the difference voltage is converted by a second 4-bit flash ADC (the LS ADC), providing the 4 least significant bits of the output data word.

The internal DAC is actually a subsection of the MS flash converter. This is accomplished by using the same resistor ladder for the A/D as well as for generating the DAC signal. The DAC output is actually the tap on the resistor ladder which most closely approximates the analog input. In addition, the "sampled data" comparators used in the ADC0820 provide the ability to compare the magnitudes of several analog signals simultaneously, without using input summing amplifiers. This is especially useful in the LS flash ADC, where the signal to be converted is an analog difference.

The Sampled-Data Comparator

Each comparator in the ADC0820 consists of a CMOS inverter with a capacitively-coupled input (Figure 4). Analog switches connect the two comparator inputs to the input capacitor (C) and also connect the inverter's input and output. This device in effect now has one differential input pair. A comparison requires two cycles, one for zeroing the comparator, and another for making the comparison.

In the first cycle, one input switch and the inverter's feedback switch (Figure 4a) are closed. In this interval, C is charged to the connected input (V1) less the inverter's bias voltage (V_S, approximately 1.6V). In the second cycle (Figure 4b), these two switches are opened and the other (V2) input's switch is closed. The input capacitor now subtracts its stored voltage from the second input and the difference is amplified by the inverter's open loop gain. The inverter's input (V_{S'}) becomes

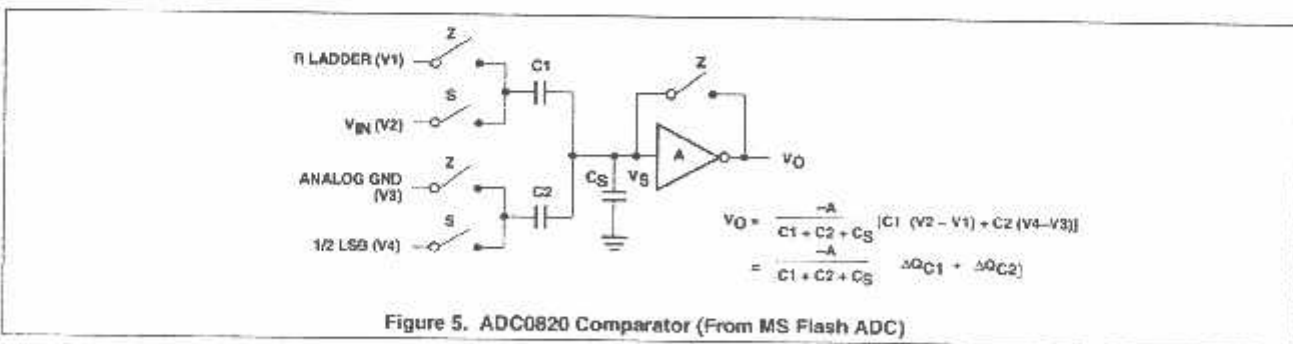
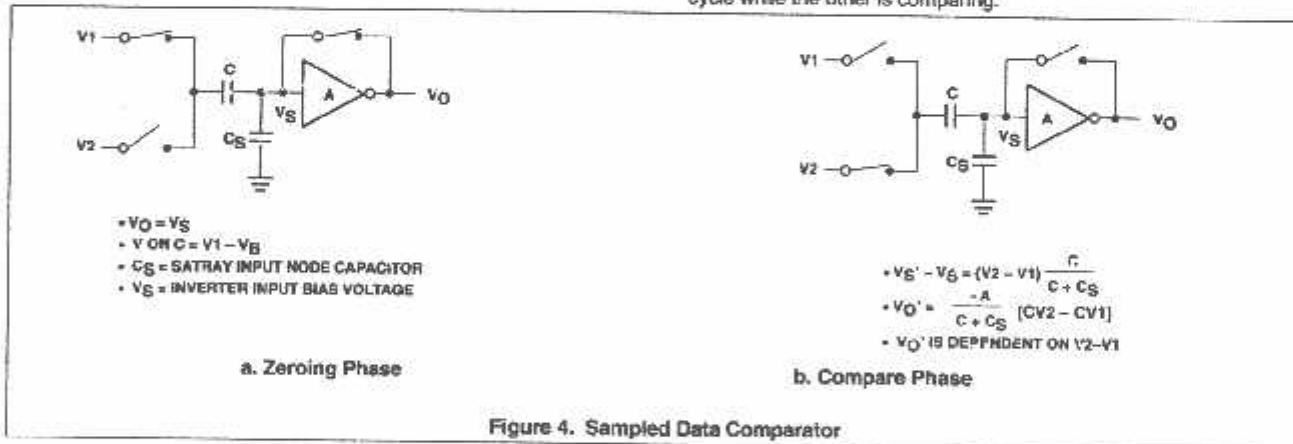
$$V'_S = V_S + (V_2 - V_1) \frac{C}{C + C_S}$$

and the output will go High or Low depending on the sign of V_S' - V_S.

The actual circuitry used in the ADC0820 is a simple but important expansion of the basic comparator described above. By adding a second capacitor and another set of switches to the input (Figure 5), the scheme can be expanded to make dual differential comparisons. In this circuit, the feedback switch and one input switch on each capacitor (Z switches) are closed in the zeroing cycle. A comparison is then made by connecting the second input on each capacitor (S switches) and opening all of the other switches. The change in voltage at the inverter's input, as a result of the change in charge on each input capacitor, will now depend on both input signal differences.

Architecture

In the ADC0820, 15 comparators are used in the MS and LS 4-bit flash A/D converters. The MS (most significant) flash ADC also has one additional comparator to detect input over-range. These two sets of comparators operate alternately, with one group in its zeroing cycle while the other is comparing.



8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

To start a conversion in the WR-RD mode, the WR line is brought Low. At this instant the MS comparators go from zeroing to comparison mode (Figure 8). When WR is returned High after at least 600ns, the output from the first set of comparators (the first flash) is decoded and latched. At this point the two 4-bit converters change modes and the LS (least significant) flash ADC enters its compare cycle. No less than 600ns later, the RD line may be pulled Low to latch the lower four data bits and finish the 8-bit conversion. When RD goes Low, the flash A/Ds change state once again in preparation for the next conversion.

Figure 8 also outlines how the converter's interface timing relates to its analog input (V_{IN}). In WR-RD mode, V_{IN} is measured while WR is Low. In RD mode, sampling occurs during the first 800ns of RD. Because of the input connections to the ADC0820's LS and MS comparators, the converter has the ability to sample V_{IN} at one instant, despite the fact that two separate 4-bit conversions are being done. More specifically, when WR is Low the MS flash is in compare mode (connected to V_{IN}), and the LS flash is in zero mode (also connected to V_{IN}). Therefore both flash ADCs sample V_{IN} at the same time.

Digital Interface

The ADC0820 has two basic interface modes which are selected by strapping the Mode pin High or Low.

RD Mode (Figure 8a)

With the Mode pin grounded, the converter is set to Read mode. In this configuration, a complete conversion is done by pulling RD Low until output data appears. An INT line is provided which goes Low at the end of the conversion as well as a RDY output which can be used to signal a processor that the converter is busy or can also serve as a system Transfer Acknowledge signal.

When in RD mode, the comparator phases are internally triggered. At the falling edge of RD, the MS flash converter goes from zero to compare mode and the LS ADC's comparators enter their zero cycle. After 800ns, data from the MS flash is latched and the LS flash ADC enters compare mode. Following another 800ns, the lower four bits are recovered.

WR Then RD Mode (Figures 8b and c)

With the Mode pin tied High, the A/D will be set up for the WR-RD mode. Here, a conversion is started with the WR input; however, there are two options for reading the output data which relate to interface timing. If an interrupt-driven scheme is desired, the user can wait for INT to go Low

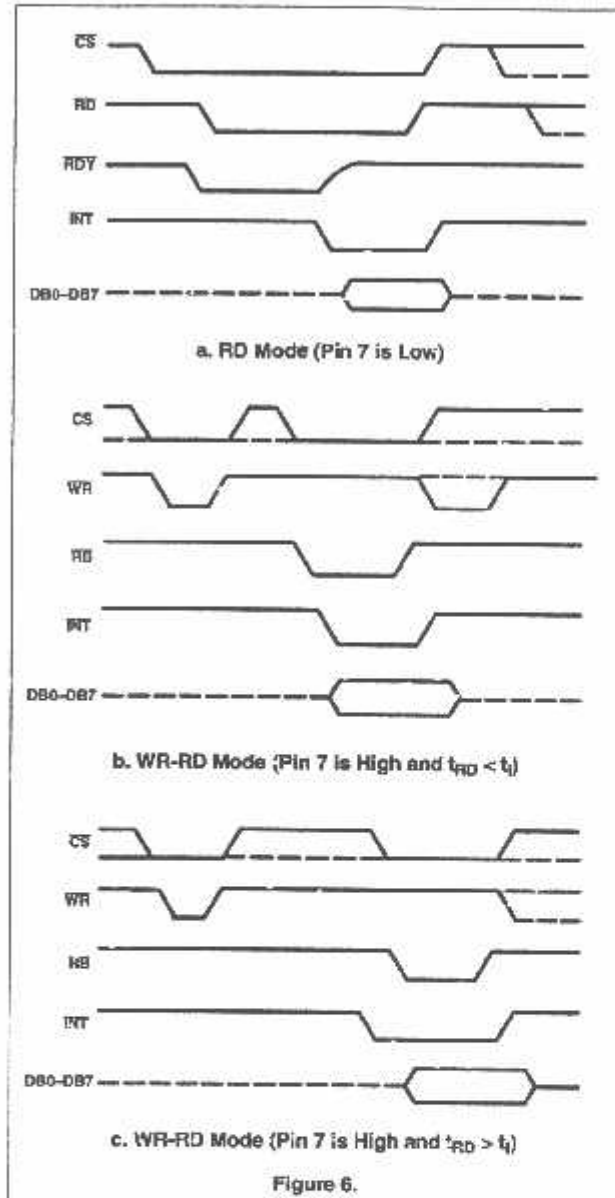


Figure 6.

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

before reading the conversion result. $\overline{\text{INT}}$ will typically go Low 800ns after $\overline{\text{WR}}$'s rising edge. However, if a shorter conversion time is desired, the processor need not wait for $\overline{\text{INT}}$ and can exercise a Read after only 600ns. If this is done, $\overline{\text{INT}}$ will immediately go Low and data will appear at the outputs.

Stand-Alone (Figure 7)

For stand-alone operation in $\overline{\text{WR}}$ -RD mode, $\overline{\text{CS}}$ and RD can be tied Low and a conversion can be started with $\overline{\text{WR}}$. Data will be valid approximately 800ns following $\overline{\text{WR}}$'s rising edge.

Other Interface Considerations

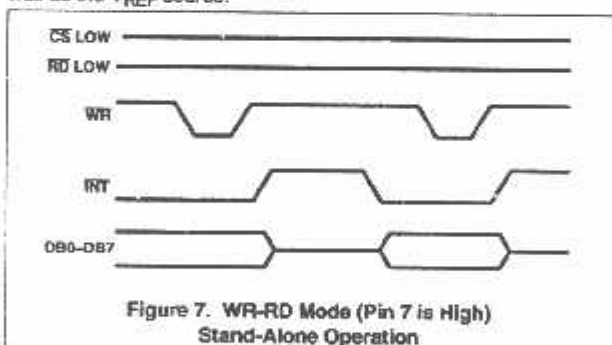
In order to maintain conversion accuracy, $\overline{\text{WR}}$ has a maximum width spec of 50 μ s. When the MS flash ADC's sampled data comparators are in comparison mode ($\overline{\text{WR}}$ is Low), the input capacitors (C, Figure 5) must hold their charge. Switch leakage can cause errors if the comparator is left in this phase for too long.

Since the MS flash ADC enters its zeroing phase at the end of a conversion, a new conversion cannot be started until this phase is complete. The minimum spec for this time is 500ns (t_p in Figures 1, 2, 3a, and 3b).

ANALOG CONSIDERATIONS

Reference and Input

The two V_{REF} inputs of the ADC0820 are fully differential and define the zero- to full-scale input range of the A/D converter. This allows the designer to easily vary the span of the analog input since this range will be equivalent to the voltage difference between $V_{\text{IN}}(+)$ and $V_{\text{IN}}(-)$. By reducing V_{REF} ($V_{\text{REF}}=V_{\text{REF}}(+)-V_{\text{REF}}(-)$) to less than 5V, the sensitivity of the converter can be increased (i.e., if $V_{\text{REF}}=2\text{V}$, then 1 LSB=7.8mV). The input/reference arrangement also facilitates ratiometric operation and, in many cases, the chip power supply can be used for transducer power as well as the V_{REF} source.



This reference flexibility lets the input span not only be varied, but also offset from zero. The voltage at $V_{\text{REF}}(-)$ sets the input level which produces a digital output of all zeroes. Though V_{IN} is not itself differential, the reference design affords nearly differential-input capability for most measurement applications. Figure 9 shows some of the configurations that are possible.

Input Current

Due to the unique conversion techniques employed by the ADC0820, the analog input behaves somewhat differently than in conventional devices. The A/D's sampled data comparators take varying amounts of input current depending on which cycle the conversion is in.

