

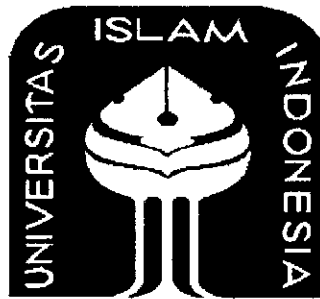
PENGENDALIAN LEVEL KETINGGIAN AIR MENGGUNAKAN KONTROL PID

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana

Pada Jurusan Teknik Elektro Fakultas Teknologi Industri

Universitas Islam Indonesia



Disusun Oleh :

Nama : Syaiful Mansyur

No Mahasiswa : 04524067

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2009

LEMBAR PENGESAHAN PEMBIMBING

PENGENDALIAN LEVEL KETINGGIAN AIR MENGGUNAKAN KONTROL PID



TUGAS AKHIR

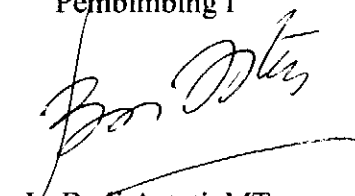
Disusun oleh :

Nama : Syaiful Mansyur

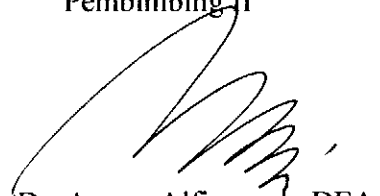
No. Mahasiswa : 04524067

Yogyakarta, April 2009

Pembimbing I


Ir. Budi Astuti, MT

Pembimbing II


Dr. Agung Alfiansyah, DEA

LEMBAR PENGESAHAN PENGUJI
PENGENDALIAN LEVEL KETINGGIAN AIR MUNGUNAKAN
KONTROL PID
TUGAS AKHIR

Oleh :

Nama : Syaiful Mansyur

No. Mahasiswa : 04524067

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Elektro

Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, April 2009

Tim Penguji,

Ir. Budi Astuti, MT

Ketua

Dr. Agung Alfiansyah, DEA

Anggota I

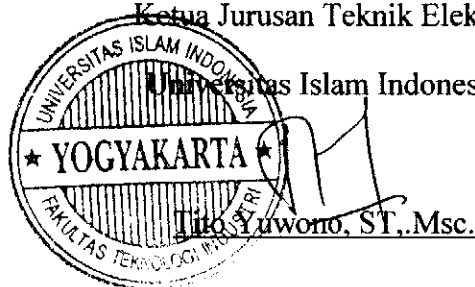
Tito Yuwono, ST., MSc

Anggota II

Mengetahui,

Ketua Jurusan Teknik Elektro

Universitas Islam Indonesia



HALAMAN PERSEMBAHAN

Tugas akhir ini Ananda persembahkan Kepada :

Ayahanda dan Ibunda tercinta

Yang selalu menjadi inspirasi dalam hidupku

*Kakakku, adikku, beserta keluarga besarku yang
selalu memberikan perhatian, semangat, motivasi dan
do'a untukku*

*Tim Robot UII atas semangat dan Ilmu yang tak
terhingga*

Motto

"Sesungguhnya orang yang dirongganya tidak terdapat Al-Qur'an

adalah seperti rumah yang rusak"

(HR. Ahmad, Tarmidzi dan Ad-Darmini)

"Dan jadikanlah sabar dan shalat sebagai penolongmu, dan sesungguhnya yang demikian itu sungguh berat, kecuali bagi orang-orang yang khusyu"

(Q.S. Al-Baqarah: 45)

"Saya hanya tahu satu hal, yaitu saya tidak tahu apapun"

(Socrates.)

"Sebaik-baiknya manusia adalah orang yang banyak manfaatnya (kebaikannya kepada orang lain)"

(H.R. Qodla'ie dari Jabir)

"Sesungguhnya susah kesusahan itu ada kemudahan"

(Q.S Asy Syarh; 5)

KATA PENGANTAR



assalamu'alaikum warahmatullahi wabarakatuh

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan Rahmat dan Hidayah-Nya sehingga penulis dapat menyelesaikan Tugas akhir yang berjudul **“Pengendalian Level Ketinggian Air Menggunakan Kontrol PID ”**. Shalawat dan salam selalu terhaturkan kepada junjungan kita Nabi Muhammad SAW yang menjadi *uswatun hasanah* bagi kita semua hingga akhir zaman.

Penyelesaian Tugas akhir ini merupakan salah satu syarat untuk memperoleh gelar sarjana Teknik Elektro pada Fakultas Teknologi Industri Universitas Islam Indonesia. Dengan selesainya Tugas akhir ini, maka seluruh aktivitas studi pada institusi tercinta ini berakhir dan membuka pintu untuk masuk ke tantangan berikutnya. Proses penyelesaian tugas akhir ini merupakan serangkaian integrasi antara teori serta praktek yang telah didapatkan semasa duduk dibangku kuliah. Ilmu dan amal menjadi poin penting akhir dari penyelesaian tugas akhir ini.

Tugas akhir ini merupakan implementasi dari proses perancangan serta pengujian sistem. Aplikasi utama dari sistem ini banyak dimanfaatkan dalam dunia industri, meskipun skala pengerjaan tugas akhir ini hanya sebatas simulasi. Namun,

begitu pentingnya fungsi dari perangkat ini, menjadikan penulis mencoba menganalisis sekaligus memodelkan sebuah perangkat yang dapat dimanfaatkan sebagai referensi dalam ilmu pengetahuan dan teknologi.

Penulis menyadari sepenuhnya bahwa isi dari tugas akhir ini masih jauh dari sempurna, namun dengan niat yang tulus dan ikhlas, penulis menerima segala macam kritik dan saran yang membangun demi kelangsungan dan kemajuan ilmu pengetahuan dan teknologi.

Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang tak terhingga kepada :

1. Allah SWT, yang dengan izin dan kuasa-Nya masih memberi kesempatan untuk menyelesaikan tugas akhir ini.
2. Baginda Nabi besar Muhammad SAW, atas cahaya Islam.
3. Kedua orangtua ku, papa Mansyur dan mama Nursiah tercinta yang senantiasa memberikan dukungan, semangat, moril, materil dan do'a setiap saat. Terimakasih atas kesabaran kalian.
4. Bapak Tito Yuwono, S.T., M.Sc selaku Ketua Jurusan Teknik Elektro. Fakultas Teknologi Industri
5. Ibu Ir. Budi Astuti, MT. dan bapak DR. Agung Alfiansyah, DEA. selaku dosen pembimbing yang sangat memberikan banyak inspirasi serta motivasi dan ilmu.

6. Kedua kakakku mbak Meny dan mbak Pety dan adikku Kiki, serta seluruh keluarga besarku yang selalu mendoakan dan memberikan dukungan dalam penyusunan tugas akhir ini.
7. *The special one* Erma Oktariani yang telah memberikan perhatian selama ini. Semangat yah kerjain skripsinya.....
8. Seluruh Dosen dan karyawan FTI-Ull yang bersedia dengan sabar membantu dan membagi ilmunya.
9. Mas Agung yang sering membantu dan menemani di Lab. IMEL hingga malam, terimakasih atas komponen elektroniknya.
10. Alvin Sahroni,ST , seorang sahabat yang telah banyak membantu. Selamat menempuh hidup baru kawan. Semoga menjadi keluarga yang sakinah, mawaddah, warohmah dan dianugerahi keturunan yang sholeh dan sholehah. Amin.
11. Teman-teman gen' kapak, Hendra, Tino, Danis, Seto, Pati, Dhani, Sinjo, Dody, semoga kita bias sukses bersama suatu saat.
12. Seluruh teman-teman elektro 04 yang menjadikanku seorang yang bermanfaat buat kalian semua. Terima kasih atas dukungan serta apa yang telah kita ciptakan bersama.
13. Zakki sang *programmer* handal.
14. Kru Laboratorium Instalasi dan Mesin Listrik, Mas Agung dan asisten A'i, Roby, Riski, Mawan.

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN DOSEN PEMBIMBING	ii
LEMBAR DOSEN PENGUJI	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTO	v
KATA PENGANTAR	vi
ABSTRAK	x
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvi
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah	1
1.2 Maksud dan Tujuan	2
1.3 Perumusan Masalah	2
1.4 Batasan Masalah	2
1.5 Langkah Penelitian	3
1.6 Sistematika Penulisan Laporan	3
BAB II STUDI PUSTAKA	
2.1 Pengendali Ketinggian Air menggunakan Kontrol PID	5
2.1.1 Sistem Pengendali	7
2.1.2 Kontrol PID (Proporsional Intergal Derivative)	10

2.1.3 Kontrol P (Kontrol Proporsional)	11
2.1.4 Kontrol I (Kontrol Integral)	13
2.1.5 Kontrol D (Kontrol Derivative)	14
2.1.6 Kombinasi Kontrol P, I, dan D	15
2.2 Potensiometer	16
2.3 ADC (<i>Analog to Digital Converter</i>)	17
2.3.1 ADC 0804	17
2.4 DAC (<i>Digital to Analog Converter</i>)	20
2.4.1 DAC 0808	20
2.5 PPI8255	22
2.6 Port Paralel	24
2.7 Rencana Penelitian	27

BAB III METODOLOGI

3.1 Perancangan Alat	29
3.2 Perancangan Sensor	29
3.3 Perancangan Rangkaian ADC	30
3.4 Perancangan Rangkaian Driver Pompa Air	31
3.5 Perancangan Rangkaian Solenoid Valve	31
3.6 Perancangan Rangkaian DAC	32
3.7 Perancangan Perangkat Lunak	33

BAB IV ANALISA DAN PEMBAHASAN

4.1 Analisis Pada Sisi Perangkat Keras	38
4.1.1 <i>Setting</i> Potensiometer	38
4.1.2 Akuisisi Data Rangkaian ADC	38
4.1.3 Solenoid valve sebagai output	42
4.2 Analisis Perangkat Lunak	43
4.3 Analisis Parameter PID untuk Optimasi Sistem	42
4.3.1 Pengujian Sistem Kontrol PID	43
4.3.1.1 Pengujian Sistem dengan perubahan <i>setting point</i>	46
4.3.1.2 Pengujian Sistem dengan Perubahan Parameter	49
4.3.1.2.1 Perubahan Parameter Kp	49
4.3.1.2.2 Perubahan Parameter Ki	52
4.3.1.2.3 Perubahan Parameter Kd	53
4.3.2 Pengujian Sistem Kontrol PI	54

BAB V PENUTUP

5.1 Kesimpulan	57
5.2 Saran	58

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1	Blog Diagram Utama	5
Gambar 2.2	Kurva karakteristik respon	9
Gambar 2.3	Kontrol Proporsional, P	10
Gambar 2.4	Kontrol Integral, I	12
Gambar 2.5	Kontrol Derivative, D	13
Gambar 2.6	Kontrol PID	14
Gambar 2.7	Bentuk Fisik Potensiometer	15
Gambar 2.8	Konfigurasi pin ADC 0804	16
Gambar 2.9	Konfersi Rangkaian R-2R Ladder	19
Gambar 2.10	DAC 0808	19
Gambar 2.11	Rangkaian Penguat Inverting	21
Gambar 2.12	Nomor Pin Port Paralel	22
Gambar 3.1	Sistem Pengatur	27
Gambar 3.2	Rangkaian Sensor Potensiometer	28
Gambar 3.3	Rangkaian ADC 0804	29
Gambar 3.4	Rangkaian Driver Pompa Air	30
Gambar 3.5	Rangkaian Driver Solenoid valve	31
Gambar 3.6	Rangkaian DAC 0808	31
Gambar 3.7	Perancangan GUI	32

Gambar 3.8	Diagram Alir program kontrol	33
Gambar 3.9	Diagram Alir Program ADC	34
Gambar 4.1	Pengujian potensiometer dengan DAQ NI 6009	39
Gambar 4.2	Tampilan GUI Kontrol PID	40
Gambar 4.4	Respon output sistem dengan $K_p=85$, $K_i=10$, $K_d=3$	46
Gambar 4.5	Grafik ADC setelah di <i>Run</i>	48
Gambar 4.6	Tampilan GUI Kontrol PID dengan $K_p=60$, $K_i=6$, $K_d=40$	50
Gambar 4.7	Respon Output PID dengan $K_p=60$, $K_i=6$, $K_d=10$	51
Gambar 4.8	Respon Output PID dengan $K_p=60$, $K_i=20$, $K_d=10$	52
Gambar 4.8	Respon Output PI dengan $K_p=70$, $K_i=8$	53