The equivalent input circuit of the ADC0820 is shown in Figure 10a. When a conversion starts ($\overline{\text{WR}}$ Low, $\overline{\text{WR}}$ -RD mode), all input switches close, connecting V_{IN} to 31 1pF capacitors. Although the two 4-bit flash circuits are not both in their compare cycle at the same time, V_{IN} still sees all input capacitors at once. This is because the MS flash converter is connected to the input during its compare interval and the LS flash is connected to the input during its zeroing phase. In other words, the LS ADC uses V_{IN} as its zero-phase input.

The input capacitors must charge to the input voltage through the on resistance of the analog switches (about 5k Ω to 10k Ω). In addition, about 12pF of input stray capacitance must also be charged. For large source resistances, the analog input can be modeled as an RC network as shown in Figure 10b. As R_S increases, it will take longer for the input capacitance to charge.

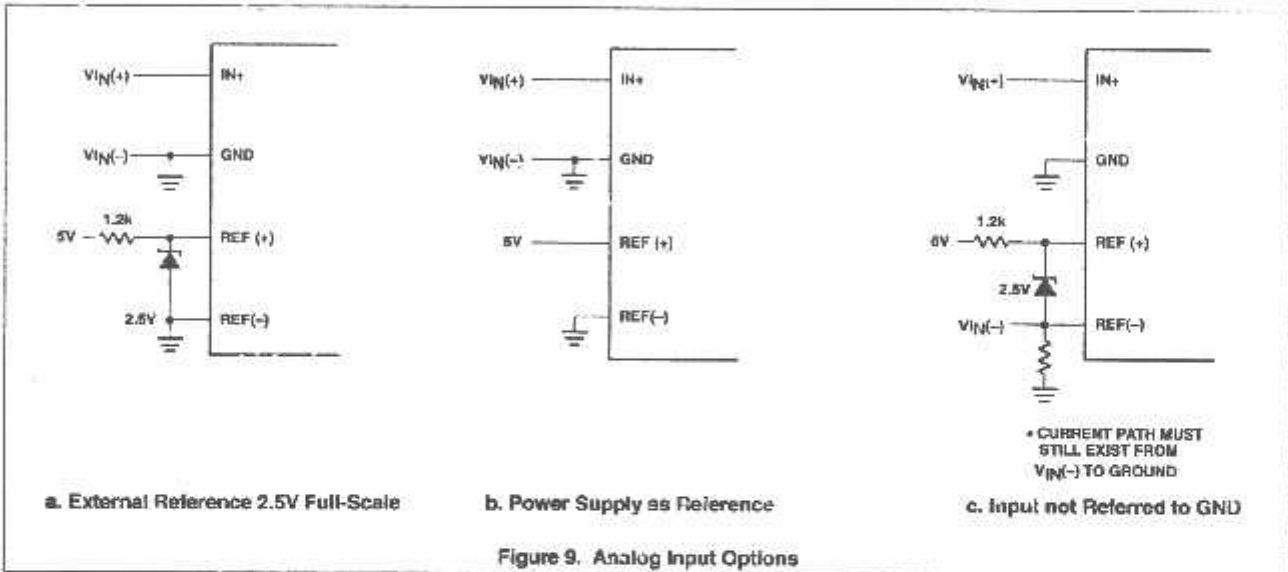
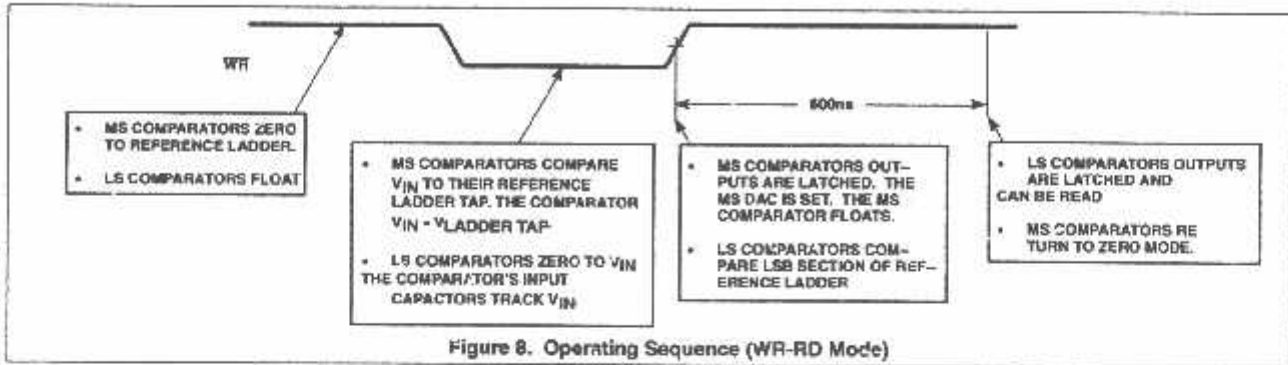
In RD mode, the input switches are closed for approximately 800ns at the start of the conversion. In $\overline{\text{WR}}$ -RD mode, the time that the switches are closed to allow this charging is the time that $\overline{\text{WR}}$ is Low. Since other factors force this time to be at least 600ns, input time constants of 100ns can be accommodated without special consideration. Typical total input capacitance values of 45pF allow R_S to be 1.5k Ω without lengthening $\overline{\text{WR}}$ to give V_{IN} more time to settle.

Input Filtering

It should be made clear that transients in the analog input signal, caused by charging current flowing into V_{IN} , will not degrade the A/D's performance in most cases. In effect, the ADC0820 does not "look" at the input when these transients occur. The comparators' outputs are not latched while $\overline{\text{WR}}$ is Low, so at least 600ns will be provided to charge the ADC's input capacitance. It is

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820



8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

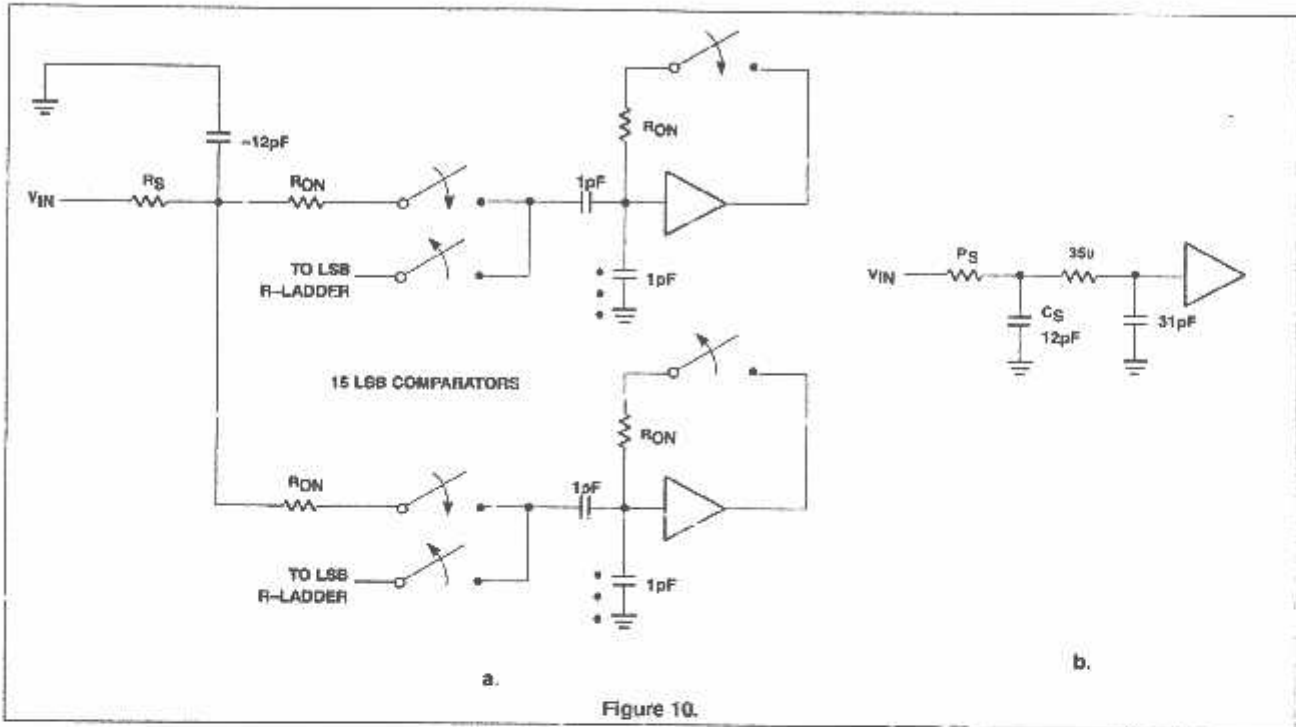


Figure 10.

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

therefore not necessary to filter out these transients by putting an external cap on the V_{IN} terminal, if an input amplifier that can settle within 600ns is used to drive the input. The NE530 is a suitable op amp for driving the input of the ADC0820.

Inherent Sample-Hold

Another benefit of the ADC0820's input mechanism is its ability to measure a variety of high-speed signals without the help of an external sample-and-hold. In a conventional SAR type converter, regardless of its speed, the input must remain at least 1/2LSB stable throughout the conversion process if full accuracy is to be maintained. Consequently, for many high-speed signals, this signal must be externally sampled, and held stationary during the conversion.

Sampled data comparators, by nature of their input switching, already accomplish this function to a large degree (Section 1.2). Although the conversion time for the ADC0820 is 1.5 μ s, the time through which V_{IN} must be 1/2LSB stable is much smaller. Since the MS flash ADC uses V_{IN} as its "compare" input and the LS ADC uses V_{IN} as its "zero" input, the ADC0820 only "samples" V_{IN} when WR is Low. Even though the two flashes are not done simultaneously, the analog signal is measured at one instant. The value of V_{IN} approximately 100ns after the rising edge of WR (100ns due to internal logic propagation delay) will be the measured value.

Input signals with slew rates typically below 100mV/ μ s can be converted without error. However, because of the input time constants, and charge injection through the opened comparator input switches, faster signals may cause errors. Still, the ADC0820's loss in accuracy for a given increase in signal slope is far less than what would be witnessed in a conventional successive approximation device. An SAR type converter with a conversion time as fast as 1 μ s would still not be able to measure a 5V, 1kHz sine wave without the aid of an external sample-and-hold. The ADC0820, with no such help, can typically measure 5V, 7kHz waveforms.

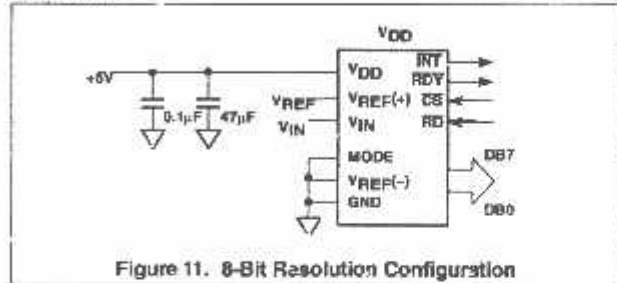


Figure 11. 8-Bit Resolution Configuration

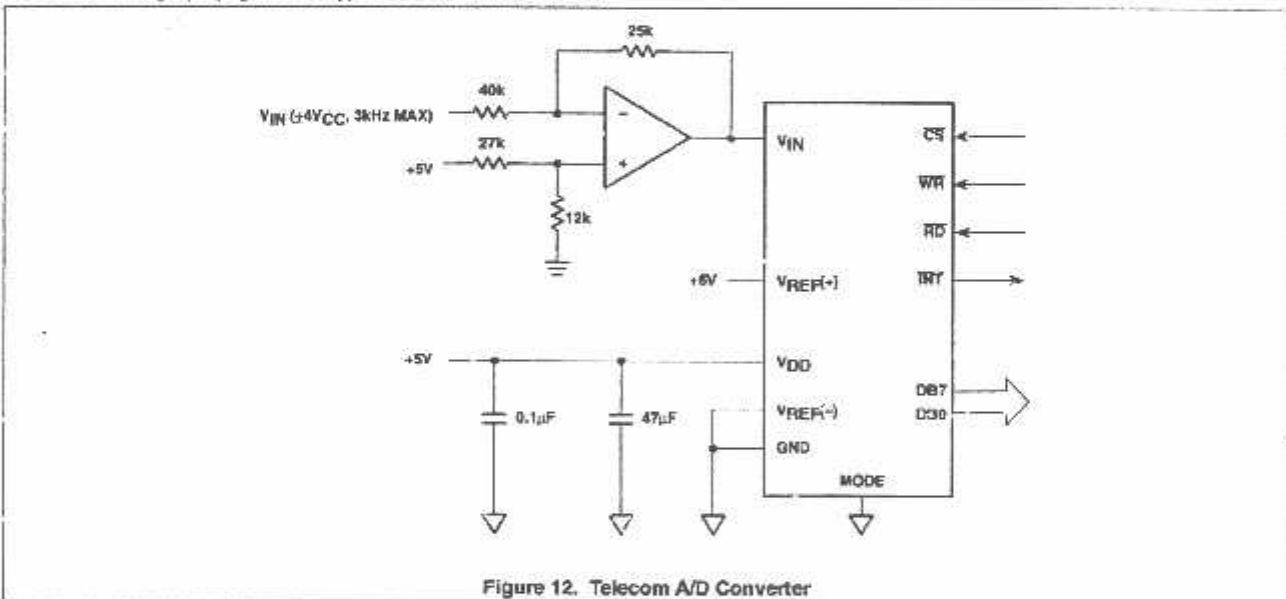


Figure 12. Telecom A/D Converter

8-Bit, high-speed, μ P-compatible A/D converter with track/hold function

ADC0820

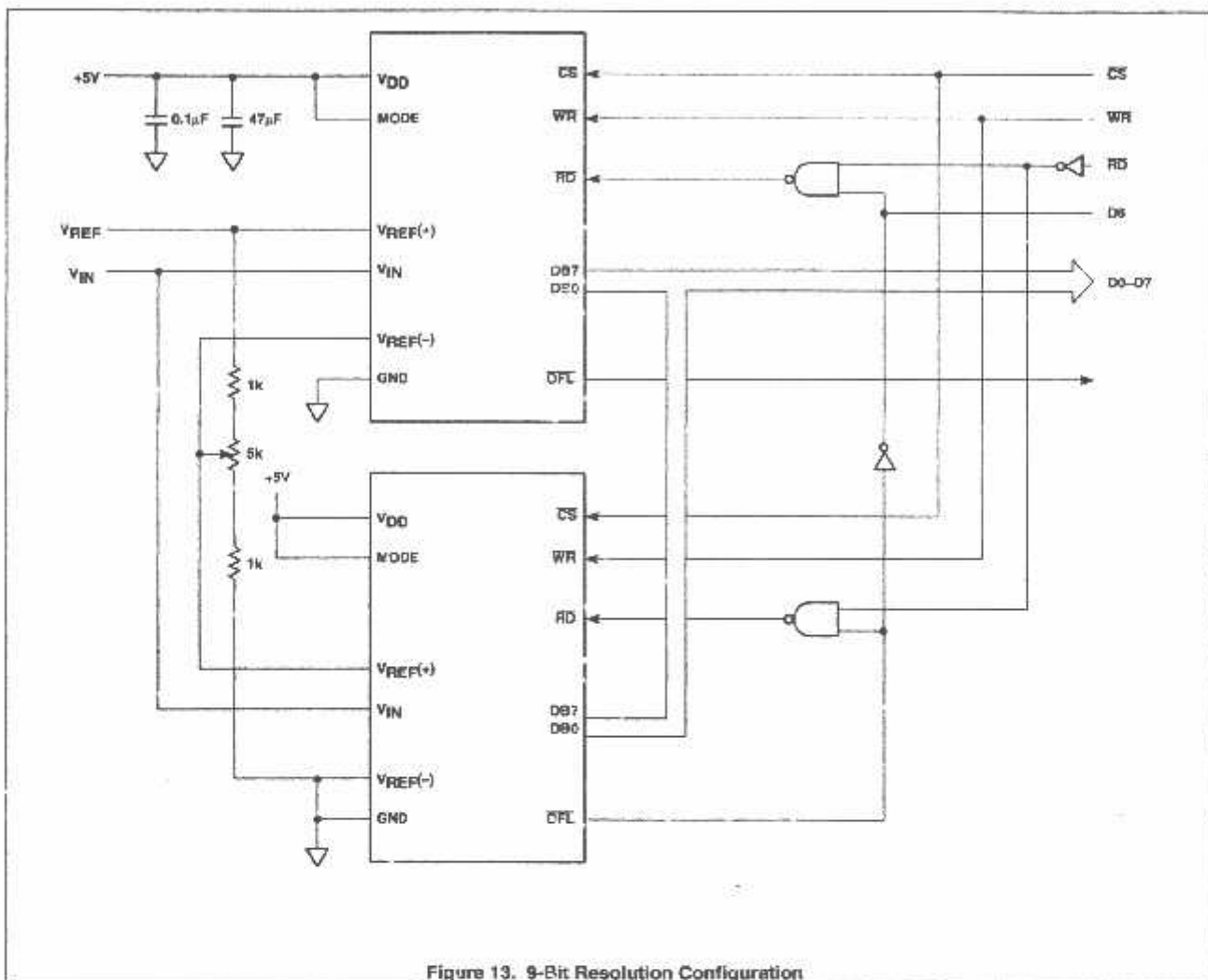


Figure 13. 9-Bit Resolution Configuration

**APLIKASI MIKROKONTROLER AT89C51
SEBAGAI PEMBANGKIT PWM SINUSOIDA 1 FASA
UNTUK MENGENDALIKAN PUTARAN MOTOR SINKRON**

Hari Wahyu W¹, Bambang Sutopo²

¹ Penulis, Mahasiswa S-1 Teknik Elektro – UGM, Yogyakarta

² Dosen Pembimbing I, Staff Pengajar pada Jurusan Teknik Elektro – UGM, Yogyakarta

ABSTRACT

Single phase pulse width modulation inverter is a circuit which convert DC voltage to AC voltage for one phase. Generating PWM signal digitally give good performance because their immune from noisy. Designing a PWM signal generator using microcontroller has several advantages, such as easy to programed and network inverter become modestly. The aim of this thesis is designing generating of signal PWM one phase by using microcontroller AT89C51. By using this inverter, hence operation of speed of motor AC can be controlled with more carefully. This network inverter is designed so that summarize, therefore a minimum system of mikrokontroler only rely on the single chip mode.

Observation shows that the design of PWM generator work well. The PWM signal which produced has 24 pulse each periode and frequency interval between 20 – 66,67 Hz with the increase and degradation of each every 0,067 Hz.

ABSTRAKSI

Pulse Width Modulation Inverter satu fase adalah rangkaian pengubah tegangan searah menjadi tegangan bolak balik untuk satu fase. Pembangkitan sinyal PWM secara digital dapat memberikan unjuk kerja sistem yang bagus karena lebih kebal terhadap gangguan/derau. Perancangan sebuah pembangkit sinyal PWM menggunakan mikrokontroler memiliki beberapa keuntungan yaitu mudah diprogram dan rangkaian inverter menjadi sederhana. Tujuan tugas akhir ini adalah merancang pembangkit sinyal PWM satu fase dengan menggunakan mikrokontroler AT89C51. Dengan menggunakan inverter ini, maka pengendalian kecepatan motor AC dapat dilakukan dengan lebih teliti. Rangkaian inverter ini dirancang supaya ringkas, oleh karena itu pada sistem minimal mikrokontroler hanya mengandalkan ragam chip tunggal.

Hasil pengamatan menunjukkan bahwa rancangan pembangkit PWM telah berfungsi dengan baik. Sinyal PWM yang dibangkitkan memiliki 24 pulsa setiap periode dan rentang frekuensi antara 20 – 66,67 Hz dengan kenaikan dan penurunan setiap 0,067 Hz.

Keyword: Inverter PWM Satu Fase, AT89C51

1. Pendahuluan

Motor AC memiliki keunggulan dalam hal kesederhanaan dan murah biaya perawatan sehingga jenis motor ini banyak dipakai di lingkungan industri maupun rumah tangga. Pengendalian kecepatan putaran motor AC dapat dilakukan dengan beberapa cara diantaranya dengan kendali tegangan dan frekuensi.

Inverter adalah konverter DC ke AC dengan tegangan dan frekuensi keluaran dapat diatur sehingga motor AC dapat dikendalikan dengan fleksibel. Ada beberapa jenis inverter diantaranya adalah inverter PWM (*Pulse Width Modulation*). Keuntungan operasi inverter PWM sebagai teknik konversi dibanding dengan jenis-jenis inverter lainnya adalah rendahnya distorsi harmonik pada tegangan keluaran dibanding dengan jenis inverter lainnya. Selain itu teknik PWM sangat

praktis dan ekonomis untuk diterapkan berkat semakin pesatnya perkembangan komponen semikonduktor (terutama komponen daya yang mempunyai waktu penyaktaran sangat cepat).

Pada pengendalian kecepatan motor AC, inverter PWM mempunyai kelebihan yaitu mampu menggerakkan motor induksi dengan putaran halus dan rentang yang lebar. Selain itu apabila pembangkitan sinyal PWM dilakukan secara digital akan dapat diperoleh unjuk kerja sistem yang bagus karena lebih kebal terhadap derau.

2. Tinjauan Pustaka

Perancangan dengan FPGA dapat dilakukan dengan cepat, mudah dimodifikasi dan sesuai untuk *prototyping*, tetapi akan relatif mahal dan tidak ekonomis untuk produksi yang besar (Sutopo, 2000). Penggunaan dengan ASIC

(Application Specific Integrated Circuit) akan lebih sesuai untuk produksi besar, tetapi perancangan dengan ASIC akan lebih kompleks dan memerlukan waktu yang lebih lama.

Menurut Agus Bejo (2003), pembangkitan sinyal dengan menggunakan FPGA, di satu sisi dapat memenuhi tuntutan akan kecepatan tetapi disisi lain kurang fleksibel dalam pengoperasian kendalinya. Frekuensi sinyal PWM yang dihasilkan memiliki rentang antara 3 - 110 Hz dengan kenaikan terkecil 1 Hz. Penggunaar mikrokontroler dalam pembangkitan sinyal PWM di satu sisi memiliki kelebihan berupa fleksibilitas dan realibilitas namun disisi lain memiliki kekurangan dalam hal beban komputasi. Oleh karena itu diperlukan suatu cara agar komputasi PWM tidak membebani mikrokontroler.

3. Dasar Teori

Inverter PWM Sinusoida satu fase menghasilkan pulsa PWM bolak balik satu fase dengan nilai tegangan bolak balik efektifnya dirumuskan sebagai berikut:

$$V_{rms} = \sqrt{\frac{1}{T} \int_0^T v^2 dt} \quad (1)$$

dengan V_{rms} = tegangan efektif
 v = fungsi tegangan
 T = periode

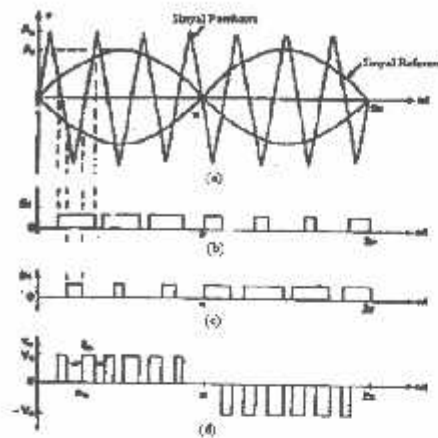
Oleh karena pada inverter SPWM nilai tegangan masukan DC adalah konstan maka tegangan rms dapat juga dirumuskan :

$$V_{rms} = V_{DC} \sqrt{\frac{\sum t_p}{T}} \quad (2)$$

dengan V_{rms} = tegangan efektif
 V_{DC} = tegangan searah inverter
 t_p = lebar pulsa tinggi dalam 1 periode
 T = periode

Untuk menghasilkan sinyal PWM tersebut dapat menggunakan 2 buah sinyal sinus dan 1 sinyal segitiga atau dengan menggunakan 1 buah sinyal sinus dan 2 buah sinyal segitiga.

Pada proses pembangkitan SPWM dengan menggunakan 2 buah sinyal sinus dan sebuah sinyal segitiga, dilakukan perbandingan amplitudo antara sinyal segitiga dengan sinyal sinus. Sinyal penggerak akan dibangkitkan apabila amplitudo sinyal sinus lebih besar daripada amplitudo sinyal segitiga. Masing-masing sinyal penggerak digunakan untuk penyaktaran sehingga diperoleh sinyal PWM. Proses pembangkitan SPWM tersebut dapat dilihat pada gambar 1.



Gambar 1. (a) Proses perbandingan antara sinyal pembawa dengan sinyal referensi, (b) Sinyal penggerak A, (c) Sinyal penggerak BN, (d) Sinyal SPWM

Proses pembangkitan SPWM secara digital dapat dilakukan dengan 2 cara, yaitu:

1. Dengan membangkitkan gelombang segitiga dan gelombang sinus secara diskret dengan metode *look up table*. Kemudian dilakukan perbandingan untuk masing-masing nilai amplitudo gelombang sinus dan segitiga seperti pada gambar 1. Cara ini sama halnya dengan membangkitkan gelombang sinus analog dan gelombang segitiga analog secara digital.
2. Dengan mencari terlebih dahulu waktu untuk setiap pulsa masing-masing sinyal penggerak, untuk dijadikan data dalam proses pembangkitan sinyal penggerak secara *look up table*. Cara inilah yang dipakai dalam perancangan tugas akhir ini.

4. Metodologi Penelitian

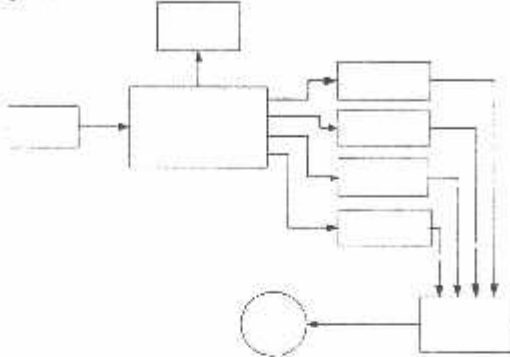
Metodologi yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Studi literatur mengenai Inverter PWM satu fase dan Mikrokontroler AT89C51.
2. Merancang dan mensimulasi PWM Satu Fase secara software dengan simulink dari MATLAB 6.0.
3. Merancang dan membuat sistem secara hardware.
4. Merancang perangkat lunak pembangkit sinyal penggerak dengan menggunakan bahasa Assembly.
5. Menguji dan mengambil data dari perancangan.
6. Menganalisa hasil dan membuat kesimpulan.