DAFTAR TABEL

Tabel 2.1	Tanggapan sistem kontrol PID terhadap perubahan parameter	10
Tabel 2.2	Fungsi Pin Pada Port paralel	22
Tabel 2.3	Alamat Pada Port Paralel	24
Tabel 2.4	Register Port Printer	24
Tabel 4.1	Hasil Pengujian Sensor	37
Tabel 4.2	Hasil Pengujian Pada Rangkaian ADC 0804	38
Tabel 4.3	Hasil Pengujian terhadap <i>setting point</i> PID	44
Tabel 4.4	Hasil Pengujian terhadap parameter K_p	47

ABSTRAK

Tugas akhir ini menjelaskan tentang aplikasi sistem kontrol pada pengaturan level ketinggian air. Metode kontrol yang digunakan dalam sistem ini adalah kontrol proporsional integral derivatif (PID). Penggunaan kontrol PID dapat membuat sistem menjadi dinamik atau stabil dengan memasukkan parameter kontrol yang tepat. Pengujian sistem telah dilakukan untuk mengontrol level air dengan ketinggian antara 0 – 30 cm atau dengan memasukkan *setting point* 0 – 50 %, pengujian ini menggunakan dua tangki air dimana tangki bawah sebagai penampung dan tangki yang bagian atas adalah tangki untuk ketinggian air yang telah diukur. Semua proses kontrol dan akuisisi data dilakukan dengan menggunakan komputer (PC) dan DELPHI7 sebagai tampilan visualnya. Dari hasil pengujian didapatkan bahwa parameter kontrol PID yang tepat untuk pengontrolan sistem ini adalah, $K_p = 80$, $K_i = 10$, $K_d = 3$ dan pada tangki air terbaca 14cm. Pada pengontrolan kontrol PI maka parameter yang tepat di sistem ini adalah, $K_p = 70$, $K_i = 8$ dan pada tangki air terbaca 15cm.

Kata kunci: sistem kontrol, kontrol proporsional integral derivatif (PID), kontrol level air.



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era globalisasi, sektor industri memegang peranan yang sangat penting khususnya di Indonesia. Didalam industri, sangat dibutuhkan sistem kontrol yang baik untuk dapat menunjang proses berjalannya industri tersebut dan untuk meningkatkan efisiensi dalam proses produksi. Karena itu dalam hal ini akan dibuat suatu sistem kontrol yang diharapkan dapat memberikan kontribusi dalam dunia industri.

Perkembangan teknik kontrol sudah merambah dari peralatan industri yang kompleks, perlengkapan militer sampai ke peralatan rumah tangga. Beberapa sistem kontrol yang mudah di jumpai diantaranya adalah pengatur suhu, pengatur kelembaban ruangan, pengendalian level air, pengatur gerakan robot, dan lain sebagainya. Sistem kontrol dibutuhkan untuk memperbaiki tanggapan sistem dinamik agar didapat sinyal keluaran seperti yang diinginkan. Sistem kontrol yang baik adalah yang mempunyai tanggapan yang baik terhadap sinyal masukan yang beragam. Dalam perancangan sistem kontrol ini diperlukan gambaran tanggapan sistem dengan sinyal masukan dan aksi pengontrolan yang meliputi :

- 1) Kestabilan sistem yang dirancang.
- 2) Tanggapan sistem terhadap berbagai jenis aksi pengontrolan.

1.2 Perumusan Masalah

Dari uraian singkat diatas, maka permasalahan yang akan dibahas dalam penulisan ini dapat di jelaskan sebagai berikut :

- 1) Bagaimana mendapatkan sebuah pengendalian yang sempurna dalam sebuah kontrol pengendalian level air.
- 2) Bagaimana mencari fungsi alih dari sistem tersebut.
- 3) Bagaimana merancang sebuah tangki air yang sebenarnya agar dapat menjadi simulasi untuk sistem pengontrolan ini.

1.3 Tujuan dan Manfaat

Adapun tujuan dan manfaat dari penelitian dan perancangan sistem ini adalah:

- a. Dapat membangun suatu sistem pengendalian berbasis kontrol PID yang dapat mengendalikan tinggi rendah air.
- b. Mampu menerapkan sistem pengendalian ini kedalam dunia industri saat ini.
- c. Mendapatkan sistem pengontrolan yang baik untuk sebuah sistem yang belum dinamik.

1.4 Batasan Masalah

Pembahasan dalam penelitian dan perancangan ini dibatasi oleh :

- a. Sensor yang digunakan untuk mengukur ketinggian air adalah potensiometer.

- b. Pengujian sistem ini dilakukan untuk mengontrol level air dengan ketinggian 0 – 50% pada tampilan tangki visual.
- c. Metode pengaturan yang digunakan adalah PID (*Proporsional Integral Derivative*) yang diimplementasikan secara digital.
- d. Metode *tunning* atau penyetelan kontrol PID yang digunakan adalah metode manual (*trial & error*).
- e. Instrumen pengatur untuk semua proses kontrol dan akuisisi data dilakukan dengan menggunakan komputer (PC).
- f. Pengujian instrumen hanya dilakukan pada plant air dan pipa masukan serta keluaran yang memiliki ukuran tertentu.
- g. Pengamatan hanya dilakukan pada gelombang keluaran pada valve yang diatur.
- h. *Setting point* terbatas pada pembacaan ADC antara 0-255 bit.

1.5 Sistematika Penulisan

Sistematika penulisan tugas akhir dengan judul “pengendalian level ketinggian air menggunakan kontrol PID” ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang masalah yang akan diteliti, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, dan sistematika penelitian.

BAB II STUDI PUSTAKA

Bab ini memuat sistem kerja dan analisis dari penelitian dan tugas akhir yang sejenis yang sudah dibuat. Dari analisis ini akan ditentukan pada tugas akhir ini.

BAB III METODOLOGI

Bab ini menjelaskan perancangan *hardware* dan *software* yang digunakan, bagian-bagian yang terlibat dalam sistem beserta cara mengimplementasikan dari setiap bagian tersebut.

BAB IV PENGUJIAN, ANALISIS DAN PEMBAHASAN

Bab ini membahas mengenai hasil-hasil pengujian yang diperoleh dari uji coba perakitan dibandingkan dengan dasar teori sistem.

BAB V PENUTUP

Bab ini memuat kesimpulan dan saran-saran dari proses perancangan yang telah dilakukan.

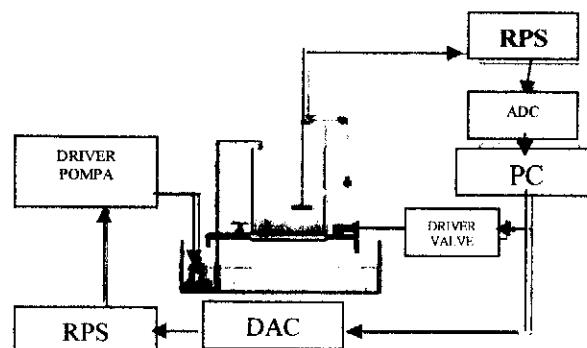
BAB II

STUDI PUSTAKA

2.1 Pengendali Level Ketinggian Air dengan Menggunakan Kontrol PID

Perancangan pengendalian level air telah banyak dilakukan salah satunya pada penelitian yang telah ada yaitu “Pengendali Level Ketinggian Air dengan Menggunakan Kontrol PID”,(Thiang, Yohanes TDS, Andre Mulya, 2004). Makalah ini menjelaskan tentang aplikasi sistem kontrol pada pengaturan level ketinggian air. Metode kontrol yang digunakan dalam sistem ini adalah kontrol proporsional – integral - derivatif (PID). Semua proses kontrol dan akuisisi data dilakukan dengan menggunakan komputer (PC). Pengujian sistem telah dilakukan untuk mengontrol level air dengan ketinggian antara 0 – 30 cm. Pengujian dilakukan dengan beberapa variasi parameter kontrol PID dengan input berupa fungsi step. Dari hasil pengujian didapatkan bahwa parameter sistem kontrol level ketinggian air yang paling tepat adalah $K_p = 10$, $K_i = 0,1$ dan $K_d = 5$.

Sistem kerja dari perancangan alat tersebut adalah sebagai berikut :



Gambar 2.1 Blok diagram Utama

Gambar 2.1 menunjukkan blok diagram sistem yang telah dibuat. *Plant* terdiri dari dua buah bejana atau tangki yang terbuat dari kaca. Tangki yang bawah merupakan tempat penampungan air berukuran 39 x 34 x 35 cm. Tangki yang di atas berfungsi sebagai *plant* yang diatur level ketinggian airnya. Tangki ini berukuran 19 x 14 x 40,5 cm. Sebuah pompa digunakan untuk menaikkan air dari tangki penampungan ke dalam tangki *plant*. Spesifikasi pompa yang digunakan adalah 12 VDC, 3,5 A. Untuk mengatur pembuangan air digunakan sebuah stop kran dan sebuah *solenoid valve*. *Stop* kran digunakan untuk pembuangan air secara manual dan *solenoid valve* untuk pembuangan air yang dikontrol melalui *personal computer* (PC). Sensor yang digunakan untuk mengukur ketinggian air adalah potensiometer. Untuk mengkonversi gerakan naik-turun ketinggian air menjadi gerakan rotasi yang akan memutar potensiometer, digunakan sebuah pelampung dan pemberat yang dirangkai dalam sebuah sistem katrol. Besar potensiometer yang digunakan adalah 5 k Ω . Rangkaian potensiometer yang mengubah ketinggian air menjadi tegangan. Rangkaian ini mengeluarkan tegangan output sebesar 0 – 1,194 volt untuk *range* pengukuran ketinggian air dari 0 – 30 cm.

Dalam perancangan alat sebelumnya ada sebuah kran air manual untuk pembuangan air yang nanti akan digunakan untuk pengujian sistem dengan gangguan, pompa air yang digunakan adalah pompa air DC 12Volt, serta solenoid valve nya mempunyai spesifikasi sistem tegangan AC 220 Volt.

Sedangkan pada tugas akhir yang dibuat sekarang tidak digunakan kran air manual sehingga tidak memperhitungkan pengujian sistem dengan gangguan, spesifikasi pompa air yang digunakan yaitu AC 220 Volt, serta solenoid valve DC 12 Volt. Hal ini dilakukan karena pada tugas akhir ini pengujian dilakukan dengan mengatur output keluaran yang dilakukan oleh solenoid valve, sedangkan input masukannya konstan dari pompa air. Perbedaan lain lagi dari sistem sebelumnya yaitu pada tampilan visual. Pada alat sebelumnya tidak memiliki tampilan visual pada PC atau hanya menampilkan grafik PID. Sedangkan pada alat ini terdapat visual tangki yang *real time* yakni kita dapat memonitor langsung ketinggian air yang ditampilkan di PC secara *real time*. Serta terdapat pula tampilan grafik dari keluaran ADC serta grafik kontrol PID.

Untuk tampilan visualisasi dan simulasi kontrol PID pada tangki air, penulis mengadopsi simulasi pengendali PID yang dikembangkan oleh Andrew R Clayton (<http://www.torrey.net/authorsmore.php?id=6911>)

2.1.1 Sistem Pengendali

Pengendali merupakan komponen sistem yang berfungsi mengolah informasi yang berasal dari sinyal *state* (state sistem, state input, state output), menjadi sinyal kontrol sedemikian rupa sehingga karakteristik dinamik sistem hasil desain memenuhi spesifikasi performansi tertentu. Dalam perencanaan sebuah sistem kendali, hal pertama yang harus dilakukan adalah mendefinisikan struktur sistem tersebut secara tepat. Perencanaan ini biasanya dilakukan agar memenuhi terhadap spesifikasi berikut :

- Ketelitian
- Kecepatan memberi jawaban
- *Settling time* paling lama
- Stabilitas

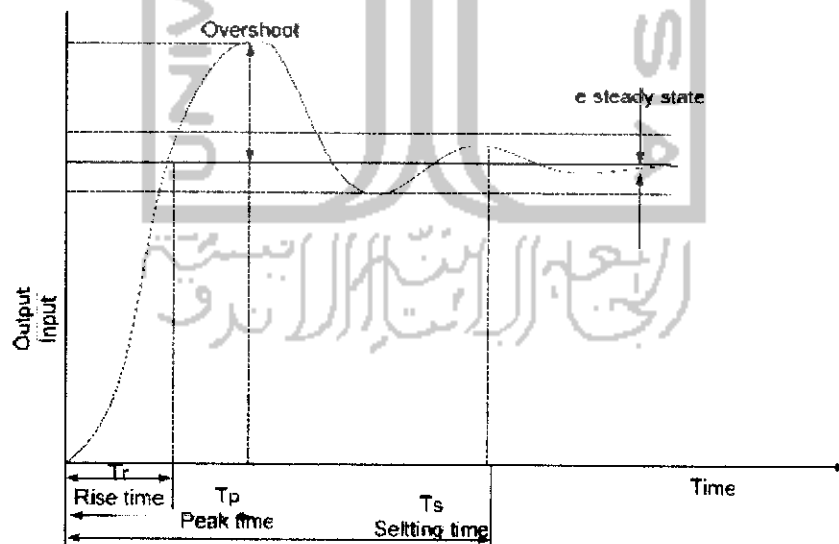
Dalam setiap perancangan pengendali atau spesifikasi kontroler, diperlukan suatu spesifikasi kontrol yang digunakan sebagai acuan perancangan. Pada perancangan kontroler untuk mengetahui level ketinggian air dengan mengatur *valve* keluaran tank, digunakan spesifikasi *performance* untuk respon *transient* sebagai acuan untuk mendapatkan suatu kontroler yang baik. Respons *transient* suatu sistem kontrol secara praktek selalu menunjukkan osilasi teredam sebelum mencapai keadaan mantap (*steady state*).