5. Hasil Implementasi dan Pembahasan

5.1 Perancangan Sistem

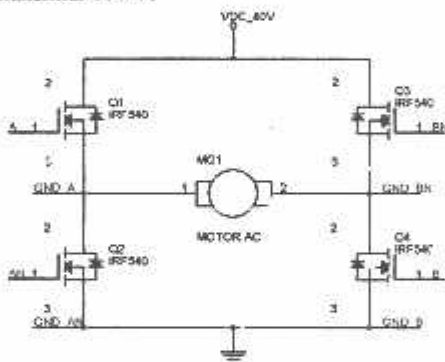
Secara lebih detailnya sistem pengendalian kecepatan motor sinkron dapat dilihat dalam gambar 2.



Gambar 2. Diagram kotak inverter PWM 1 fasa

A. Untai H-Bridge

Rangkaian *H-bridge* berfungsi sebagai rangkaian penyalur tegangan sekitar $80 V_{DC}$. Untuk melakukan penyaluran, diperlukan MOSFET yaitu IRF 540N yang memiliki waktu naik dan waktu turun yang cepat, dan V_{DS} maksimal 100 V.

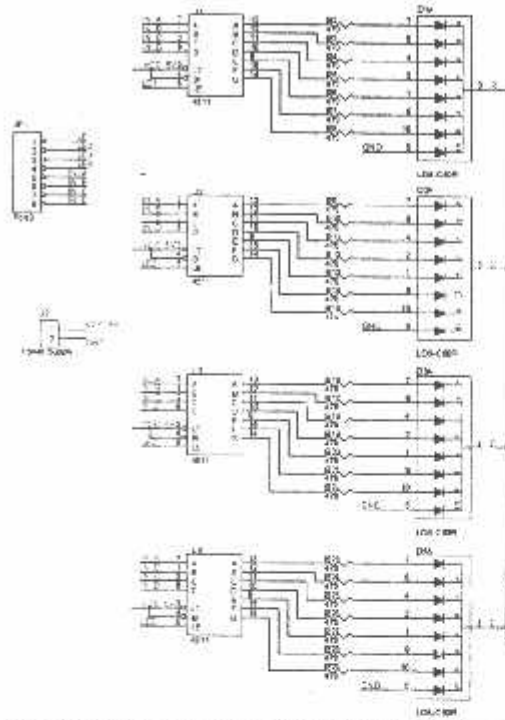


Gambar 3 Rangkaian H-bridge

B. Untai Penampil 7 Segmen

Untuk memudahkan dalam proses pemrograman, maka digunakan IC *BCD to 7 Segment Latch* (4511). Dengan menggunakan IC ini, maka untuk pengiriman ke setiap segmen hanya sebanyak 2 kali, yaitu proses pengisian data di IC, dan proses penguncian (*latch*).

Program penampil 7 segmen dirancang untuk menampilkan data hasil masukan dari input yang telah disimpan di RAM alamat 27h, 28h dan 29h untuk ditampilkan ke penampil 7 segmen digit ke-2, 3 dan 4. Sedangkan digit ke-1 tidak digunakan karena keterbatasan RAM internal.



Gambar 4 Rancangan rangkaian penampil 7 segmen

Yang pertama-tama dilakukan yaitu mengambil data dari RAM, kemudian 4 bit high data tersebut di-OR-kan dengan bit untuk menonaktifkan *latch* (bit 0 pada *pin latch enable*) dari IC 4511. Kemudian data tersebut dibalik 4 bit tinggi dengan 4 bit rendahnya lalu dikirim ke *port 2*. Setelah itu 4 bit rendahnya dibuat 1 semua untuk menahan data di IC dengan menggunakan instruksi ORL lalu dikirimkan ke *port 2*. Langkah tersebut diulang sampai data dari RAM alamat 27h sampai 29h tertampil ke penampil 7 segmen.

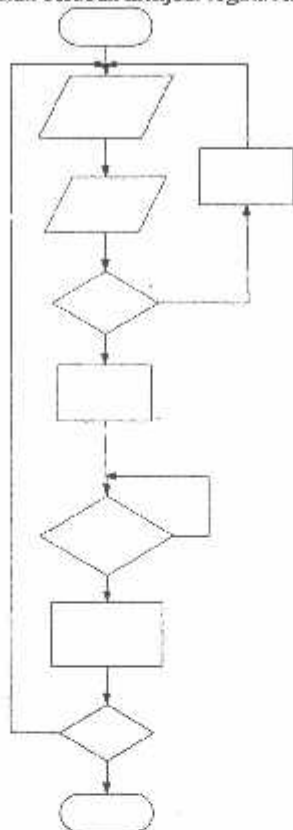
C. Untai Keypad



Gambar 5 Rancangan posisi keypad

Metode yang digunakan yaitu dengan menggunakan pemayaran kolom secara urut. Pada saat pemayaran kolom 1 (C1) maka bit 0 diberi logika rendah, sedangkan bit yang lain diberi logika tinggi. Apabila tidak ada penekanan, maka kondisi hasil pembacaan *port* menunjukkan data yang sama dengan pengeluaran data tadi. Apabila ada

penekanan salah satu saklar, maka salah satu dari 4 bit tinggi akan berubah menjadi logika rendah



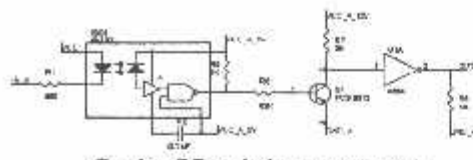
Gambar 6 Diagram alir program keypad

D. Sistem Minimal AT89C51

Pengendali yang dirancang adalah menggunakan mikrokontroler dan bekerja dalam ragam *single chip operation* (mode operasi keping tunggal) yang tidak memerlukan memori luar. Kristal yang digunakan untuk mengoperasikan mikrokontroler adalah 12 MHz. Penggunaan kristal 12 MHz menyebabkan detak dalam pada mikrokontroler menjadi $12 \text{ Mhz} / 12 = 1 \text{ MHz}$, yang artinya setiap periode detak waktunya 1 mikrodetik, sehingga memudahkan untuk mengubah-ubah data pada penggunaan pewaktu karena periode detak pewaktuannya tidak ganjil.

E. Untai Penggerak

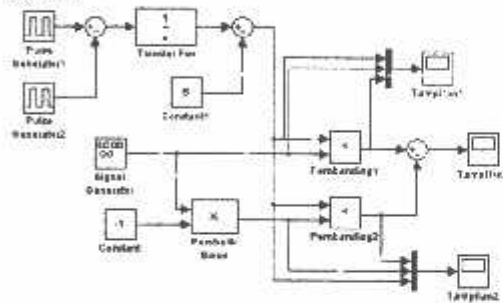
Rangkaian penggerak berfungsi untuk mengisolasi antara rangkaian sistem minimal AT89C51 terhadap rangkaian *H-bridge*. Selain itu, tugas yang lainnya yaitu menyesuaikan tegangan antara mikrokontroler dengan tegangan kerja dari rangkaian *H-bridge*.



Gambar 7 Rangkaian penggerak A

F. Perangkat Lunak Pembangkit PWM

Untuk mendapatkan data pewaktuan masing-masing sinyal penggerak PWM, maka terlebih dahulu dibuat simulasi pembangkitan sinyal PWM dengan menggunakan MATLAB. Dengan menggunakan fasilitas simulink di MATLAB maka didapatkan simulasi seperti pada gambar 8 sebagai berikut:



Gambar 8 Simulink MATLAB untuk simulasi PWM

Pada tugas akhir ini akan dibuat inverter PWM sinusoida 12 pulsa setiap setengah periode. Untuk mendapatkan sinyal penggerak PWM 12 pulsa, maka dibutuhkan 13 buah sinyal segitiga setiap 1 gelombang sinusoida. Karena di pustaka simulink MATLAB tidak ada pembangkit gelombang segitiga, maka cara lainnya yaitu dengan mengintegrasikan gelombang kotak. Amplitudo gelombang segitiga dibuat 5 V.

Blok pembanding1 dan pembanding2 akan membandingkan antara gelombang segitiga dengan gelombang sinusoida. Cara kerjanya yaitu jika nilai gelombang segitiga lebih kecil dari gelombang sinusoida pada saat t yang sama, maka hasilnya 1. Apabila sebaliknya, maka hasilnya 0.

Data diambil dengan menghitung masing-masing pewaktuan sinyal penggerak pada saat tinggi dan rendah sampai seperempat periode. Data tidak diambil selama satu periode karena setelah mencapai seperempat periode, maka pewaktuan untuk logika tinggi dan rendah sinyal penggerak merupakan perulangan dari pewaktuan yang seperempat periode pertama tadi, sehingga menghemat penggunaan ROM. Berarti ada 7 bit data yang harus diambil selama seperempat periode pertama. Pola perulangan pewaktuan sinyal penggerak dan pengambilan ketujuh buah data dapat dilihat pada gambar 9.

Mulai

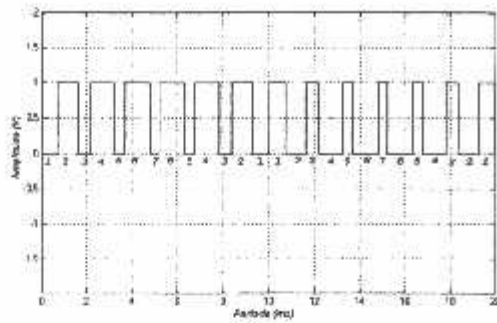
Output data untuk Scan Ci

Deteksi baris (R)

Parity error?

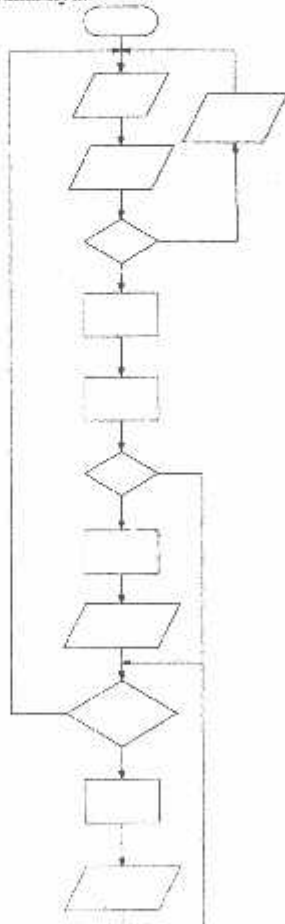
P=0

Delay antibouncing



Gambar 9 Proses pengambilan data sinyal penggerak

Perancangan perangkat lunak menggunakan bahasa assembler. Program dibuat untuk menghasilkan secara langsung keempat sinyal penggerak PWM dengan frekuensi dan indeks modulasi tertentu sesuai dengan masukan kecepatan RPM motor. Metode yang digunakan yaitu *look up table* berdasarkan data yang telah didapat dari hasil simulasi MATLAB. Berikut ini adalah diagram alir perangkat lunaknya:

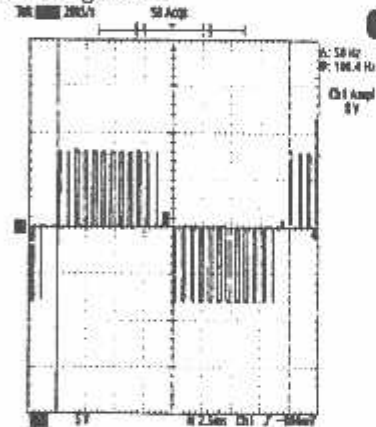


Gambar 10 Diagram alir program utama pembangkit sinyal penggerak

Karena data yang akan diolah besarnya 2 byte maka pengolahannya menggunakan program perkalian 16 bit yang akan menghasilkan data 32 bit, dan pembagiannya menggunakan program pembagian 32 bit yang akan menghasilkan data 16 bit.

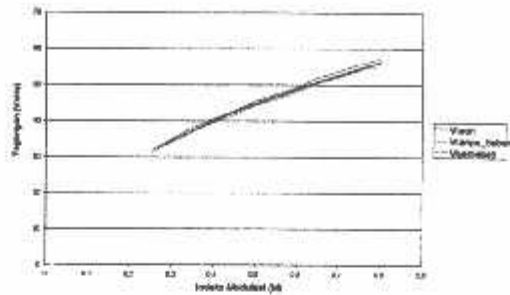
Motor sinkron tidak dapat langsung diputar pada kecepatan tinggi karena dalam keadaan awal rotor belum berputar, dibutuhkan torsi yang lebih besar daripada setelah berputar. Untuk memutar motor agar diperoleh kecepatan yang tinggi, maka pertama-tama motor diputar pada 300 RPM dengan tegangan di atas tegangan operasional. Setelah motor berputar, lalu tegangan diturunkan pada tegangan operasional, dan dinaikkan frekuensi dan tegangannya untuk mendapatkan kecepatan yang tinggi. Oleh karena itu dibutuhkan program untuk *starting* untuk memutar motor ke putaran rendah dengan tegangan yang lebih tinggi daripada tegangan *steady state*-nya pada putaran yang sama.

5.2 Hasil Pengamatan



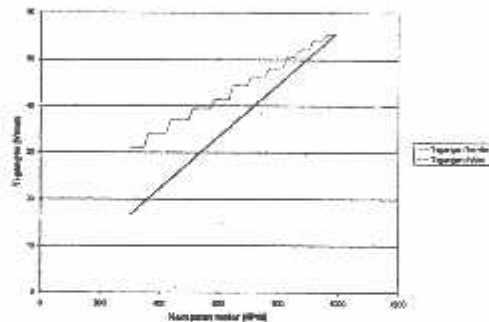
Gambar 11 Sinyal keluaran inverter pada frekuensi masukan 50 Hz

Gambar 11 merupakan salah satu hasil pengamatan sinyal keluaran inverter dengan menggunakan osiloskop. Pada pengamatan ini digunakan masukan data 750 RPM sehingga dengan menggunakan rumus (2.6) diperoleh frekuensi tegangan AC yaitu 50 Hz. Karena menggunakan *probe* dengan faktor pengali 10x, maka berdasarkan hasil osiloskop terukur tegangan 80 V.



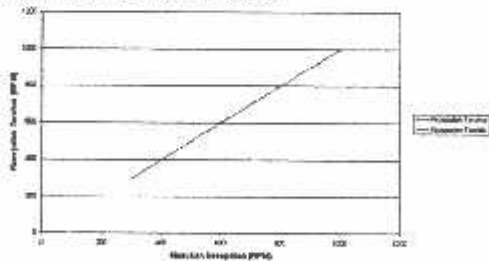
Gambar 12 Karakteristik nilai indeks modulasi terhadap tegangan SPWM

Dari gambar 12, tampak bahwa tegangan yang terukur tanpa beban hampir sama dengan tegangan teori. Apabila inverter diberi beban maka tegangan turun sekitar 1 V. Hal ini disebabkan karena MOSFET masih mempunyai hambatan dalam pada saat ON walaupun sangat kecil sekitar $44 \text{ m}\Omega$ (data dari datasheets) dan terpasang seri dengan beban.



Gambar 13 Karakteristik tegangan terhadap putaran motor

Deri gambar 13 tampak bahwa kenaikan tegangan yang terukur tidak linear karena kenaikan indeks modulasi setiap 0.25 sehingga untuk membuat kenaikan tegangan secara linear tidak bisa karena jarak untuk setiap kenaikan indeks modulasi masih terlalu besar.



Gambar 14 Hasil pengamatan kecepatan terukur terhadap kecepatan teoritis

Hasil yang diperoleh dari gambar 14 bahwa kecepatan terukur hampir sama dengan kecepatan

teoritis (masukan data RPM) dengan selisih terbesar 0,2 RPM.

6. Kesimpulan

Dari hasil simulasi, pengamatan dan pengujian pada rancangan yang dibuat dapat diperoleh kesimpulan sebagai berikut :

1. Frekuensi sinyal PWM yang dihasilkan memiliki rentang antara 20 – 66,6 Hertz dengan kenaikan setiap 0,067 Hz yang digunakan untuk memutar motor dari 300 – 999 RPM dengan kenaikan setiap 1 RPM.
2. Pengendalian kecepatan motor hanya 3 digit RPM saja karena keterbatasan RAM dalam AT89C51.
3. Tegangan efektif keluaran (V_{rms}) inverter berbanding lurus dengan nilai indeks modulasinya.
4. Frekuensi sinyal PWM yang dihasilkan sama dengan frekuensi masukan yang diharapkan, yang ditunjukkan dengan kecepatan putaran motor sama dengan masukan kecepatan motor.
5. Untuk mengendalikan kecepatan putaran motor sinkron, kerjakan tegangan terhadap frekuensi belum dapat dilakukan secara proporsional karena kenaikan indeks modulasi masih terlalu besar.

7. Daftar Pustaka

- Agfianto, E.P., 2002, *Belajar Mikrokontroler AT89C51/52/53 (Teori dan Aplikasi)*, Gava Media, Yogyakarta.
- Bejo, A., 2003, *PWM Saja Fcra Berbasis FPGA Dengan Kendali Mikrokontroler*, Tugas Akhir, Universitas Gadjah Mada, Yogyakarta (tidak diterbitkan).
- Chapman, J.S., 1985, *Electric Machinery Fundamentals*, McGraw-Hill Book Company, Singapore.
- Kenjo, T., 1990, *Power Electronics for The Microprocessor Age*, Oxford University Press, New York.
- Nalwan, A.P., 2003, *Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51*, Elex Media Komputindo, Jakarta.
- Rashid, H.M., 1999, *Power Electronics Circuits, Devices, and Applications*, Prentice Hall, New Delhi.

Makalah Sidang Praktikum Teknik Digital
**PENGENDALIAN MOTOR STEPPER METODE HALF STEP DENGAN
RANGKAIAN DIGITAL.**
Harindra Wisnu P (L2F004481)

Abstrak

Perkembangan teknologi di bidang elektronika semakin pesat. Seiring kebutuhan, berkembang pula teknologi digital yang tidak lagi bergantung pada level-level tegangan yang linear melainkan kondisi '1' atau '0' penggunaannya pun semakin berkembang didukung sistem biner yang memudahkan pengembangan lebih lanjut. Salah satu contoh implementasinya adalah dalam pengendalian motor stepper, motor dengan ketelitian sudut putar yang disesuaikan dengan kondisi masukannya.

1. Pendahuluan

Semakin berkembangnya teknologi digital sebenarnya bukan tanpa alasan. Pada dasarnya teknologi ini menyederhanakan apa yang sebenarnya ada di dunia ini. Misalkan untuk sebuah kurva linear, dengan teknologi digital kurva itu akan di *sampling* sedemikian rupa menjadi beberapa level sehingga tidak perlu memperhatikan setiap titik pada kurva tersebut, melainkan hanya beberapa level sesuai keinginan perancang. Semakin banyak level *sampling* ini, akan semakin presisi dan semakin mendekati kenyataan.

Sistem digital semakin berkembang dengan adanya teknologi biner. Yaitu metode analisa yang hanya mengandalkan bit-bit atau besaran dengan dua kondisi yaitu '1' dan '0'. Semakin besar analisa yang akan dilakukan, bit yang digunakan juga akan semakin banyak. Namun konversi perhitungan dari desimal ke biner ini memungkinkan implementasi yang lebih mudah terutama di bidang elektronika dengan adanya komponen-komponen digital yang bekerja dengan sistem biner pula. Masukan dideteksi sebagai kondisi rendah atau tinggi, dan memberikan keluaran kondisi rendah atau tinggi pula.

Teknologi digital semakin berkembang dengan diciptakannya gerbang-gerbang logika mulai dari OR, AND, XOR, NOT, dan lain-lain. Gerbang-gerbang tadi mengolah masukan biner dan menghasilkan keluaran biner pula. Gerbang itu dapat disusun

sedemikian rupa untuk menghasilkan rangkaian digital yang kita inginkan. Keberadaan IC digital yang telah membanjiri pasaran menjadikannya pilihan tersendiri bagi konsumen yang menginginkan pengendalian sistem dengan perancangan yang mudah dan praktis dibandingkan harus menghitung besaran-besaran dalam perancangan rangkaian elektronika dengan sistem analog.

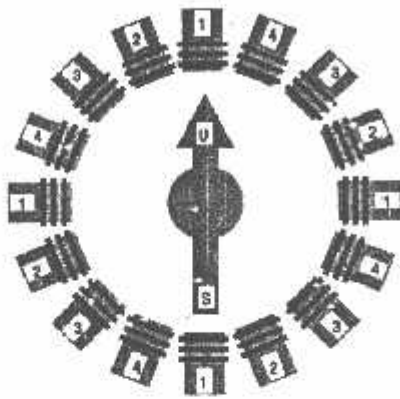
Namun kemudahan ini harus dibayar dengan ketepatan yang relatif lebih rendah dari sistem ini. Untuk kebutuhan keluaran analog, diperlukan konverter digital ke analog yang memiliki skala konversi tertentu. Skala ini tidak akan seteliti bila kita menggunakan sistem analog. Namun untuk kebutuhan pengendalian yang hanya butuh kondisi aktif maupun tidak, rangkaian digital dapat diandalkan.

2. Motor Stepper

Motor stepper adalah perangkat pengendali yang mengkonversi bit-bit inputnya menjadi posisi rotor. Sistem kerjanya adalah motor ini memiliki beberapa pin masukan yang menjadi kutub-kutub magnet di dalam motor. Bila salah satu pin lebih tinggi dari pin seberang, pin itu akan mengaktifkan kutub di dalam motor sebagai kutub utara, dan kutub seberang akan menjadi kutub selatan. Dengan adanya dua kutub ini, rotor di dalam motor stepper yang memiliki kutub permanen akan mengarah sesuai masukan. Kutub utara rotor akan

mengarah ke kutub selatan stator, dan kutub selatan rotor akan mengarah ke kutub utara stator.

Untuk lebih jelasnya dapat dilihat pada gambar berikut.



Gambar 1 skema motor stepper

Gambar diatas menampilkan penampang motor stepper dengan empat koil. Setiap koil memiliki empat kondisi kutub. Bila kondisi satu yang aktif, posisi rotor akan nampak seperti diatas. Dan bila kondisi bergeser ke dua, rotor akan berputar ke kiri dengan sudut putar sesuai jarak kondisi satu dan dua. Namun bila setelah kondisi satu kondisi empat yang aktif, rotor akan menuju ke koil dengan pin empat paling dekat dengan pin satu kondisi sebelumnya. Hal ini menyebabkan rotor berputar ke kanan dan seterusnya. Ketelitian sudut putar pada motor stepper sebanding dengan banyaknya koil dan kondisi masukannya. Pada kondisi diatas, stepper dengan empat koil dan empat kondisi kutub dengan metode full step akan mampu menghadap ke 16 sudut berbeda.

Ada dua tipe pengendalian dengan metode full step yaitu dengan pembangkitan tunggal dan pembangkitan ganda.

Untuk tipe pembangkitan tunggal dapat dilihat lebih jelas pada tabel berikut.

Tabel 1 karakteristik stepper pembangkitan tunggal

Step	1	2	3	4
Koil 1	on	off	off	Off
Koil 2	off	on	off	Off
Koil 3	off	off	on	Off
Koil 4	off	off	off	On

Pada stepper diatas hanya ada satu kondisi yang aktif. Misal koil satu aktif dan lainnya mati, maka rotor akan menghadap ke kutub satu. Dan bila koil dua aktif, dan lainnya mati, rotor akan menghadap ke kutub dua dan seterusnya.

Sedangkan untuk pembangkitan ganda dapat dilihat pada tabel berikut.

Tabel 2 karakteristik stepper pembangkitan ganda

Step	1	2	3	4
Koil 1	on	on	off	off
Koil 2	off	on	on	off
Koil 3	off	off	on	on
Koil 4	on	off	off	on

Pada gambar diatas, dua koil aktif bersamaan, dan dua lainnya mati. Hal ini akan menyebabkan rotor menghadap diantara kutub yang aktif. Misalkan untuk kondisi koil satu dan empat aktif, rotor akan menghadap ke titik diantara kutub satu dan empat. Lalu untuk kondisi koil satu dan dua aktif, rotor akan menghadap ke titik diantara kutub satu dan dua, dan seterusnya.

Dengan sedikit pengembangan ditemukanlah metode half step yang mengkombinasikan kedua tipe pengendalian diatas untuk mendapatkan ketelitian dua kali lipat dari metode full step. Metode half step diimplementasikan dengan menggunakan pembangkitan tunggal dan ganda secara bergantian, sehingga sudut putar semakin kecil, putaran semakin halus, dan arah rotor

semakin banyak. Dengan demikian ketelitian motor meningkat menjadi dua kalinya.

Metode ini menyusun masukan sedemikian rupa sehingga setelah rotor menghadap ke salah satu kutub, sebelum menghadap ke kutub lainnya rotor menghadap ke titik diantara kedua kutub tersebut. Keadaan ini dicapai dengan mengganti tipe pembangkitannya. Untuk mendapatkan sudut putar ini, input diatur sedemikian rupa seperti tabel dibawah.

Tabel 3 karakteristik masukan motor stepper metode half step

Step	D3	D2	D1	D0	Decimal
1	0	0	0	1	01
2	0	0	1	1	03
3	0	0	1	0	02
4	0	1	1	0	06
5	0	1	0	0	04
6	1	1	0	0	12
7	1	0	0	0	08
8	1	0	0	1	09

Dari tabel diatas dapat diilustrasikan yaitu pada step pertama pin D0 aktif, yang menyebabkan rotor menghadap ke kutub pertama, lalu step kedua, pin D0 dan D1 aktif, yang menyebabkan rotor menghadap ke titik diantara kutub pertama dan kutub kedua. Baru pada step ketiga pin D1 aktif dan lainnya mati, yang menyebabkan rotor menghadap ke kutub kedua. Dan seterusnya.

3. Rangkaian Pengendali

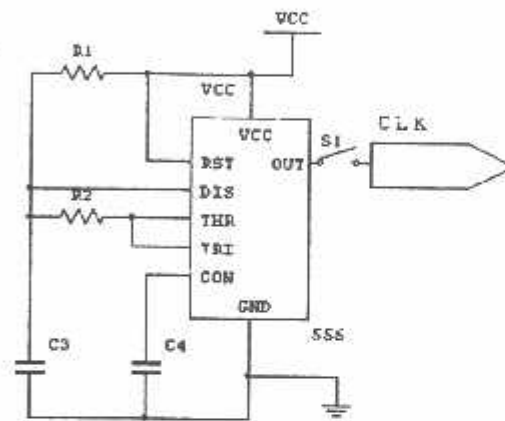
Dengan karakteristik input seperti pada tabel 3, penulis mencoba merancang rangkaian digital untuk menghasilkan keluaran digital seperti karakteristik input yang dibutuhkan motor stepper.