Menurut Ogata, parameter-parameter yang diperlukan untuk menganalisa karakteristik performansi sistem kontrol sebagai berikut :

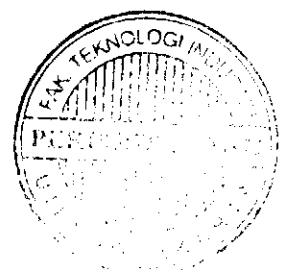
- 1) Waktu tunda (*delay time*), t_d : adalah waktu yang diperlukan oleh respons untuk mencapai setengah nilai akhir untuk waktu yang pertama.
- 2) Waktu naik (*rise time*), t_r : adalah waktu yang diperlukan oleh respon untuk naik dari 10% menjadi 90%, 5% menjadi 95%, atau 0% menjadi 100% dari nilai akhir yang biasa digunakan. Untuk sistem atas redaman waktu naik yang biasa digunakan 10% menjadi 90%.
- 3) Waktu puncak (*peak time*), t_p : adalah waktu yang diperlukan oleh respon untuk mencapai puncak pertama lewatan (*overshoot*).

Maksimum (persen) *overshoot* (*overshoot maximum*), M_p : adalah nilai puncak kurva, respons ukur dari satuan. Besarnya persen lewatan maksimum menunjukkan kestabilan relatif dari sistem.

- 4) Waktu penetapan atau waktu turun (*settling time*), t_s : adalah waktu yang diperlukan untuk merespon kurva agar dapat mencapai dan tetap berada dalam gugus nilai akhir ukuran yang disederhanakan dengan presentase mutlak harga akhirnya (biasanya 2% atau 5%).
- 5) Respon *steady state* diamati mulai saat respon masuk dalam keadaan “*steady state*” sampai waktu tak terbatas.



Gambar 2.2 Kurva karakteristik respon *water tank*



2.1.2 Kontrol PID (Proporsional Integral Derivative)

Didalam suatu sistem kontrol kita mengenal adanya beberapa macam aksi kontrol, diantaranya yaitu aksi kontrol proporsional, aksi kontrol integral dan aksi kontrol derivative. Masing masing aksi pada kontrol ini mempunyai keunggulan-keunggulan tertentu, dimana aksi kontrol proporsional mempunyai keunggulan risetime yang cepat, aksi control integral mempunyai keunggulan untuk memperkecil *error* sistem , dan aksi kontrol derivative mempunyai keunggulan untuk memperkecil *d-error* (error sebelumnya) atau meredam *overshot/undershot*. Untuk itu agar kita dapat menghasilkan output dengan risetime yang tinggi dan error yang kecil kita dapat menggabungkan ketiga aksi kontrol ini menjadi aksi kontrol PID.

PID adalah sebuah model matematika berisi Gain+Integral+Derivative yang tepat digunakan untuk mengurangi gap antara variabel yang diinginkan dan variabel kenyataan dengan melakukan penyetelan pada variabel yang dapat mempengaruhi variabel kenyataan.

PID adalah hasil karya matematis yang cukup jitu untuk melakukan iterasi untuk membuat Error menjadi 0. Apa artinya 0 itu? artinya adalah *Process Variable* = *Set Point* atau dapat dikatakan Variabel kenyataan = Variabel yang diinginkan. Jika keduanya sudah sama maka itulah kestabilan dan tujuan pengontrolan sudah tercapai.

Variabel kenyataan = Tanki Kosong

Variabel yang diinginkan = Level air yang diinginkan

Variabel yang mempengaruhi = Kocoran air masuk ke dalam tanki.

Tanggapan Loop Tertutup	Waktu Naik	<i>Overshoot</i>	Waktu Turun	Kesalahan Keadaan Tunak
Proporsional (K_p)	Memurun	Meningkat	Perubahan Kecil	Memurun
Integral (K_i)	Memurun	Meningkat	Meningkat	Hilang
Derivative (K_d)	Perubahan Kecil	Memurun	Memurun	Perubahan Kecil

Tabel 2.1 Tanggapan sistem kontrol PID terhadap perubahan parameter

Pada awalnya setiap jenis kontrol memiliki persamaan matematis dalam domain waktu. Untuk memudahkan penulisan program maka setiap persamaan matematis pada tiap-tiap kontrol akan diubah dalam bentuk diskrit atau di transformasi laplace kan. Berikut ini penjelasan tentang apa saja keunggulan dan kerugian pada masing-masing jenis kontrol.

2.1.3 Kontrol P (Kontrol Proporsional)

Untuk kontroler dengan aksi kontrol proposional, hubungan antara masukan kontroler $u(t)$ dan sinyal pembangkit $e(t)$ adalah

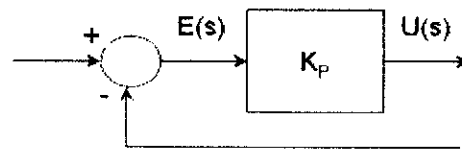
$$U(t) = K_p e(t) \quad (2.1)$$

Atau dalam besaran transformasi laplace

$$\frac{U(s)}{E(s)} = K_p \quad (2.2)$$

Dimana K_p adalah gain proposional

Dengan diagram blok seperti gambar di bawah ini

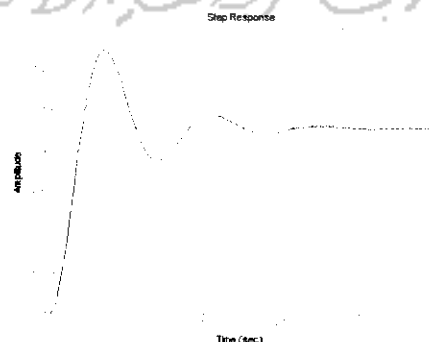


Gambar 2.3. Diagram blok kontroler proposional

Pengaruh kontrol proposional pada sistem yang menggunakannya adalah sebagai berikut :

1. Menambah atau mengurangi kestabilan.
2. Dapat memperbaiki respon transien khususnya *rise time* dan *settling time*.
3. Mengurang error steady state.

Penggunaan kontrol P saja memiliki berbagai keterbatasan karena sifat kontrol yang dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu mencapai konvergensi meskipun error keadaan tenangnya (*steady state error*) relatif besar. Sebagai materi pembelajaran, kontrol P dianggap sangat baik untuk permulaan.



Gambar 2.4 Tanggapan sistem terhadap aksi kontrol proporsional

Penambahan aksi kontrol P mempunyai pengaruh mengurangi waktu naik dan kesalahan keadaan tunak, tetapi konsekuensinya *overshoot* naik cukup besar. Kenaikan *overshoot* ini sebanding dengan kenaikan nilai parameter K_p . Waktu turun juga menunjukkan kecenderungan yang membesar.

2.1.4 Kontrol I (Kontrol Integral)

Untuk pengendali Integral nilai masukan kontroler $u(t)$ diubah pada laju proporsional dari sinyal pembangkit kesalahan $e(t)$. sehingga

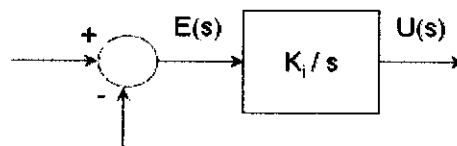
$$\frac{du(t)}{dt} = K_i e(t) \quad (2.3)$$

Atau

$$u(t) = K_i \int_0^t e(t) dt \quad (2.4)$$

dimana K_i adalah konstanta integral. Fungsi alih dari konstanta integral adalah

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.5)$$



Gambar 2.5. Diagram blok kontroler integral

Pengaruh pada sistem yang menggunakan jenis kontrol integral adalah sebagai berikut :

1. Menghilangkan error steady state.
2. Dapat menimbulkan ketidakstabilan pada sistem karena menambah orde sistem

Gambar 2.3 blok diagram merupakan aksi kontrol I yang dapat memperbaiki respon *steady-state* namun pemilihan K_i yang tepat dapat menyebabkan ke tidak stabilan system. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan output berisolasi.

2.1.5 Kontrol D (Kontrol Derivative)

Alat kontrol jenis ini digunakan untuk memperbaiki/mempercepat prestasi respons transient sebuah sistem kontrol. Alat ini selalu disertai oleh tipe P, sedangkan tipe I hanya digunakan apabila diperlukan. Nilai keluaran dari kontrol derivatif dapat dirumuskan sebagai berikut :

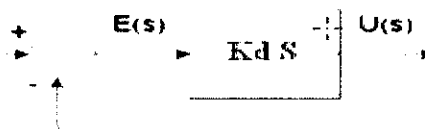
$$U(t) = K_d \frac{de(t)}{dt} \quad (2.6)$$

Dimana K_d adalah konstanta derivatif.

Atau dalam besaran transformasi laplace.

$$\frac{U(s)}{E(s)} = K_d s \quad (2.7)$$

Dengan diagram blok.



Gambar 2.6. Diagram blok kontroler derivative

Diag Pengaruh pada sistem yang menggunakan jenis kontrol integral adalah sebagai berikut :

1. Memperbaiki respon transien, karena memberikan aksi saat ada perubahan *error*.
2. Memberikan efek redaman pada sistem yang berosilasi.

2.1.6 Kombinasi Kontrol P, I, dan D

Kombinasi dari ketiga jenis kontrol ini mempunyai keuntungan dibanding masing-masing kontroler diantaranya :

P : memperbaiki respon transien.

I : menghilangkan error steady state.

D : memberikan efek redaman.

Persamaan dengan ketiga kombinasi ini diberikan oleh.

$$U(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2.6)$$

1. *Trin* Dimana K_p adalah konstanta proposional, T_i waktu integral, T_d waktu turunan. Atau dalam besaran transformasi laplace.

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (2.7)$$

$$\frac{U(s)}{E(s)} = K_p - \frac{K_i}{s} + K_d s \quad (2.8)$$

2.3 ADC (*Analog to Digital Converter*)

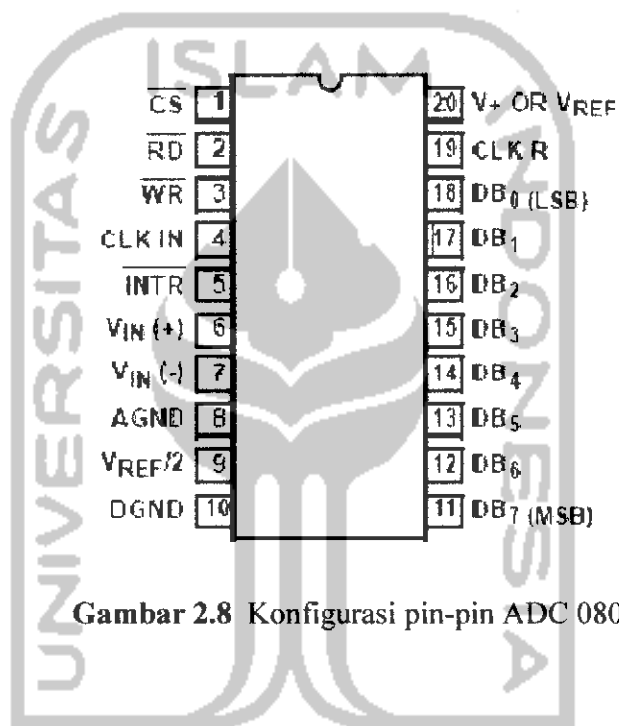
ADC (*Analog to Digital Converter*) adalah suatu piranti elektronik yang berfungsi untuk mengubah sinyal analog ke dalam sinyal digital. Contoh besaran analog misalnya tegangan, suhu, tekanan, dan sebagainya. ADC ini digunakan untuk mengkonversi besaran analog atau tegangan menjadi bit biner agar dapat diakuisisi oleh komputer (PC). Masukan digital ini nanti akan diterjemahkan ke dalam fungsi logika.