Penulis membagi rangkaian total menjadi tiga blok yang memiliki karakteristik masing-masing. Ketiga blok rangkaian digital tersebut antara lain:

- Pembangkit sinyal clock
- Penghitung sinyal clock
- Pengkonversi bit keluaran

3.1 Pembangkit sinyal clock

Pembangkitan sinyal clock disini menggunakan rangkaian Astable Multivibrator dengan menggunakan IC 555 sebagai inti rangkaian. Clock ini berfungsi mengatur frekuensi putaran motor stepper. Semakin tinggi frekuensi keluaran dari IC 555, akan semakin cepat pula putaran motor stepper, dan sebaliknya semakin rendah frekuensi keluaran IC 555 akan semakin lambat pula putaran motor stepper. Skema rangkaian dapat dilihat pada gambar berikut.



Gambar 2 skema rangkaian pembangkit sinyal clock

Sebagaimana disebutkan sebelumnya, skema diatas menggunakan IC 555 sebagai komponen utama dengan difungsikan sebagai Astable Multivibrator. Karakteristik keluaran dari rangkaian ini adalah gelombang persegi kontinyu dengan frekuensi dan duty cycle tertentu.

Besarnya frekuensi dipengaruhi oleh R1, R2, dan C3, untuk mengubah frekuensi dapat menggantinya dengan komponen variabel dengan besaran tertentu sehingga dapat diubah-ubah sewaktu-waktu sesuai keinginan kita.

Sedangkan saklar diatas digunakan untuk menyambung dan memutuskan hubungan ke rangkaian berikutnya. Ini

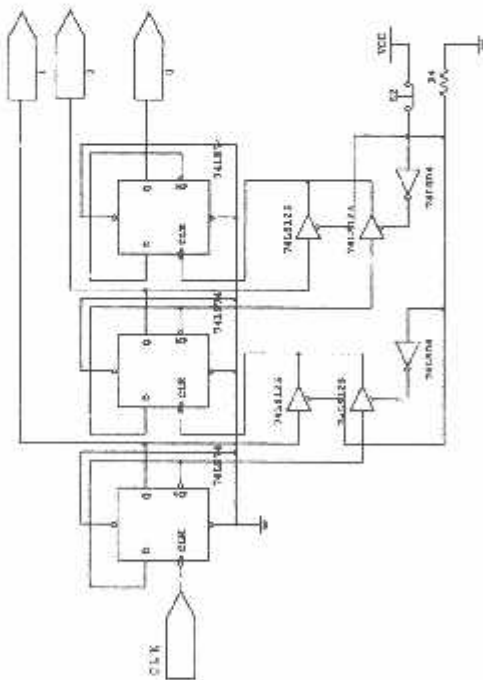
dipasang bila pada suatu saat ingin menghentikan putaran motor stepper, cukup membuka koneksi dari clock sehingga proses terhenti.

3.2 Penghitung sinyal clock

Sinyal clock yang berupa gelombang persegi kontinu difungsikan sebagai input *signal counter* atau penghitung sinyal. Rangkaian ini menggunakan D Flip-Flop sebagai IC utama.

Dari data karakteristik input motor stepper diatas, didapatkan delapan kondisi berbeda. Untuk menciptakannya, diperlukan delapan tingkat penghitungan sinyal, dan setelah mencapai tingkat ke delapan kembali ke posisi awal. Delapan kondisi yang berbeda ini dapat direpresentasikan dengan tiga bit data yang memiliki kombinasi bit sebanyak delapan atau $2^3 = 8$.

Skema rangkaiannya adalah sebagai berikut.



Gambar 3 skema rangkaian penghitung sinyal clock.

Sebagaimana gambar diatas, sinyal clock dari blok rangkaian sebelumnya diinputkan ke D Flip-Flop pertama sebagai sumber clock. Dengan set dan clear dihubungkan ke ground sehingga keluaran pada Q akan memunculkan sinyal clock berupa gelombang persegi dengan frekuensi setengah dari frekuensi clock masukan. Karena kondisi keluaran baru berubah bila clock masukan berubah dari '1' ke '0' dan akan tetap saat clock berubah dari '0' ke '1'.

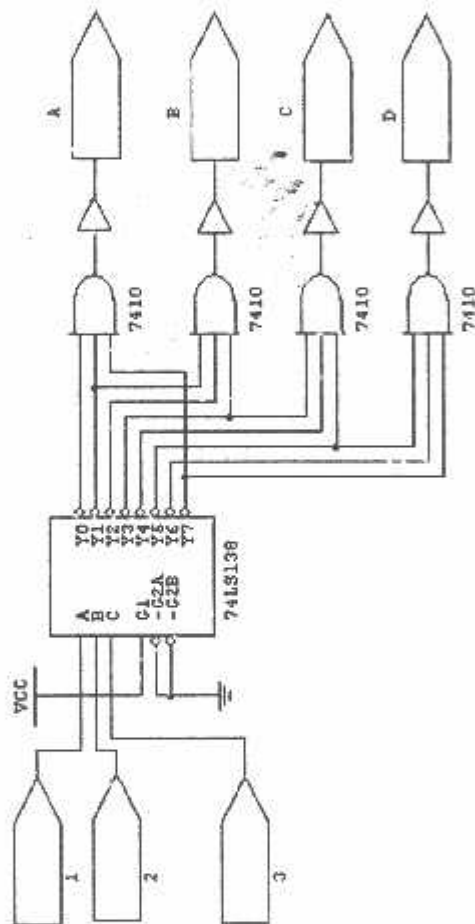
Penggunaan kedua keluaran Flip-Flop ini memungkinkan penghitungan maju maupun mundur. Bila menggunakan keluaran pertama, bit kedua akan berubah bila bit pertama berubah dari '1' ke '0' sehingga dari awalnya '000', '001', '010' dan seterusnya, sehingga Flip-Flop menghitung maju. Sebaliknya bila menggunakan keluaran kedua yang merupakan kebalikan dari keluaran pertama, bit kedua akan berubah bila bit pertama berubah dari '0' ke '1' bila posisi awal '111' maka selanjutnya '110', '101', '100', dan seterusnya. Sehingga rangkaian ini menghitung mundur. Pengendalian keluaran mana yang mensuplai clock Flip-Flop berikutnya dilakukan oleh *tristate buffer* yang ada di bawah rangkaian. Dua *tristate buffer* dengan input masing-masing keluaran pertama dan kedua Flip-Flop digunakan untuk mensaklar salah satu dari inputnya untuk input clock Flip-Flop berikutnya. Pemilihan input dilakukan dengan pengendali *tristate buffer* salah satunya diberi *inverter*, lalu dihubungkan dengan pengendali yang sama. Disini digunakan push button S2. Kondisi mula-mula akan low atau '0' karena arus dibuang ke *ground* yang terlebih dahulu dilewatkan pada R4 untuk mengurangi arusnya. Bila *push button* ditekan, kondisi akan berubah menjadi '1' dengan suplai dari VCC.

Dengan demikian bila *push button* tidak ditekan, *buffer* yang aktif adalah bagian output pertama atau 'Q' sehingga

akan menghitung maju, sebaliknya bila *push button* ditekan, penghitung sinyal clock akan menghitung mundur yang direpresentasikan oleh tiga keluarannya.

3.3 Pengkonversi bit

Keluaran dari blok rangkaian sebelumnya masih berupa tiga bit data. Untuk memunculkan delapan kondisi yang diinginkan kita perlukan pengkonversi bit. Dalam hal ini kami gunakan rangkaian berikut.



Gambar 4 Skema rangkaian pengkonversi bit.

Pada rangkaian diatas, bit-bit keluaran mula-mula dimasukkan ke 3 to 8 decoder untuk memunculkan delapan kondisi yang hanya aktif salah satunya pada suatu waktu tertentu. Setelah itu

baru ditentukan keluaran mana yang akan aktif dengan gerbang-gerbang logika.

Tabel 4 karakteristik konverter bit

STEP	INPUT			3 to 8 Pin Aktif (LOW)	OUTPUT			
	MS B		LS B		A	B	C	D
1	0	0	0	Y0	1	0	0	0
2	0	0	1	Y1	1	1	0	0
3	0	1	0	Y2	0	1	0	0
4	0	1	1	Y3	0	1	1	0
5	1	0	0	Y4	0	0	1	0
6	1	0	1	Y5	0	0	1	1
7	1	1	0	Y6	0	0	0	1
8	1	1	1	Y7	1	0	0	1

Dari karakteristik keluaran diatas, pin output A akan aktif (HI) pada saat Y0, Y1, dan Y7 aktif (LOW). Maka untuk mengaktifkannya digunakanlah gerbang AND tiga input dari Y0, Y1, dan Y7 sehingga pin A akan aktif pada salah satu kondisi tersebut. Begitu pula pin B, dengan gerbang AND menggunakan masukan dari Y1, Y2, Y3, pin B akan aktif (HI) bila salah satu kondisi tersebut terpenuhi aktif (LOW).

Penggunaan *buffer* sebelum keluaran bertujuan untuk mencegah arus balik ke rangkaian diatas. Driver motor stepper menggunakan arus yang cukup besar meski tegangannya hanya 5 volt. Bila tidak diberi *buffer*, dikhawatirkan arus yang cukup besar ini dapat merusak rangkaian sebelumnya, dan mengakibatkan proses pengendalian terganggu.

Keluaran dari sistem ini berupa empat pin yang siap digunakan sebagai input untuk pengendalian motor stepper. Perlu diingat keluaran sistem ini memiliki arus cukup rendah, sehingga sebelum digunakan untuk mengendalikan motor stepper perlu dilewatkan pada rangkaian driver sebagai penguat arus, atau dapat juga digunakan untuk mensaklar relay sebagai drivernya.

4. Kesimpulan dan Saran

Sistem digital berkembang begitu pesat didukung kemudahan dalam analisa dan perancangannya. Misalnya pada analisa dengan sistem biner seperti contoh diatas, kondisi diwakili dengan bit-bit yang hanya memiliki dua kondisi yaitu HI dan LOW, untuk merepresentasikan delapan kondisi berbeda diperlukan tiga bit data. Bit-bit data tadi dapat dikonversi menjadi berbagai ragam sesuai kondisi yang kita inginkan misalnya untuk pengendalian motor stepper diatas.

Penggunaan komponen variabel pada rangkaian Astable Multivibrator dapat memungkinkan kita untuk mengubah frekuensi kerja sistem sewaktu-waktu bila diinginkan. Namun perlu diperhatikan batasan-batasan besaran yang memungkinkan pada rangkaian.

Penghitungan jumlah sinyal dapat dilakukan dengan menggunakan Flip-Flop. Karakteristik Flip-Flop bila set dan clear diberi kondisi LOW adalah keluaran akan berubah bila input clock berubah dari High ke Low. Hal ini ditunjukkan dengan tanda bulat sebelum clock yang menunjukkan clock pada komponen tersebut aktif Low.

Komponen tristate baik itu buffer maupun inverter dapat difungsikan sebagai saklar digital untuk berbagai kegunaan seperti switching pada rangkaian diatas. Masukan dari tristate aktif Low, sehingga terhubung saat masukan tristate rendah, dan terbuka saat masukan tristate tinggi.

Konversi bit difungsikan untuk mengubah bit-bit keluaran dari sistem

menjadi kondisi yang diinginkan untuk pengendalian sistem. Konversi bit dapat dilakukan dengan menggunakan IC-IC decoder atau gerbang-gerbang logika maupun kombinasi keduanya.

Rangkaian-rangkaian digital pada umumnya bekerja pada level arus yang relatif rendah. Untuk pengendalian sistem yang lebih besar terutama mekanik, diperlukan driver untuk memperkuat arus kerja, dan untuk melindungi rangkaian digital ini dari kerusakan akibat kelebihan arus

DAFTAR PUSTAKA

Sutadi, Dwi. *I/O Bus & Motherboard*. Penerbit Andi. Jogjakarta. 2004

Tocci, Ronald J. *Digital Systems*. Prentice Hall, Inc. New Jersey 1991



HARINDRA WISNU P
(L2F004481).

Dilahirkan di Blora 20 tahun 3 hari yang lalu. Menempuh pendidikan dari sekolah dasar sampai sekolah menengah pertama di Blora dan melanjutkan sekolah menengah atas di Semarang. Dari tahun 2004 sampai saat ini sedang menyelesaikan studi Strata-1 di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang, konsentrasi Informatika dan Komputer.

Semarang, 26 November 2006

The following material is excerpted from:

Parallel Port Complete

Programming, Interfacing, & Using the PC's Parallel Printer Port

by Jan Axelson

ISBN 0-9660839-1-5

copyright 1996 by Jan Axelson

Lakeview Research

2209 Winnebago St.

Madison, WI 53704

USA

Phone: 608-241-5824

Fax: 608-241-5848

Email: jaxelson@lv.com

WWW: <http://www.lv.com>

You may distribute this work (*Parallel Port Complete*, Chapter 1) if you agree to distribute it in full and unchanged and agree to charge no fee for such distribution with the exception of reasonable media charges.