Dalam perancangan ini yang akan diukur adalah suatu level ketinggian air dengan menggunakan potensiometer $5K\Omega$ sepuluh kali putar atau lebih dikenal dengan nama *potentio multiturn*. Potensiometer ini menghasilkan tegangan analog yang tidak dapat dibaca langsung oleh PC. Namun dengan kondisi semacam ini bukan berarti PC sama sekali tidak bisa mengukur tegangan analog. Ada metode khusus yang dapat digunakan untuk memecahkan masalah ini, yaitu ADC. Besaran analog yang dihasilkan oleh potensiometer ini akan diubah kedalam besaran digital yang mampu digunakan untuk fungsi logika dan pemrograman.

2.3.1 ADC 0804

ADC 0804 terdiri dari jaringan rangkaian 256R. Salah satu tipe ADC jenis pendekatan beruntun yang sudah dikemas dan terintegrasi dalam satu chip IC. Dalam hal ini kecepatan konversi IC mendekati peringkat kedua setelah ADC tipe flash conversion yang harganya sangat mahal. Secara garis besar ADC ini terdiri dari tiga

bagian yaitu register pendekatan beruntun atau *Successive Approximation Register* (SAR), *Digital to Analog Converter* (DAC), dan komparator. SAR terdiri dari tiga sub bagian yaitu trial register, output register, dan control unit.



Gambar 2.8 Konfigurasi pin-pin ADC 0804

Penjelasan pin-pin ADC 0804 seperti pada gambar 2.6 :

- Pin CS (Chip Select) : merupakan pin untuk mengaktifkan ADC. Jika pin ini aktif (diberi logika rendah) maka ADC siap bekerja baik untuk operasi baca (WR aktif) maupun operasi aktif (RD aktif).
- Pin RD : Digunakan untuk operasi baca. Pin ini bersifat aktif rendah (akan aktif jika diberi logika aktif rendah). Jika pin ini aktif dan disertai aktifnya pin CS maka ADC telah siap dibaca.

- c. Pin WR : Digunakan untuk operasi tulis dan bersifat aktif rendah. Jika pin ini aktif disertai aktifnya pin CS maka ADC siap untuk membaca tegangan input.
- d. Pin INTR : (sebagai penyedia fasilitas interupsi). Pin ini memungkinkan adanya interupsi ke mikroprosesor jika buffer output ADC telah terisi data. Hal ini sebagai pemberitahuan ke prosesor bahwa ADC telah siap dibaca datanya.
- e. Vin + : pin ini sebagai setelan masukan analog maksimum yang diinginkan.
- f. Vin - : Pin ini sebagai setelan masukan analog minimum yang diinginkan.
- g. Pin CLK IN (masukan sinyal clock) : merupakan tempat dimana dapat diberikan clock masukan. Bersama dengan CLK R dapat digunakan sebagai clock pembangkit internal ADC.
- h. GND analog : Merupakan referensi ground bagi ladder-ladder logika analog.
- i. Vref / 2 : Pin ini diset sebagai nilai tengah dari VCC yang diinginkan.
- j. GND digital : Merupakan referensi ground dari generator clock yang nantinya akan mengendalikan shift register.
- k. Pin output data digital (pin no 11 - 18) : Pin-pin ini adalah 8 saluran output digital yang akan diberikan ke mikroprosesor.
- l. Pin CLK R : merupakan penyusun perangkat pembangkit clock internal bersama dengan CLK IN.
- m. Vcc / Vref (pin 20) : merupakan tempat catu daya.



Clock ADC dipakai clock internal (karena tersedia dalam ADC ini), yaitu dengan memberikan untai R-C pada masukan CLK In dan CLK R. Dengan ketentuan nilai R tidak boleh terlalu besar karena akan mengganggu operasi konversi

2.4 DAC (*Digital to Analog Converter*)

ADC (*Digital to Analog Converter*) secara umum merupakan suatu penerjemah informasi berbentuk digital ke dalam informasi berbentuk ekuivalennya. Sebagai contoh sebuah data komputer digital diubah ke dalam bentuk analog untuk menggerakkan motor DC pada printer. Peralatan pengubah sinyal digital ke sinyal analog disebut juga pendekode atau decoder. DAC yang digunakan disini yaitu DAC menggunakan jaringan tangga resistor R-2R.

2.4.1 DAC 0808

IC DAC 0808 adalah jenis D/A Conversion yang mempunyai 8 bit masukan dan dibangun dengan metode konfersi rangkaian R-2R ladder. Masukan dari data biner IC ini sesuai dengan gerbang (gate) TTL, sehingga keluaran dari port standart melalui PIO, IC DAC 0808 ini mempunyai konfigurasi pena dan blok diagram seperti pada Gambar 2.7 dan 2.8 sebagai berikut :

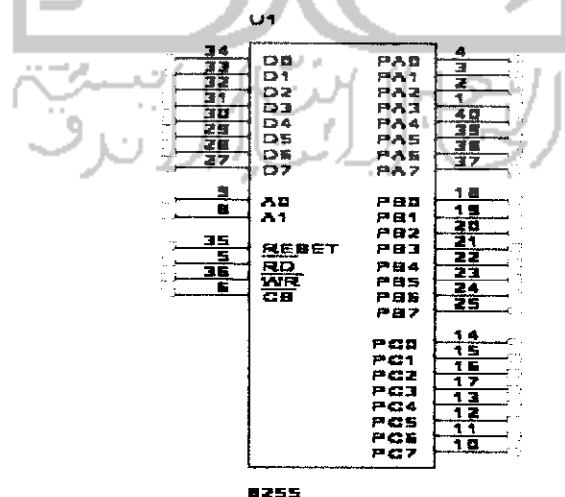
tegangan keluaran ditentukan oleh $V_{ref} (+)$ dan $V_{ref} (-)$, dengan persamaan outputnya adalah :

$$V_{out} = V_{ref} \times (A1/2 + A2/4 + A3/8 + \dots + A8/256) \quad (2.9)$$

Kisaran mengenai keluaran DAC 0808 dapat diatur sesuai dengan catu daya tegangan referensi yang telah diberikan. Pedoman keluaran minimum juga dapat kita atur atau diset dengan membandingkan tegangan pada referensi.

Ditinjau dari segi fungsi DAC itu sendiri merupakan komponen yang sekedar merubah data biner menjadi tegangan analog, namun jika dilihat dari teknologi pemrograman DAC dapat berkembang menjadi sebuah peralatan yang canggih dimana dalam pemasangannya tidak terlepas dari ADC (Analog Digital Converter) sebagai peralatan masukan.

2.5 PPI (Programable Peripheral Interface) 8255



Gambar 2.11 Konfigurasi Pin PPI 8255

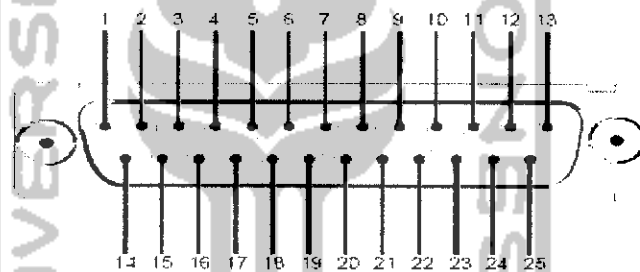
PPI 8255 (Programmable Peripheral Interface 8255) merupakan komponen interface yang dapat digunakan untuk keperluan input dan output. IC ini mempunyai 24 bit I/O yang dikelompokkan menjadi 3 buah port yang masing-masing mempunyai 8 bit dengan nama masing-masing PORT A, PORT B dan PORT C. D0 sampai dengan D7 merupakan jalur data yang digunakan untuk mentransfer data, memprogram 8255 dan membaca status 8255. Proses baca atau menulis pada PPI 8255 dikendalikan melalui pin WR (Write) dan RD (Read). Mode input atau output PPI8255 diatur dengan mengisi data pada Register Control Word PPI 8255. Alamat PORT A, PORT B, PORT C dan Control Word dibedakan berdasarkan logika di alamat A0 dan A1. Perintah untuk membaca atau menulis ke port dilakukan dengan memberikan logika 0 atau 1 pada pin WR dan RD. PPI 8255 dapat difungsikan atau tidak dengan cara memberi logika 0 atau 1 pada pin CS (Chip Select).

Tabel 2.2 Dasar operasi PPI 8255

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Operasi Input (READ)
0	0	0	1	0	Port A → Bus Data
0	1	0	1	0	Port B → Bus Data
1	0	0	1	0	Port C → Bus Data
					Operasi Output (WRITE)
0	0	1	0	0	Bus Data → Port A
0	1	1	0	0	Bus Data → Port B
1	0	1	0	0	Bus Data → Port C
1	1	1	0	0	Bus Data → Control Word
					Disable Function
X	X	X	X	1	Bus Data → 3-state
1	1	0	1	0	Illegal Condition
X	X	1	1	0	Bus Data → 3-state

2.6 Port Paralel

Port paralel banyak digunakan dalam berbagai macam aplikasi antarmuka. Port ini mengizinkan untuk menggunakan 8 bit masukan atau 12 bit keluaran pada saat bersamaan, dengan hanya membutuhkan rangkaian eksternal sederhana untuk melakukan suatu tugas tertentu. Port ini terdiri dari 4 jalur kontrol, 5 jalur status, dan 8 jalur data. Biasanya dapat dijumpai sebagai port *printer*, dalam bentuk konektor DB-25 betina (*female*).



Gambar 2.12 Nomor Pin Port Paralel

Seiring dengan perkembangan desain komputer, beberapa perusahaan memperkenalkan port printer dengan versi yang lebih baik dengan faktor kecepatan dan jumlah informasi yang dipindahkan sebagai landasan, berikut jenis-jenis port yang tersedia :

a) Original (SPP)

SPP (*Standart Parallel Port*) menggunakan mode *Nibble* yang mentransfer 4 bit setiap saatnya.

b) Tipe PS/2 (dua arah sederhana)

PS/2 adalah port data dua arah yang tidak mendukung mode EPP dan ECP. Port dua arah membolehkan peripheral mentransfer 8 bit setiap saat ke komputer.

c) EPP

EPP (*Enhanced Parallel Port*) bersifat dua arah dan dapat memutar arah transfer secara langsung sehingga sangat efisien ketika digunakan piranti yang mentransfer data dalam dua arah.

d) ECP

ECP (*Extended Capabilities Port*) bersifat dua arah dan dapat mentransfer data pada kecepatan bus ISA. ECP mempunyai *buffer* dan mendukung transfer data secara DMA (*Direct Memory Access*) serta kompresi data.

e) BiDir

BiDir (*Bi Directional*) bersifat dua arah dan dapat mengubah sifat tersebut dengan cara mengubah bit pin kontrol (umumnya bit-5)

Tabel 2.2 Tabel fungsi Pin pada Port Paralel

DB 25	Centronics	SPP Signal	Arah In/Out	Simbol	Port
1	1	Strobe	In/Out	-PC0	Kontrol
2	2	Data 0	Out	+D0	Data
3	3	Data 1	Out	+D1	Data
4	4	Data 2	Out	+D2	Data
5	5	Data 3	Out	+D3	Data
6	6	Data 4	Out	+D4	Data
7	7	Data 5	Out	+D5	Data
8	8	Data 6	Out	+D6	Data
9	9	Data 7	Out	+D7	Data
10	10	ACK	In	+PS6	Status
11	11	Busy	In	-PS7	Status
12	12	Paper-Out/Paper-End	In	+PS5	Status
13	13	Select	In	+PS4	Status
14	14	Autofeed	In/Out	-PC1	Kontrol
15	32	Error	In	+PS3	Status
16	31	Initialize	In/Out	+PC2	Kontrol
17	36	Select-In	In/Out	-PC3	Kontrol
18-25	19-30	Ground	-	GND	-

Tanda + di depan simbol sinyal menunjukkan sinyal tersebut aktif tinggi, sedangkan tanda - (minus) menunjukkan bahwa sinyal aktif rendah.

Tabel 2.3 Alamat pada Port Paralel

Alamat	Keterangan
3BC – 3BF	Digunakan untuk port paralel yang terpadu dengan kartu-kartu video, tidak mendukung alamat ECP
378 – 37F	Biasa digunakan untuk LPT 1
278 – 27F	Biasa digunakan untuk LPT 2

Port kontrol 5 bit digunakan untuk mengatur arah pada mode *bidirectional*, dengan keterangan sebagai berikut :

1. Bit 5 diisi bernilai 0 untuk operasi penulisan data (keluaran ini merupakan nilai *default* saat komputer dinyalakan)
2. Bit 5 bernilai 1 untuk operasi pembacaan data (ke dalam)

Alamat register ditentukan berdasarkan alamat dasarnya, sebagai contoh jika port printer yang digunakan adalah LPT1 yang alamat dasarnya adalah \$378, maka data port, port status, dan port kontrol berturut-turut adalah $\$378+0 = \378 , $\$378+1 = \379 , $\$378+2 = \$37A$, seperti pada tabel 2.4 berikut :

Tabel 2.4 Register Port Printer

Nama register	Alamat
Register data	Dasar + 0
Register status	Dasar + 1
Register kontrol	Dasar + 2

2.7 Rencana Penelitian

Rencana penelitian dari perancangan “Pengendalian Level Ketinggian Air Menggunakan Kontrol PID” adalah dengan merancang perangkat keras (*hardware*) dan perangkat lunak (*software*). Keduanya digabungkan untuk mendapatkan tujuan dari sistem perancangan ini.

Perancangan perangkat keras dari alat ini meliputi rangkaian catu daya untuk memberi catuan pada keseluruhan rangkaian, yaitu dengan menggunakan transformator CT step down, dioda sebagai penyearah, *ELCO* (kondensator elektronik) sebagai filter dan IC regulator 7805, 7812 untuk output tegangan plus (+), serta IC 7915 untuk menghasilkan output tegangan minus (-) pada regulator. Rangkaian Op-amp sebagai penguat non inverting yang terdiri dari IC Op-Amp LM 351 dan beberapa resistor. Rangkaian ADC dengan menggunakan IC ADC0804, resistor dan kapasitor. Dimana kapasitor ini dirangkai sebagai mode *free running* yang akan mengkonversi tegangan analog menjadi data digital secara terus menerus. Rangkaian DAC 0808 yang merupakan DAC 8 bit digunakan untuk mengubah data digital menjadi tegangan analog yang digunakan untuk mengendalikan dan mengatur *solenoid valve*.

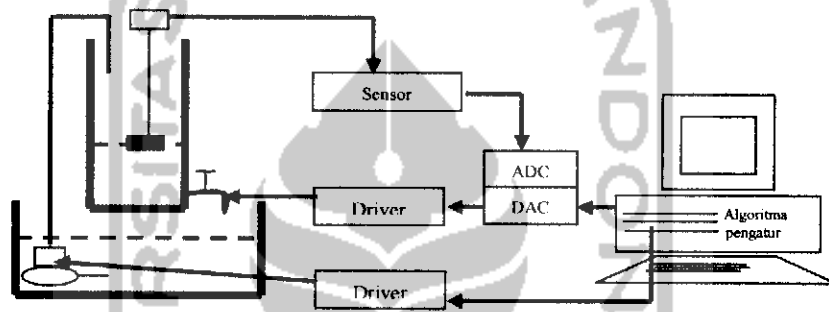
Perancangan perangkat lunak menggunakan bahasa pemrograman Pascal dengan Software Delphi 7 sebagai pemrogramannya. Perancangan program berupa inisialisasi program, pendeklarasian variabel, perancangan *procedure*, perancangan visualisasi, serta pembacaan data dari port paralel.

Parameter yang akan diukur adalah perubahan *setting point*, perubahan parameter K_p , K_i , dan K_d serta penggunaan kontrol PI pada sistem.

BAB III

METODOLOGI

3.1 Perancangan Alat



Gambar 3.1 Block Diagram sistem pengatur

Secara *real plant* terdiri dari dua buah bejana atau tangki yang terbuat dari kaca. Tangki yang bawah merupakan tempat penampungan air berukuran 39 x 34 x 35 cm. Tangki yang di atas berfungsi sebagai *plant* yang diatur level ketinggian airnya. Tangki ini berukuran 19 x 14 x 40,5 cm. Sebuah pompa digunakan untuk menaikkan air dari tangki penampungan ke dalam tangki *plant*. Spesifikasi pompa yang digunakan adalah 220 VAC, 50Hz 50 Watt. Untuk mengatur pembuangan air digunakan sebuah *solenoid valve*. *Solenoid valve* digunakan untuk pembuangan air yang dikontrol melalui *personal computer* (PC). Sensor yang digunakan untuk mengukur ketinggian air adalah potensiometer. Untuk mengkonversi gerakan naik-turun ketinggian air menjadi gerakan rotasi yang akan memutar potensiometer,

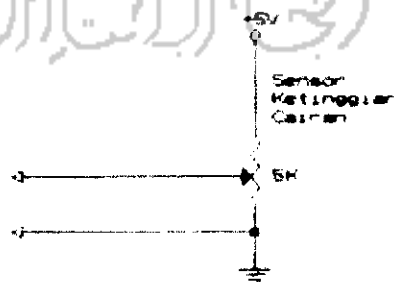
digunakan sebuah pelampung dan pemberat yang dirangkai dalam sebuah sistem katrol. Besar potensiometer yang digunakan adalah 5 k Ω .

Pada gambar 3.1 ditunjukkan blok diagram dari bagian utama sistem pengatur ketinggian air ini, ketinggian permukaan air yang diinginkan diketikkan melalui komputer dan selanjutnya dijadikan sebagai sinyal masukan sistem.

Fungsi tersebut diimplementasikan dalam perangkat lunak. Ketinggian permukaan air yang terjadi dideteksi oleh sebuah sensor dan hasil pengukurannya dijadikan sebagai sinyal umpan balik. Sinyal masukan dan sinyal umpan balik selanjutnya dibandingkan oleh elemen komparator.

3.2 Perancangan Sensor

Berikut gambar 3.2 memperlihatkan rangkaian potensiometer yang mengubah ketinggian air menjadi tegangan. Rangkaian ini mengeluarkan tegangan output sebesar 0 – 1,194 volt untuk *range* pengukuran ketinggian air dari 0 – 30 cm.

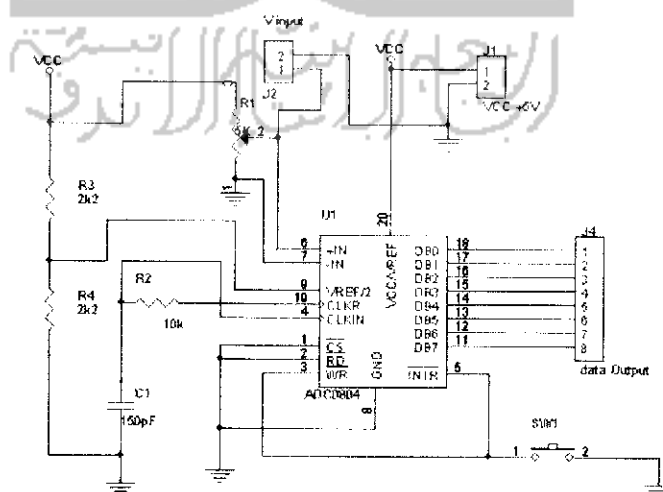


Gambar 3.2 Rangkaian Sensor Potensiometer

3.3 Perancangan Rangkaian ADC

Rangkaian *analog to digital converter* (ADC) diperlukan untuk mengubah data ketinggian air yang sinyal *analog* menjadi *digital* untuk dapat dibaca oleh PC. Dalam aplikasi ini, ADC yang digunakan adalah ADC 0804 yang mempunyai satu input analog dan output digital 8 bit.

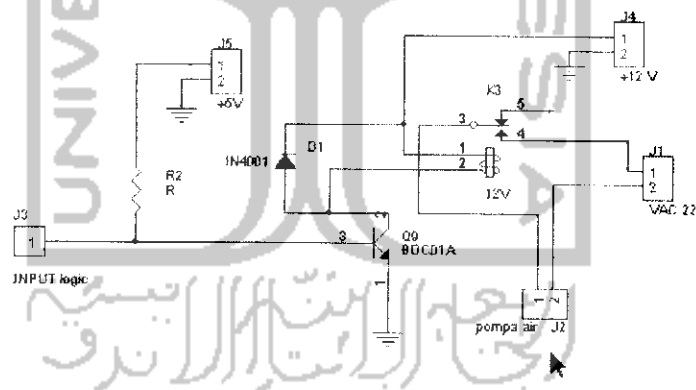
Output ADC dihubungkan ke PC melalui paralel port LPT1. LPT1 hanya mempunyai 5 bit input sedangkan ADC mempunyai 8 bit, maka digunakan sebuah multiplexer yaitu IC 74LS157. IC ini berfungsi sebagai ‘saklar’, jika pin 1 yang terhubung ke *strobe* diberi *low* data akan berpindah dari A ke Y dan jika diberi sinyal *high* maka data berpindah dari B ke Y. Data di A merupakan bit-bit rendah (D3...D0) dan data di B adalah bit-bit tinggi (D7...D4). Kemudian lewat software pergantian data 4 bit tadi akan direkonstruksi ulang menjadi 8 bit kembali. Cara seperti ini dinamakan *mode nibble*.



Gambar 3.3 Rangkaian ADC 0804

3.4 Perancangan Rangkaian Driver Pompa Air

Pompa air dikontrol keluarannya oleh PC secara konstan atau terus menerus. Rangkaian driver pompa air dibentuk dari sebuah relay dan transistor BD139. Tegangan basis dari transistor BD139 di-drive dari PPI 8255 yang terkoneksi dengan komputer melalui port paralel LPT1. Sebuah resistor di *pull up* dengan catu daya 5 volt untuk dapat menggerakkan basis transistor BD139 tersebut. Ketika pada basis transistor tersebut diberi logika 0 dari PC maka pompa air akan aktif. Sebaliknya, ketika pada basis transistor diberi logika 1 maka pompa air akan mati. Kedua hal ini terjadi karena pada basis transistor telah di *pull up* dengan catu daya 5 volt.

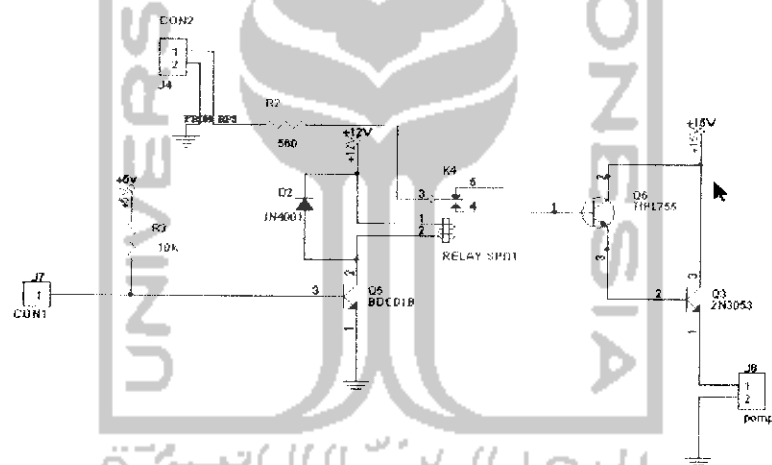


Gambar 3.4 Rangkaian Driver Pompa Air

3.5 Perancangan Rangkaian Driver Solenoid Valve

Proses kontrol PID dilakukan oleh PC. Output dari kontroler digunakan untuk menggerakkan solenoid valve untuk pembuangan air. Rangkaian driver *solenoid* digunakan dua buah transistor NPN yaitu 2N3005 dan TIP31A. Kedua transistor ini

dihubungkan membentuk rangkaian darlington agar mendapatkan penguatan yang besar, sehingga arus I_b transistor yang digunakan kecil. Sebuah transistor BD139 dan relay digunakan untuk mematikan dan menyalakan *solenoid valve*. Pada rangkaian ini sebuah resistor juga di *pull up* oleh catu daya 5 volt yang prinsip kerjanya sama dengan prinsip kerja dari driver pompa air. Rangkaian driver *solenoid valve* mendapat inputan dari output rangkaian digital to analog converter (DAC) yang dilewatkan sebuah rangkaian *inverting*.