The information, computer programs, schematic diagrams, documentation, and other material in this work are provided "as is," without warranty of any kind, expressed or implied, including without limitation any warranty concerning the accuracy, adequacy, or completeness of the material or the results obtained from using the material. *Neither the publisher nor the author shall be responsible for any claims attributable to errors, omissions, or other inaccuracies in the material in this book and the accompanying diskette. In no event shall the publisher or author be liable for direct, indirect, special, incidental, or consequential damages in connection with, or arising out of, the construction, performance, or other use of the materials contained herein.*

Essentials

A first step in exploring the parallel port is learning how to get the most from a port with your everyday applications and peripherals. Things to know include how to find, configure, and install a port, how and when to use the new bidirectional, EPP, and ECP modes, and how to handle a system with multiple parallel-port peripherals. This chapter presents essential information and tips relating to these topics:

Defining the Port

What is the "parallel port"? In the computer world, a *port* is a set of signal lines that the microprocessor, or CPU, uses to exchange data with other components. Typical uses for ports are communicating with printers, modems, keyboards, and displays, or just about any component or device except system memory. Most computer ports are digital, where each signal, or bit, is 0 or 1. A parallel port transfers multiple bits at once, while a serial port transfers a bit at a time (though it may transfer in both directions at once).

This book is about a specific type of parallel port: the one found on just about every PC, or IBM-compatible personal computer. Along with the RS-232 serial port, the parallel port is a workhorse of PC communications. On newer PCs, you

may find other ports such as SCSI, USB, and FireWire, but the parallel port remains popular because it's capable, flexible, and every PC has one.

The term *PC-compatible*, or *PC* for short, refers to the IBM PC and any of the many, many personal computers derived from it. From another angle, a PC is any computer that can run Microsoft's MS-DOS operating system and whose expansion bus is compatible with the ISA bus in the original IBM PC. The category includes the PC, XT, AT, PS/2, and most computers with 80x86, Pentium, and compatible CPUs. It does not include the Macintosh, Amiga, or IBM mainframes, though these and other computer types may have ports that are similar to the parallel port on the PC.

The original PC's parallel port had eight outputs, five inputs, and four bidirectional lines. These are enough for communicating with many types of peripherals. On many newer PCs, the eight outputs can also serve as inputs, for faster communications with scanners, drives, and other devices that send data to the PC.

The parallel port was designed as a printer port, and many of the original names for the port's signals (*PaperEnd*, *AutoLineFeed*) reflect that use. But these days, you can find all kinds of things besides printers connected to the port. The term *peripheral*, or *peripheral device* is a catch-all category that includes printers, scanners, modems, and other devices that connect to a PC.

Port Types

As the design of the PC evolved, several manufacturers introduced improved versions of the parallel port. The new port types are compatible with the original design, but add new abilities, mainly for increased speed.

Speed is important because as computers and peripherals have gotten faster, the jobs they do have become more complicated, and the amount of information they need to exchange has increased. The original parallel port was plenty fast enough for sending bytes representing ASCII text characters to a dot-matrix or daisy-wheel printer. But modern printers need to receive much more information to print a page with multiple fonts and detailed graphics, often in color. The faster the computer can transmit the information, the faster the printer can begin processing and printing the result.

A fast interface also makes it feasible to use portable, external versions of peripherals that you would otherwise have to install inside the computer. A parallel-port tape or disk drive is easy to move from system to system, and for occasional use, such as making back-ups, you can use one unit for several systems. Because a backup may involve copying hundreds of Megabytes, the interface has to be fast to be worthwhile.

This book covers the new port types in detail, but for now, here is a summary of the available types:

Original (SPP)

The parallel port in the original IBM PC, and any port that emulates the original port's design, is sometimes called the *SPP*, for standard parallel port, even though the original port had no written standard beyond the schematic diagrams and documentation for the IBM PC. Other names used are *AT-type* or *ISA-compatible*.

The port in the original PC was based on an existing Centronics printer interface. However, the PC introduced a few differences, which other systems have continued:

SPPs can transfer eight bits at once to a peripheral, using a protocol similar to that used by the original Centronics interface. The SPP doesn't have a byte-wide input port, but for PC-to-peripheral transfers, SPPs can use a Nibble mode that transfers each byte 4 bits at a time. Nibble mode is slow, but has become popular as a way to use the parallel port for input.

PS/2-type (Simple Bidirectional)

An early improvement to the parallel port was the bidirectional data port introduced on IBM's model PS/2. The bidirectional port enables a peripheral to transfer eight bits at once to a PC. The term *PS/2-type* has come to refer to any parallel port that has a bidirectional data port but doesn't support the EPP or ECP modes described below. Byte mode is an 8-bit data-transfer protocol that PS/2-type ports can use to transfer data from the peripheral to the PC.

EPP

The EPP (enhanced parallel port) was originally developed by chip maker Intel, PC manufacturer Zenith, and Xircom, a maker of parallel-port networking products. As on the PS/2-type port, the data lines are bidirectional. An EPP can read or write a byte of data in one cycle of the ISA expansion bus, or about 1 microsecond, including handshaking, compared to four cycles for an SPP or PS/2-type port. An EPP can switch directions quickly, so it's very efficient when used with disk and tape drives and other devices that transfer data in both directions. An EPP can also emulate an SPP, and some EPPs can emulate a PS/2-type port.

ECP

The ECP (extended capabilities port) was first proposed by Hewlett Packard and Microsoft. Like the EPP, the ECP is bidirectional and can transfer data at ISA-bus speeds. ECPs have buffers and support for DMA (direct memory access) transfers.

and data compression. ECP transfers are useful for printers, scanners, and other peripherals that transfer large blocks of data. An ECP can also emulate an SPP or PS/2-type port, and many ECPs can emulate an EPP as well.

Multi-mode Ports

Many newer ports are multi-mode ports that can emulate some or all of the above types. They often include configuration options that can make all of the port types available, or allow certain modes while locking out the others.

System Resources

The parallel port uses a variety of the computer's resources. Every port uses a range of addresses, though the number and location of addresses varies. Many ports have an assigned IRQ (interrupt request) level, and ECPs may have an assigned DMA channel. The resources assigned to a port can't conflict with those used by other system components, including other parallel ports.

Addressing

The standard parallel port uses three contiguous addresses, usually in one of these ranges:

3BCh, 3BDh, 3BEh
378h, 379h, 37Ah
278h, 279h, 27Ah

The first address in the range is the port's base address, also called the Data register or just the port address. The second address is the port's Status register, and the third is the Control register. (See Appendix C for a review of hexadecimal numbers.)

EPPs and ECPs reserve additional addresses for each port. An EPP adds five registers at *base address + 3* through *base address + 7*, and an ECP adds three registers at *base address + 400h* through *base address + 402h*. For a base address of 378h, the EPP registers are at 37Bh through 37Fh, and the ECP registers are at 778h through 77Fh.

On early PCs, the parallel port had a base address of 3BCh. On newer systems, the parallel port is most often at 378h. But all three addresses are reserved for parallel ports, and if the port's hardware allows it, you can configure a port at any of the addresses. However, you normally can't have an EPP at base address 3BCh, because the added EPP registers at this address may be used by the video display.

IBM's Type 3 PS/2 port also had three additional registers, at base address +3 through base address + 5, and allowed a base address of 1278h or 1378h.

Most often, DOS and Windows refer to the first port in numerical order as *LPT1*, the second, *LPT2*, and the third, *LPT3*. So on bootup, *LPT1* is most often at 378h, but it may be at any of the three addresses. *LPT2*, if it exists, may be at 378h or 278h, and *LPT3* can only be at 278h. Various configuration techniques can change these assignments, however, so not all systems will follow this convention. *LPT* stands for line printer, reflecting the port's original intended use.

If your port's hardware allows it, you can add a port at any unused port address in the system. Not all software will recognize these non-standard ports as *LPT* ports, but you can access them with software that writes directly to the port registers.

Interrupts

Most parallel ports are capable of detecting interrupt signals from a peripheral. The peripheral may use an interrupt to announce that it's ready to receive a byte, or that it has a byte to send. To use interrupts, a parallel port must have an assigned interrupt-request level (*IRQ*).

Conventionally, *LPT1* uses *IRQ7* and *LPT2* uses *IRQ5*. But *IRQ5* is used by many sound cards, and because free *IRQ* levels can be scarce on a system, even *IRQ7* may be reserved by another device. Some ports allow choosing other *IRQ* levels besides these two.

Many printer drivers and many other applications and drivers that access the parallel port don't require parallel-port interrupts. If you select no *IRQ* level for a port, the port will still work in most cases, though sometimes not as efficiently, and you can use the *IRQ* level for something else.

DMA Channels

PCs can use direct memory access (*DMA*) for data transfers at the parallel port. During the *DMA* transfers, the CPU is free to do other things, so *DMA* transfers can result in faster performance overall. In order to use *DMA*, the port must have an assigned *DMA* channel, in the range 0 to 3.

Finding Existing Ports

DOS and Windows include utilities for finding existing ports and examining other system resources. In Windows 95, click on *Control Panel, System, Devices, Ports*, and click on a port to see its assigned address and (optional) *IRQ* level and *DMA*

channel. In Windows 3.1 or DOS, you can use Microsoft's Diagnostic (*msd.exe*) to view ports, assigned IRQ levels, and other system details.

Configuring

The parallel port that comes with a PC will have an assigned address and possibly an IRQ level and DMA channel. Multi-mode ports may also be configured with specific modes enabled. You can change some or all of these assignments to match your needs. If you're adding a new port, you need to configure it, making sure that it doesn't conflict with existing ports and other resources.

Port Options

There is no standard method for configuring a port. Some ports, especially older ones, use jumper blocks or switches to select different options. Others allow configuring in software, using a utility provided on disk. A port on a system motherboard may have configuration options in the system setup screens (the CMOS setup) that you can access on bootup. On ports that meet Microsoft's Plug and Play standard, Windows 95 can automatically assign an available port address and IRQ level to a port.

Check your system or port's documentation for specifics on how to configure a port. Some ports allow a choice of just one or two of the three conventional base addresses. A few allow you to choose any uncommitted address, including non-standard ones. On some boards, the jumpers or switches are labeled, which is extremely handy when you don't have other documentation (or can't find it).

If your port supports ECP transfers, assign it an IRQ level and DMA channel if possible. Most ECP drivers do use these, and if they're not available, the driver will revert to a slower mode.

Multi-mode Ports

Configuring a multi-mode port needs special consideration. A multi-mode port's controller chip supports a variety of modes that emulate different port types. In addition to the configuration options described above, on most multi-mode ports, you also have to select a port type to emulate.

The problem is that there is no single standard for the basic setup on the controller chips, and there are many different chips! Usually the setup involves writing to configuration registers in the chip, but the location and means of accessing the registers varies.

For this reason, every port *should* come with a simple way to configure the port. If the port is on the motherboard, look in the CMOS setup screens that you can access on bootup. Other ports may use jumpers to enable the modes, or have configuration software on disk.

The provided setup routines don't always offer all of the available options or explain the meaning of each option clearly. For example, one CMOS setup I've seen allows only the choice of AT or PS/2-type port. The PS/2 option actually configures the port as an ECP, with the ECP's PS/2 mode selected, but there is no documentation explaining this. The only way to find out what mode is actually selected is to read the chip's configuration registers. And although the port also supports EPP, the CMOS setup includes no way to enable it, so again, accessing the configuration registers is the only option.

If your port is EPP- or ECP-capable but the setup utility doesn't offer these as choices, a last resort is to identify the controller chip, obtain and study its data sheet, and write your own program to configure the port.

The exact terminology and the number of available options can vary, but these are typical configuration options for a multi-mode port:

SPP. Emulates the original port. Also called *AT-type* or *ISA-compatible*.

PS/2, or simple bidirectional. Like an SPP, except that the data port is bidirectional.

EPP. Can do EPP transfers. Also emulates an SPP. Some EPPs can emulate a PS/2-type port.

ECP. Can do ECP transfers. The ECP's internal modes enable the port to emulate an SPP or PS/2-type port. An additional internal mode, *Fast Centronics*, or *Parallel-Port FIFO*, uses the ECP's buffer for faster data transfers with many old-style (SPP) peripherals.

ECP + EPP. An ECP that supports the ECP's internal mode 100, which emulates an EPP. The most flexible port type, because it can emulate all of the others.

Drivers

After setting up the port's hardware, you may need to configure your operating system and applications to use the new port.

For DOS and Windows 3.1 systems, on bootup the operating system looks for ports at the three conventional addresses and assigns each an LPT number.

In Windows 3.1, to assign a printer to an LPT port, click on *Control Panel*, then *Printers*. If the printer mode isn't displayed, click *Add* and follow the prompts.

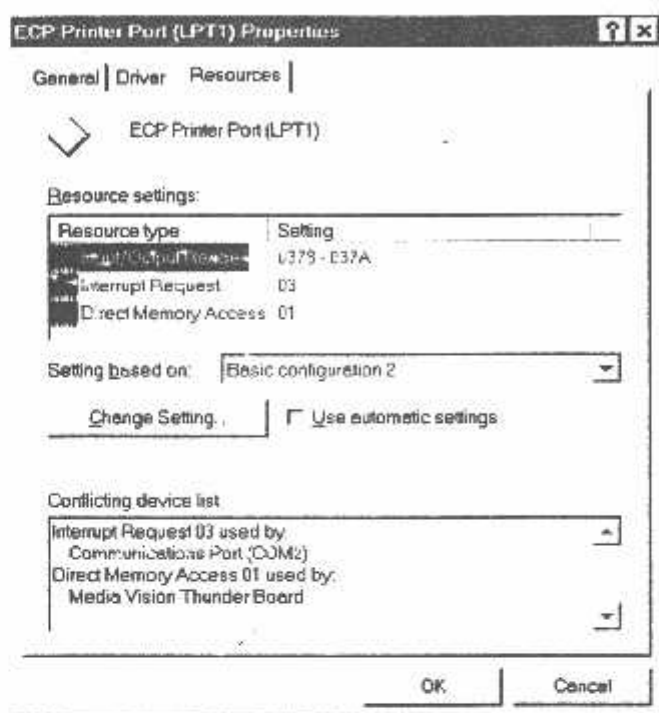


Figure 1-1: In Windows 95, you can select a port configuration in the Device Manager's Resources Window. A message warns if Windows detects any system conflicts with the selected configuration.