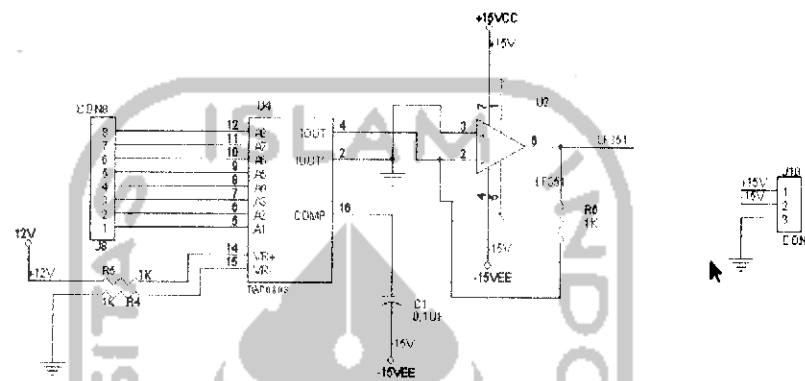


Gambar 3.5 Rangkaian Driver Solenoid valve

3.6 Perancangan Rangkaian DAC

Rangkaian DAC (*Digital to Analog Converter*) digunakan untuk mengubah data digital menjadi data analog untuk meng-aktifkan driver *solenoid valve*. Dalam aplikasi ini, DAC yang digunakan adalah DAC 0808 yang merupakan DAC 8 bit

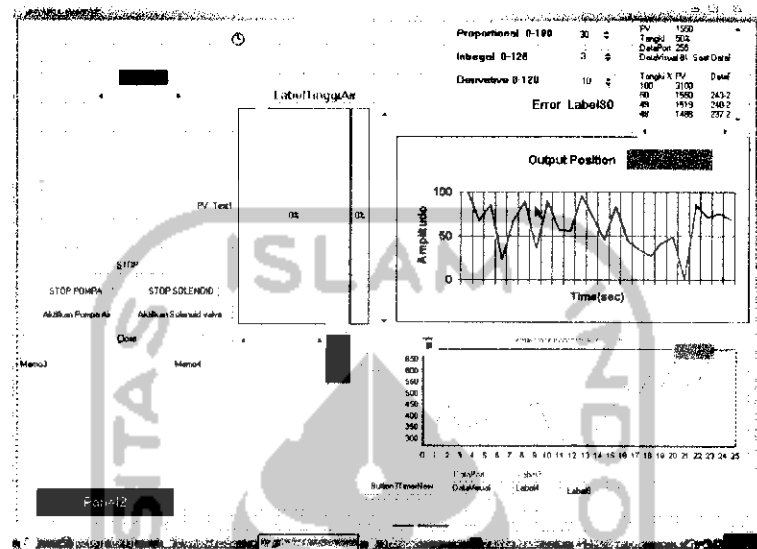
dengan output dari 0 sampai 5 volt. Rangkaian DAC mendapat input dari PPI8255 melalui paralel port dari LPT1.



Gambar 3.6 Rangkaian DAC 0808

3.7 Perancangan Perangkat Lunak

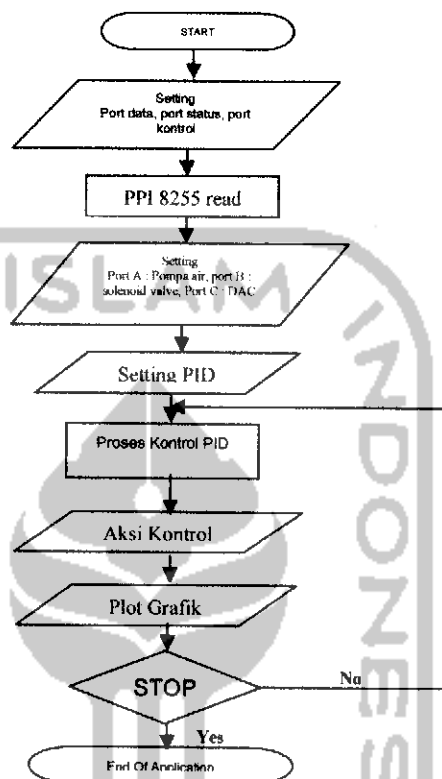
Perancangan perangkat lunak ini berangkat dari sebuah program yang dibuat oleh Clayton yang menggunakan bahasa pemrograman Delphi 7. Perancangan perangkat lunak yang akan dibuat digunakan untuk mengatur semua sistem dari perangkat keras yang telah dibuat yang bertujuan untuk mengirimkan intruksi-intruksi ke ADC, driver *solenoid valve*, driver pompa air, serta PPI8255. Perangkat lunak ini selain memberikan intruksi juga dapat membaca ADC, menampilkan visual tangki air, serta dapat menampilkan grafik PID.



Gambar 3.7 Perancangan GUI

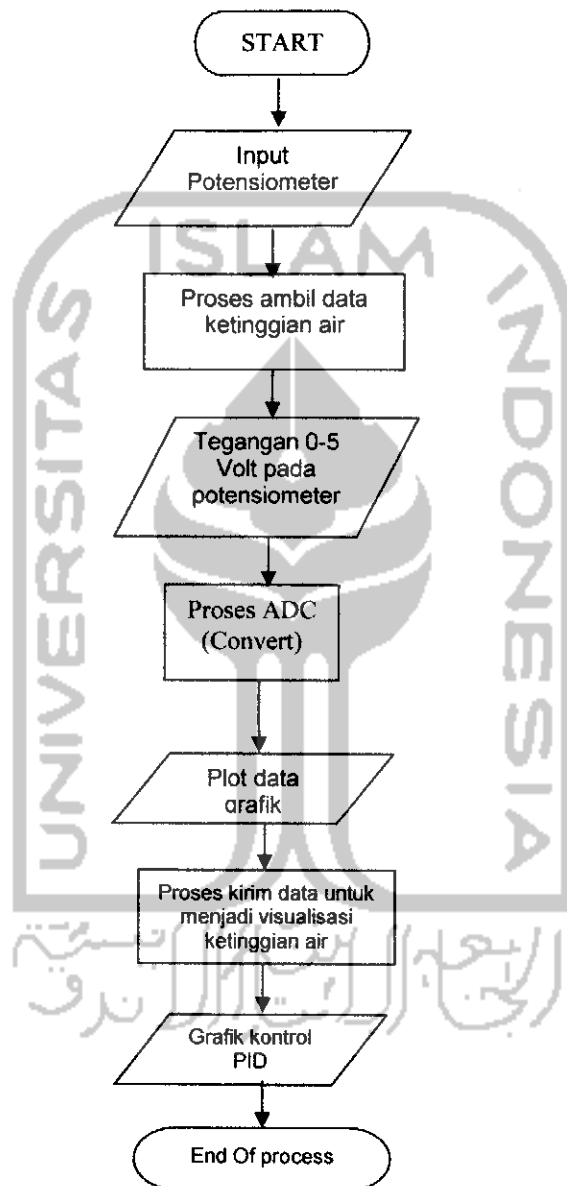
Sebelum membuat program atau perangkat lunak, terlebih dahulu membuat diagram alir (*flow chart*) dari program tersebut. Diagram alir ini dibuat berdasarkan perangkat keras yang telah dirancang sehingga dapat menjalankan sistem sesuai yang diinginkan.

Diagram alir pada gambar 3.8 adalah proses inisialisasi dari program kontrol PID. Dimana PPI 8255 yang melakukan komunikasi dengan port data dari port paralel yang Port A, Port B, dan port C dapat digunakan sebagai input/output. Port A digunakan untuk mengaktifkan pompa air. Port B digunakan untuk mengontrol *solenoid valve* yang mendapat inputan dari DAC yang dikontrol melalui Port C dari PPI 8255.



Gambar 3.8 Diagram alir program kontrol

Diagram alir diatas juga menjelaskan program yang dibuat bertugas untuk memonitor ketinggian air, melakukan proses kontrol dan menggerakkan pompa air dan *solenoid valv*. Proses ini terus dilakukan sampai ketinggian air dalam tangki sesuai dengan yang diinginkan. Proses perhitungan kontrol PID menggunakan persamaan PID pada umumnya yang telah diubah dalam bentuk diskrit untuk memudahkan pemrograman. DAC digunakan untuk mengontrol *solenoid valve* yang berfungsi sebagai *output* kontrol dari proses PID. Ketika ketinggian air yang akan diukur telah mencapai ketinggian yang diinginkan maka *solenoid valve* akan membuka sesuai *output* pada tangki visual.

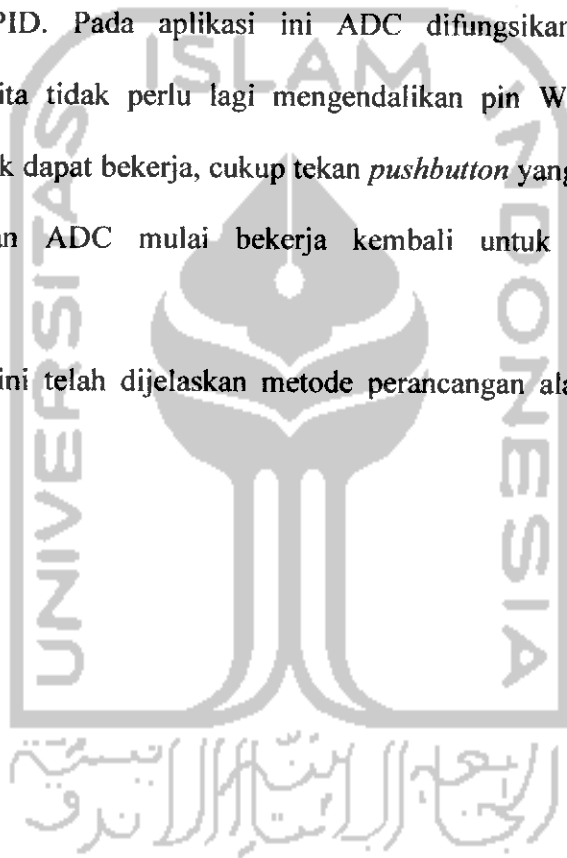


Gambar 3.8 Diagram alir program ADC

Diagram alir pada gambar 3.8 diatas merupakan inisialisasi dari program pembacaan ADC. Pada awalnya potensiometer yang digunakan sebagai sensor untuk membaca ketinggian air mengeluarkan tegangan *output* sebesar 0-5 Volt dan

tegangan analog tersebut akan di konversi oleh ADC dan menjadikannya data digital untuk menghasilkan data dan grafik ketinggian air. Dari grafik dan data tersebut akan dikonversi lagi untuk visual tangki air yang berhubungan langsung dengan grafik proses kontrol PID. Pada aplikasi ini ADC difungsikan dalam mode *free-running*, artinya kita tidak perlu lagi mengendalikan pin WR, RD dan CS. Jika sesekali ADC tidak dapat bekerja, cukup tekan *pushbutton* yang terhubung ke *ground* untuk memastikan ADC mulai bekerja kembali untuk membaca data dari potensiometer.

Pada bab ini telah dijelaskan metode perancangan alat dari seluruh sistem yang akan dibuat.