Select the desired printer mode, then click *Connect* to view the available ports. Select a port and click *OK*, or *Cancel* to make no changes.

In Windows 95, the Control Panel lists available ports under *System Properties, Device Manager, Ports*. There's also a brief description of the port. *Printer Port* means that Windows treats the port as an ordinary SPP, while *ECP Printer Port* means that Windows will use the abilities of an ECP if possible. To change the driver, select the port, then *Properties, Driver*, and *Show All Drivers*. Select the driver and click *OK*. If an ECP doesn't have an IRQ and DMA channel, the Windows 95 printer driver will use the ECP's Fast Centronics mode, which transfers data faster than an SPP, but not as fast as ECP.

The Device Manager also shows the port's configuration. Select the port, then click *Resources*. Figure 1-1 shows an example. Windows attempts to detect these settings automatically. If the configuration shown doesn't match your hardware setup, de-select the *Use Automatic Settings* check box and select a different configuration. If none matches, you can change a setting by double-clicking on the

resource type and entering a new value. Windows displays a message if it detects any conflicts with the selected settings. To assign a printer to a port, click on *Control Panel, Printers*, and select the printer to assign.

Parallel-port devices that don't use the Windows printer drivers should come with their own configuration utilities. DOS programs generally have their own printer drivers and methods for selecting a port as well.

Adding a Port

Most PCs come with one parallel port. If there's a spare expansion slot, it's easy to add one or two more. Expansion cards with parallel ports are widely available.

Cards with support for bidirectional, EPP, and ECP modes are the best choice unless you're sure that you won't need the new modes, or you want to spend as little as possible. Cards with just an SPP are available for as little as \$15. A card salvaged from an old computer may cost you nothing at all.

You can get more use from a slot by buying a card with more than a parallel port. Because the port circuits are quite simple, many multi-function cards include a parallel port. Some have serial and game ports, while others combine a disk controller or other circuits with the parallel port. On older systems, the parallel port is on an expansion card with the video adapter. These should include a way to disable the video adapter, so you can use the parallel port in any system.

When buying a multi-mode port, it's especially important to be sure the port comes with utilities or documentation that shows you how to configure the port in all of its modes. Some multi-mode ports default to an SPP configuration, where all of the advanced modes are locked out. Before you can use the advanced modes, you have to enable them. Because the configuration methods vary from port to port, you need documentation.

Also, because the configuration procedures and other port details vary from chip to chip, manufacturers of ECP and EPP devices may guarantee compatibility with specific chips, computers, or expansion cards. If you're in the market for a new parallel port or peripheral, it's worth trying to find out if the peripheral supports using EPP or ECP mode with your port.

Port Hardware

The parallel port's hardware includes the back-panel connector and the circuits and cabling between the connector and the system's expansion bus. The PC's microprocessor uses the expansion bus's data, address, and control lines to trans-



Figure 1-2: The photo on the left shows the back panel of an expansion card, with a parallel port's 25-pin female D-sub connector on the left side of the panel. (The other connector is for a video monitor.) The photo on the right shows the 36-pin female Centronics connector used on most printers.

fer information between the parallel port and the CPU, memory, and other system components.

Connectors

The PC's back panel has the connector for plugging in a cable to a printer or other device with a parallel-port interface. Most parallel ports use the 25-contact D-sub connector shown in Figure 1-2. The shell (the enclosure that surrounds the contacts) is roughly in the shape of an upper-case D. Other names for this connector are the subminiature D, DB25, D-shell, or just D connector. The IEEE 1284 standard for the parallel port calls it the IEEE 1284-A connector.

Newer parallel ports may use the new, compact, 36-contact IEEE 1284-C connector described in Chapter 6.

The connector on the computer is female, where the individual contacts are sockets, or receptacles. The cable has a mating male connector, whose contacts are pins, or plugs.

The parallel-port connector is usually the only female 25-pin D-sub on the back panel, so there should be little confusion with other connectors. Some serial ports use a 25-contact D-sub, but with few exceptions, a 25-pin serial D-sub on a PC is male, with the female connector on the cable—the reverse of the parallel-port convention. (Other serial ports use 9-pin D-sub's instead.)

SCSI is another interface whose connector might occasionally be confused with the parallel port's. The SCSI interface used by disk drives, scanners, and other devices usually has a 50-contact connector, but some SCSI devices use a 25-contact D-sub that is identical to the parallel port's connector.

If you're unsure about which is the parallel-port connector, check your system documentation. When all else fails, opening up the enclosure and tracing the cable from the connector to an expansion board may offer clues.

The Circuits Inside

Inside the computer, the parallel-port circuits may be on the motherboard or on a card that plugs into the expansion bus.

The motherboard is the main circuit board that holds the computer's microprocessor chip as well as other circuits and slots for expansion cards. Because just about all computers have a parallel port, the port circuits are often right on the motherboard, freeing the expansion slot for other uses. Notebook and laptop computers don't have expansion slots, so the port circuits in these computers must reside on the system's main circuit board.

The port circuits connect to address, data, and control lines on the expansion bus, and these in turn interface to the microprocessor and other system components.

Cables

Most printer cables have a 25-pin male D-sub connector on one end and a male 36-contact connector on the other. Many refer to the 36-contact connector as the Centronics connector, because it's the same type formerly used on Centronics printers. Other names are parallel-interface connector or just printer connector. IEEE 1284 calls it the 1284-B connector.

Peripherals other than printers may use different connectors and require different cables. Some use a 25-pin D-sub like the one on the PC. A device that uses only a few of the port's signals may use a telephone connector, either a 4-wire RJ11 or an 8-wire RJ45. Newer peripherals may have the 36-contact 1284-C connector.

In any case, because the parallel-port's outputs aren't designed for transmitting over long distances, it's best to keep the cable short: 6 to 10 feet, or 33 feet for an IEEE-1284-compliant cable. Chapter 6 has more on cable choices.

Multiple Uses for One Port

If you have more than one parallel-port peripheral, the easiest solution is to add a port for each. But there may be times when multiple ports aren't an option. In this case, the alternatives are to swap cables as needed, use a switch box, or daisy-chain multiple devices to one port.

If you use only one device at a time and switch only occasionally, it's easy enough to move the cable when you want to use a different device.

For frequent swapping, a more convenient solution is a switch box. A typical manual switch box has three female D-sub connectors. A switch enables you route

Chapter 1

the contacts of one connector to either of the others. To use the switch box to access two peripherals on one port, you'll need a cable with two male D-sub's to connect the PC to the switch box, plus an appropriate cable from the switch box to each peripheral.

You can also use a switch box to enable two PCs to share one printer or other peripheral. This requires two cables with two male D-sub's on each, and one peripheral cable. Switch boxes with many other connector types are also available.

Manual switches are inexpensive, though some printer manufacturers warn that using them may damage the devices they connect to. A safer choice is a switch that uses active electronic circuits to route the signals. Some auto-sensing switches enable you to connect multiple computers to one printer, with first-come, first-served access. When a printer is idle, any computer can access it. When the printer is in use, the switch prevents the other computers from accessing it. However, these switches may not work properly if the peripherals use bidirectional communications, or if the peripheral uses the control or status signals in an unconventional way.

The parallel ports on some newer peripherals support a daisy-chain protocol that allows up to eight devices to connect to a single port. The PC assigns a unique address to each peripheral, which then ignores communications intended for the other devices in the chain. The software drivers for these devices must use the protocol when they access the port. The last device in the chain can be daisy-chain-unaware; it doesn't have to support the protocol. Chapter 11 has more on daisy chains.

Security Keys

Security keys, or dongles, are a form of copy protection that often uses the parallel port. Some software—usually expensive, specialized applications—includes a security key that you must plug into the parallel port in order to run the software. If you don't have the key installed, the software won't run.

The key is a small device with a male D-sub connector on one end and a female D-sub on the other. You plug the key into the parallel-port connector, then plug your regular cable into the security key. When the software runs, it attempts to find and communicate with the key, which contains a code that the software recognizes. The key usually doesn't use any conventional handshaking signals, so it should be able to live in harmony with other devices connected to the port.

The keys do require power, however. If you have a key that draws more than a small amount of current, and if your parallel port has weak outputs, you may have problems in using other devices on the same port as the key.

Alternatives to the Parallel Port

The parallel port is just one of many ways to interface inputs and outputs to a computer. In spite of its many virtues, the parallel port isn't the best solution for every project. These are some of the alternatives:

Serial Interfaces

One large group of parallel-port alternatives is serial interfaces, where data bits travel on a single wire or pair of wires (or in the case of wireless links, a single transmission path.) Both ends of the link require hardware or software to translate between serial and parallel data. There are many types of serial interfaces available for PCs, ranging from the ubiquitous RS-232 port to the newer RS-485, USB, IEEE-1394, and IrDA interfaces.

RS-232

Just about every PC has at least one RS-232 serial port. This interface is especially useful when the PC and the circuits that you want to connect are physically far apart.

As a rule, parallel-port cables should be no longer than 10 to 15 feet, though the IEEE-1284 standard describes an improved interface and cable that can be 10 meters (33 feet). In contrast, RS-232 links can be 80 feet or more, with the exact limit depending on the cable specifications and the speed of data transfers.

RS-232 links are slow, however. Along with each byte, the transmitting device normally adds a start and stop bit. Even at 115,200 bits per second, which is a typical maximum rate for a serial port, the data-transfer rate with one start and stop bit per byte is just 11,520 bytes per second.

RS-485

Another useful serial interface is RS-485, which can use cables as long as 4000 feet and allows up to 32 devices to connect to a single pair of wires. You can add an expansion card that contains an RS-485 port, or add external circuits that convert an existing RS-232 interface to RS-485. Other interfaces similar to RS-232 and RS-485 are RS-422 and RS-423.

Universal Serial Bus

A new option for I/O interfacing is the Universal Serial Bus (USB), a project of a group that includes Intel and Microsoft. A single USB port can have up to 127 devices communicating at either 1.5 Megabits/second or 12 Megabits/second over a 4-wire cable. The USB standard also describes both the hardware interface and software protocols. Newer PCs may have a USB port built-in, but because it's so new, most existing computers can't use it without added hardware and software drivers.

IEEE 1394

The IEEE-1394 high-performance serial bus, also known as Firewire, is another new interface. It allows up to 63 devices to connect to a PC, with transmission rates of up to 400 Megabits per second. The 6-wire cables can be as long as 15 feet, with daisy chains extending to over 200 feet. The interface is especially popular for connecting digital audio and video devices. IEEE-1394 expansion cards are available for PCs.

IrDA

The IrDA (Infrared Data Association) interface allows wireless serial communications over distances of 3 to 6 feet. The link transmits infrared energy at up to 115,200 bits/second. It's intended for convenient (no cables or connectors) transmitting of files between a desktop and laptop computer, or any short-range communications where a cabled interface is inconvenient. Some computers and peripherals now have IrDA interfaces built-in.

Other Parallel Interfaces

SCSI and IEEE-488 are two other parallel interfaces used by some PCs.

SCSI

SCSI (small computer system interface) is a parallel interface that allows up to seven devices to connect to a PC along a single cable, with each device having a unique address. Many computers use SCSI for interfacing to internal or external hard drives, tape back-ups, and CD-ROMs. SCSI interfaces are fast, and the cable can be as long as 19 feet (6 meters). But the parallel-port interface is simpler, cheaper, and much more common.

IEEE 488

The IEEE-488 interface began as Hewlett Packard's GPIB (general-purpose interface bus). It's a parallel interface that enables up to 15 devices to communicate at

speeds of up to 1 Megabit per second. This interface has long been popular for interfacing to lab instruments. Expansion cards with IEEE-488 interfaces are available.

Custom I/O Cards

Many other types of input and output circuits are available on custom expansion cards. An advantage of these is that you're not limited by an existing interface design. The card may contain just about any combination of analog and digital inputs and outputs. In addition, the card may hold timing or clock circuits, function generators, relay drivers, filters, or just about any type of component related to the external circuits. With the standard parallel port, you can add these components externally, but a custom I/O card allows you to place them inside the computer.

To use an expansion card, you of course need an empty expansion slot, which isn't available in portable computers and some desktop systems. And the custom hardware requires custom software.

PC Cards

Finally, instead of using the expansion bus, some I/O cards plug into a PC Card slot, which accepts slim circuit cards about the size of a playing card. An earlier name for these was PCMCIA cards, which stands for *Personal Computer Memory Card International Association*, whose members developed the standard. Many portable computers and some desktop models have PC-Card slots. Popular uses include modems and data acquisition circuits. There are even PC Cards that function as parallel ports. You don't need an internal expansion slot, and you don't have to open up the computer to plug the card in. But again, the standard parallel-port interface is cheaper and more widely available.