BAB IV

PENGUJIAN, ANALISIS, DAN PEMBAHASAN

Pada bab ini, akan dijelaskan tentang bagaimana hasil perancangan sistem dan analisis dari sistem yang telah dibangun berdasarkan perancangan yang telah dijelaskan pada bab sebelumnya. Terkait dengan analisis sistem, agar memudahkan pembahasan pada bab ini maka akan di buat sistematika pembahasan yang terdiri dari :

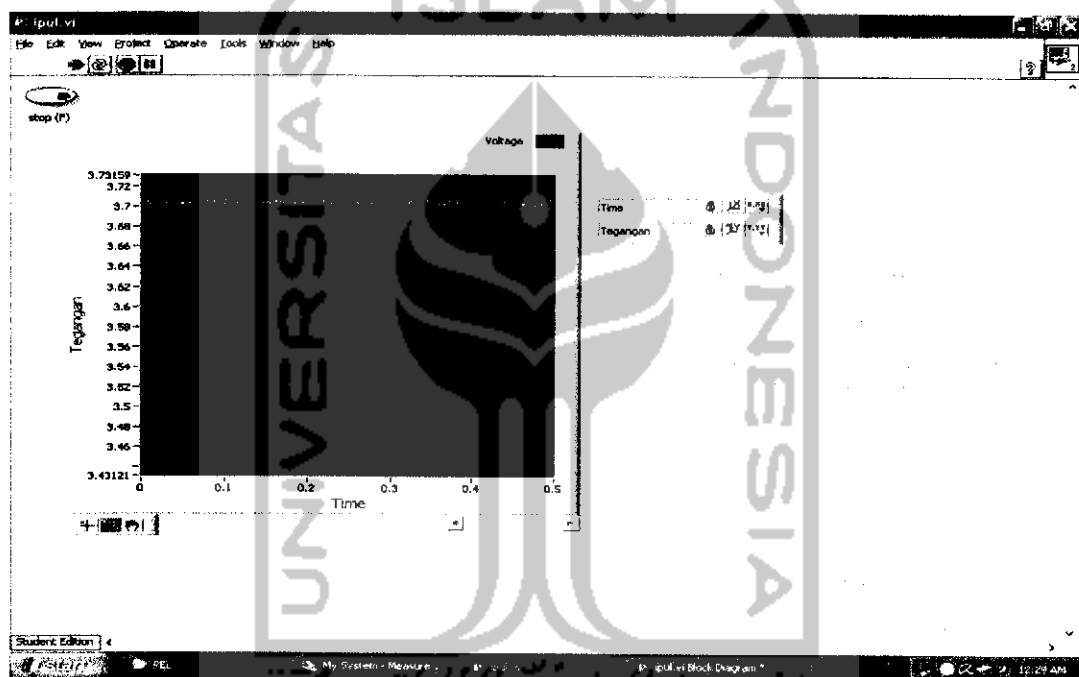
1. Analisis pada perangkat keras
2. Analisis pada perangkat lunak
3. Analisis parameter PID untuk optimasi pengontrolan ketinggian level air .

4.1 Analisis Pada Sisi Perangkat Keras

4.1.1 Setting Potensiometer

Sistem pengontrolan level ketinggian air ini terdapat di potensiometer yang berfungsi sebagai sensor. Potensiometer ini akan terhubung oleh ADC sebagai konverter. Dimana setiap putaran dari potensiometer akan mengeluarkan tegangan VDC dan akan dikonversi menjadi tegangan digital oleh ADC. Pada dasarnya ketika potensiometer ini diputar penuh, maka tegangan keluarannya adalah 0 - 5 VDC. Akan tetapi pada perancangan sistem ini, potensiometer tidak di putar penuh, oleh karena itu untuk menghasilkan akuisisi data yang diinginkan yaitu 0 - 255 bit/putaran, maka pada ADC kita harus melakukan *setting* pada tegangan referensinya.

Pengujian karakteristik ke-linearan sebuah potensiometer yang digunakan sebagai sensor pengukur level air ini dapat dilihat menggunakan sebuah *software* DAQ 6009. Pada gambar 4.1 dibawah ini dapat dilihat sebuah grafik dari pengujian potensiometer.



Gambar 4.1 Pengujian karakteristik Potensiometer menggunakan DAQ 6009

Ketika tegangan *input* maksimal potensiometer adalah 5 V, maka tegangan referensi yang harus diberikan adalah,

$$V_{REF} = V_{IN \text{ MAKS}} / 2 = 5V / 2 = 2.5 \text{ Volt} \quad (4.1)$$

Tegangan resolusi,

$$V_{RESOLUSI} = V_{IN \text{ MAKS}} / 255 = 5V / 255 = 0.0196 \text{ V/Bit} \quad (4.2)$$

Pada sistem perangkat keras yang akan dibuat tegangan inputnya adalah 5V/10 putaran = 0.5 Volt

Jadi tegangan referensi yang harus diberikan pada ADC adalah,

$$V_{REF} = V_{IN \text{ MAKS}} / 2 = 0.5V / 2 = 0.25 \text{ Volt} \quad (4.3)$$

Tegangan Resolusi,

$$V_{RESOLUSI} = V_{IN \text{ MAKS}} / 255 = 0.5V / 255 = 0.00196 \text{ V/Bit} \quad (4.4)$$

Tabel 4.1 Hasil Pengujian Sensor

Tegangan Sensor	Output (Desimal)	Output (Biner)
0	0	00000000
0.00196	1	00000001
0.00392	2	00000010
0.00588	3	00000011
0.00784	4	00000100
0.0098	5	00000101
0.01176	6	00000110
0.01372	7	00000111
0.01568	8	00001000
0.01764	9	00001001
0.0196	10	00001010
0.02156	11	00001011
0.02352	12	00001100
0.02548	13	00001101
0.02744	14	00001110
...	...	...
0.5	255	11111111

Tabel diatas menunjukkan *output* dari potensiometer, dimana setiap data yang sudah di konversi menjadi biner akan menjadi data untuk mengukur ketinggian air. Data 0 – 255 digunakan untuk mengukur ketinggian air dari 0 – 30 cm atau 0 – 50% dari tinggi tangki air.

4.1.2 Akuisisi Data Rangkaian ADC

Perancangan ADC ini menggunakan metode *free running* yaitu proses konversinya secara terus menerus selama ada perubahan tegangan *input*. Tujuan pengujian rangkaian ADC 0804 untuk mengetahui tegangan hasil konversi bilangan biner dari tegangan *input* ADC. Prosedur pengujian yaitu :

- Pemberian catu daya 5 volt pada rangkaian ADC
- Memberikan tegangan input ke ADC secara bervariasi.
- Pengambilan data ADC

Setelah mengamati hasil pengujian ADC didapatkan data pada tabel berikut.

Tabel 4.2 Hasil Pengujian Pada Rangkaian ADC 0804

Tegangan Input	HASIL KONVERSI BINER								Dalam Heksa
	A7	A6	A5	A4	A3	A2	A1	A0	
0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	1	1	33
1.1	0	0	1	1	1	0	0	0	38
1.2	0	0	1	1	1	1	0	1	3D
1.3	0	1	0	0	0	0	1	1	67
1.4	0	1	0	0	1	0	0	0	48
1.5	0	1	0	0	1	1	0	1	4D
1.6	0	1	0	1	0	0	1	0	52
1.7	0	1	0	1	0	1	1	1	57
1.8	0	1	0	0	1	1	0	0	5C
1.9	0	1	1	0	0	0	0	1	61

Tabel 4.2 Lanjutan

Tegangan Input	HASIL KONVERSI BINER								Dalam Heksa
	A7	A6	A5	A4	A3	A2	A1	A0	
2	0	1	1	0	0	1	1	1	66
2.1	0	1	1	1	1	0	1	1	6B
2.2	0	1	1	1	0	0	0	1	71
2.3	0	1	1	1	0	1	1	0	76
2.4	0	1	1	1	1	0	1	1	7B
2.5	1	0	0	0	0	0	0	0	80
2.6	1	0	0	0	0	1	0	1	85
2.7	1	0	0	0	1	0	1	0	8A
2.8	1	0	0	0	1	1	1	1	8F
2.9	1	0	0	1	0	1	0	0	94
3	1	0	0	1	1	0	1	0	9A
3.1	1	0	0	1	1	1	1	1	9F
3.2	1	0	1	0	0	1	0	0	A4
3.3	1	0	1	0	1	0	0	1	A9
3.4	1	0	1	0	1	1	1	0	AE
3.5	1	0	1	1	0	0	1	1	B3
3.6	1	0	1	1	1	0	0	0	B8
3.7	1	0	1	1	1	1	0	1	BD
3.8	1	1	0	0	0	0	1	1	C3
3.9	1	1	0	0	1	0	0	0	C8
4	1	1	0	0	1	1	0	1	CD
4.1	1	1	0	1	0	0	1	0	D2
4.2	1	1	0	1	0	1	1	1	D7
4.3	1	1	0	1	1	1	0	0	DC
4.4	1	1	1	0	0	0	0	1	E1
4.5	1	1	1	0	0	1	1	0	E6
4.6	1	1	1	0	1	0	1	1	EB
4.7	1	1	1	1	0	0	0	1	F1
4.8	1	1	1	1	0	1	1	0	F6
4.9	1	1	1	1	1	0	1	1	FB
4.98	1	1	1	1	1	1	1	1	FF

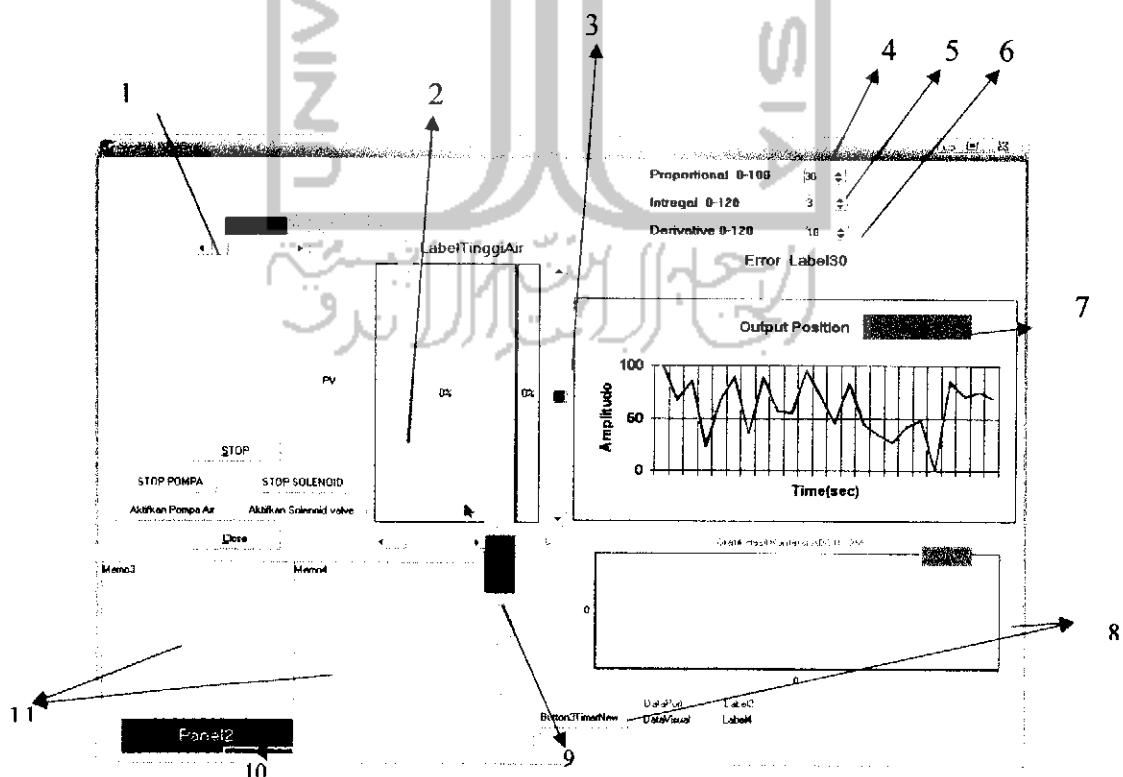
4.1.3 Solenoid Valve Sebagai Output

Output dari sistem ini menggunakan *solenoid valve*. Dimana *output valve* ini juga sebagai pembentuk grafik kontrol PID. *Valve* di kontrol melalui *input* logika dari PPI8255, dimana sebelum masuk ke *valve* yang membutuhkan tegangan

analog, maka bit dari PPI8255 akan di konversi menjadi tegangan oleh DAC 0808. *Valve* ini memerlukan input tegangan 0 – 5 Volt. Sehingga ketika input logika dari PPI8255 sebesar FFH (FF Heksa) maka DAC akan meng-konversi nya menjadi tegangan 5 Volt. *Valve* akan aktif mengikuti *output* sistem dari kontrol PID.

4.2 Analisis Perangkat Lunak

Perangkat lunak ini menggunakan bahasa pemrograman *PASCAL* dengan *software* Delphi 7. Sistem dibangun untuk menampilkan GUI (*Graphical User Interface*) dari pemrograman kontrol PID. Gambar 4.2 ini adalah hasil dari visualisasi perancangan pemrograman sistem kendali PID dengan menggunakan Delphi 7.



Gambar 4.2 Tampilan GUI kontrol PID

Keterangan Gambar :

1. **HScroll1** : Merupakan *input* debit air yang masuk kedalam tangki visual. Ketika tombol aktifkan pompa air di “klik” maka secara otomatis HScroll1 akan membuka penuh (100%) secara konstan sebagai masukan *input*. Pada tangki yang asli pompa air akan aktif dan akan mengisi tangki secara terus menerus dengan debit air yang konstan dan akan berhenti ketika tombol stop pompa di “klik”.
2. **Tangki air visual** : Ini merupakan visualisasi dari tangki air sebenarnya. Dimana ketika air terisi didalam tangki yang sebenarnya maka tangki air visual ini akan terisi juga secara otomatis. Tangki air visual ini terisi dengan pembacaan data bit dari sensor yang telah dikonversi oleh ADC. Visual dari tangki air ini berproses dengan *sampling time* 2 detik yang diatur oleh *timer*, yaitu setiap 2 detik pembacaan data dari sensor maka dilakukan proses ketinggian air yang berulang-ulang hingga batas air yang diinginkan. Proses ketinggian air ini terbaca dalam persentase .
3. **VScroll1** : Digunakan sebagai pengatur ketinggian air yang diinginkan. VScroll1 ini juga merupakan *setting point* (SP) dari level ketinggian air. Ketika VScroll1 kita *setting* sebesar 50%, maka tangki air visual akan berproses sebagai *process variable* yang akan mencapai nilai *setting point*. Hal ini terjadi karena telah mengalami proses kontrol yang sebenarnya, dimana ketika error yang kita inginkan sama dengan nol (0) maka harus terjadi proses dimana sesuai dengan persamaan :

$$E = SP - PV \quad (4.4)$$

Dimana,

E = Error

SP = *Setting Point*

PV = *Process Variable*

- 4, 5, 6. **Parameter PID** : Di bagian ini merupakan tempat untuk *setting* parameter- parameter kontrol PID (*Proporsional, Integral, Derivative*) untuk mendapatkan respon sistem yang paling baik.
7. **VtChart1** : Ini merupakan *output* grafik dari kontrol PID yang telah di *setting* parameter-nya. Karena sistem yang dibuat ini yaitu mengontrol *output* keluaran *valve*, maka grafik ini akan menampilkan nilai *output* keluaran dari *setting* parameter. Grafik VtChart1 ini akan berhubungan langsung dengan Hscroll2 (no. 9) sebagai *output* dari sistem. Dari grafik di label Y terdapat amplitudo yang merupakan *output* keluaran, sedangkan di label X merupakan pewaktu (*time*) untuk mengetahui pada detik seberapa waktu proses kontrol PID mulai terjadi.
8. **Chart1** : Ini merupakan grafik tegangan dari sensor potensiometer yang sudah dikonversi oleh ADC menjadi biner. Dimana tiap bit data dari ADC akan ditampilkan pada grafik.
9. **HScroll2** : Merupakan *output* dari sistem yang dalam perangkat keras yang nanti akan terhubung dengan *solenoid valve* sebagai keluaran air. Grafik dari

proses kontrol PID berawal dari bukaan HScroll2. Dimana HScroll2 akan sedikit demi sedikit terbuka sampai menghasilkan *output* respon yang baik. HScroll2 ini dikontrol secara otomatis yang tergantung dari *output* sistem.

10. **Panel2** : Ini digunakan sebagai penampil jam dan tanggal pada saat proses kontrol dimulai.
11. **Memo** : Ini digunakan sebagai penampil proses PID yang telah terjadi dalam perangkat lunak.

4.3 Analisis Parameter PID untuk Optimasi Level Ketinggian Air

Pengujian sistem pengaturan level ketinggian air dilakukan dengan berbagai parameter konstanta Kp (Konstanta *Proportional*), Ki (Konstanta *Integral*), dan Kd (Konstanta *Derivative*). *Range* pengukuran ketinggian air yang dilakukan adalah 0 – 50% yang dikonversikan oleh ADC menjadi data 0 – 255. Dalam pengujian ini data ketinggian air yang digunakan adalah dalam skala yang ditunjukkan oleh ADC 0 – 255. Parameter yang dimasukan berbeda – beda.

4.3.1 Pengujian Sistem Kontrol PID

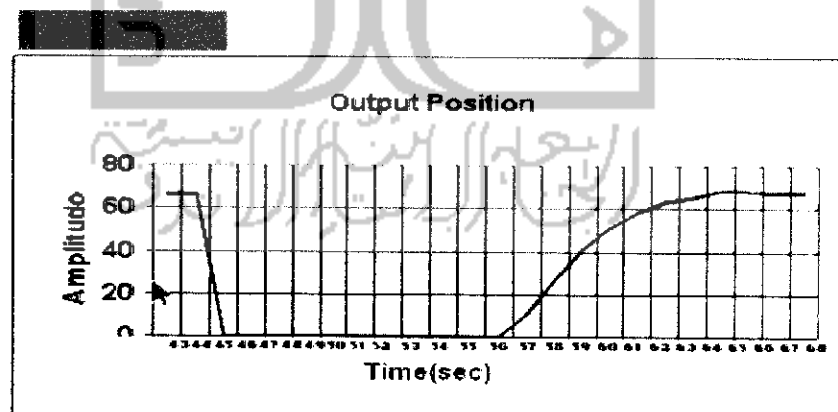
4.3.1.2 Pengujian Sistem Dengan Perubahan Parameter

4.3.1.2.1 Perubahan Parameter Kp

Tabel 4.3 Hasil Pengujian terhadap parameter Kp

Setting Point (SP) %	Konstanta PID			Data ADC	Rise Time (Sec)	Steady State (Sec)	Over Shoot	Error
	Kp	Ki	Kd	Bit				
30%	85	10	3	118	54.8	63	-	0
	75	10	3	130	56.6	67	65	-2
	65	10	3	148	59.8	72	67	0
	55	10	3	145	59.7	72	68	-1
	45	10	3	139	58.7	71	68	-5

Pada tabel percobaan diatas dapat dilihat pengaruh Kp terhadap sistem. Pertambahan harga Kp akan menaikkan penguatan sistem sehingga dapat digunakan untuk memperbesar kecepatan respon dan mengurangi error penyimpangan dalam keadaan mantap (*error steady state*).



Gambar 4.3 Respon output sistem dengan Kp=75, Ki=10, Kd=3

Gambar 4.3 menunjukkan pemilihan Kp yang tidak optimal pada sistem. Pemakaian alat kontrol jenis ini saja sering tidak memuaskan karena penambahan K selain membuat sistem lebih sensitif, tetapi juga cenderung mengakibatkan kestabilan.

Disamping itu penambahan K_p adalah terbatas dan tidak cukup untuk mencapai respon sampai suatu harga yang diinginkan.

Kenyataan dalam usaha mengatur harga K_p terdapat keadaan-keadaan yang bertentangan. Di satu pihak diinginkan mengurangi error sebanyak mungkin, tetapi hal ini akan mengakibatkan osilasi bagi respons yang berarti membutuhkan waktu yang lama pada "*setting time*", sedang di pihak lain respons terhadap setiap perubahan masukan harus terjadi secepat mungkin tetapi dengan lonjakan dan osilasi sekecil mungkin. Respons yang cepat memang dapat diperoleh dengan memperbesar K_p , tetapi hal ini juga akan mengakibatkan ketidak stabilan sistem.

Untuk mengatasi masalah-masalah ini, alat kontrol yang akan digunakan harus mempunyai persyaratan berikut :

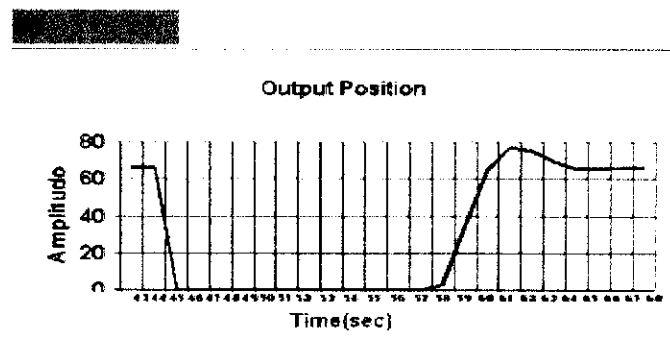
- a. penguatan yang tinggi pada frekuensi-frekuensi yang sangat rendah (untuk mengurangi kesalahan-kesalahan).
- b. Penguatan yang tinggi pada frekuensi-frekuensi tinggi (yakni dengan secepatnya mengikuti perubahan masukan bila laju perubahan transien adalah yang paling cepat). Hal ini penting untuk menjamin respon yang cepat.
- c. Pada frekuensi-frekuensi menengah (yakni dalam bagian terakhir respons transient dan sebelum "*on set*" (kondisi - kondisi mantap) penguatan sebaiknya cukup rendah agar terjamin proses yang tidak mengalami lonjakan yang berlebihan dan juga setiap kecenderungan berosilasi akan diredam dengan cepat.

4.3.1.2.2 Perubahan Parameter K_i

Tabel 4.4 Hasil Pengujian terhadap parameter Ki

Setting Point (SP) %	Konstanta PID			Data ADC	Rise Time (Sec)	Steady State (Sec)	Over Shoot	Error
	Kp	Ki	Kd	Bit				
30%	85	10	3	118	54.8	63	-	0
	85	20	3	129	58.5	68	63	0
	85	30	3	125	57.6	66	61.5	0
	85	40	3	120	56.6	64	59.6	0
	85	50	3	118	56.6	64	59.6	0

Dari tabel hasil percobaan 4.4 diatas dapat dilihat pengaruh Ki pada sebuah sistem kontrol. Dari semua parameter Ki yang dimasukkan mengalami *over shoot* dan hanya percobaan pertama pengujian terhadap parameter Ki ini. Pada dasarnya sifat dasar kontrol Ki adalah dapat memperbaiki respon *steady state*, namun pemilihan Ki yang tidak tepat atau terlalu besar dapat menyebabkan respon *transient (transient respon)* yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan Ki yang sangat tinggi justru dapat menyebabkan *output* berisolasi.

Gambar 4.4 Respon output sistem dengan $K_p=85$, $K_i=30$, $K_d=3$

Gambar 4.4 menunjukkan respon sebuah kontrol yang tidak optimal dengan pemilihan K_i yang besar.

4.3.1.2.3 Perubahan Parameter K_d

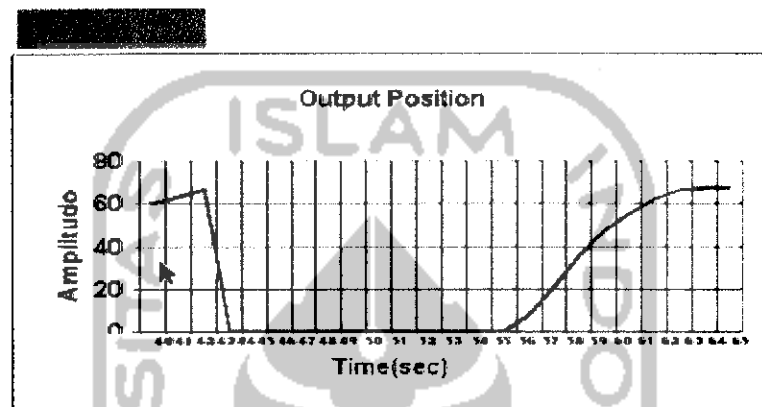
Tabel 4.5 Hasil Pengujian terhadap parameter K_d

Setting Point (SP) %	Konstanta PID			Data ADC	Rise Time (Sec)	Steady State (Sec)	Over Shoot	Error
	K_p	K_i	K_d	Bit				
30%	85	10	3	118	54.8	63	-	0
	85	10	10	118	54.8	63	.*	-1
	85	10	20	118	55.6	64	.*	-3
	85	10	30	118	55.6	64	.*	-5
	85	10	40	118	55.6	64	.*	-7

Keterangan : .* => *Over Shoot* yang tidak kelihatan di grafik

Tabel 4.3 merupakan hasil pengujian ketika parameter K_d kita berikan nilai bervariasi. Pada percobaan tersebut terdapat konstanta PID yang paling handal atau ideal untuk mengatur level ketinggian air dengan *setting point* 30%. Untuk konstanta kontrol yang paling tepat adalah dengan memasukan harga $K_d = 3$. Pemilihan K_d yang tepat dapat meningkatkan stabilitas sistem, mengurangi *over shoot*, dan meningkatkan respon transien. Dengan memberikan K_d yang ideal untuk tiap jenis pengontrolan maka dapat menghasilkan respon yang cepat mencapai *error*. Apabila K_d yang kita berikan terlalu besar maka stabilitas sistem akan terganggu. Pemberian

Kd yang kecil tidak hanya berpengaruh pada cepat nya kondisi *rise time* tapi juga berpengaruh pada saat sistem mengalami *steady state*.



Gambar 4.5 Respon output sistem dengan $K_p=85$, $K_i=10$, $K_d=40$

Gambar 4.5 menunjukkan sebuah respon output sistem yang kurang optimal dengan pemilihan K_d yang besar

4.3.1.2 Pengujian Sistem dengan perubahan *setting point*

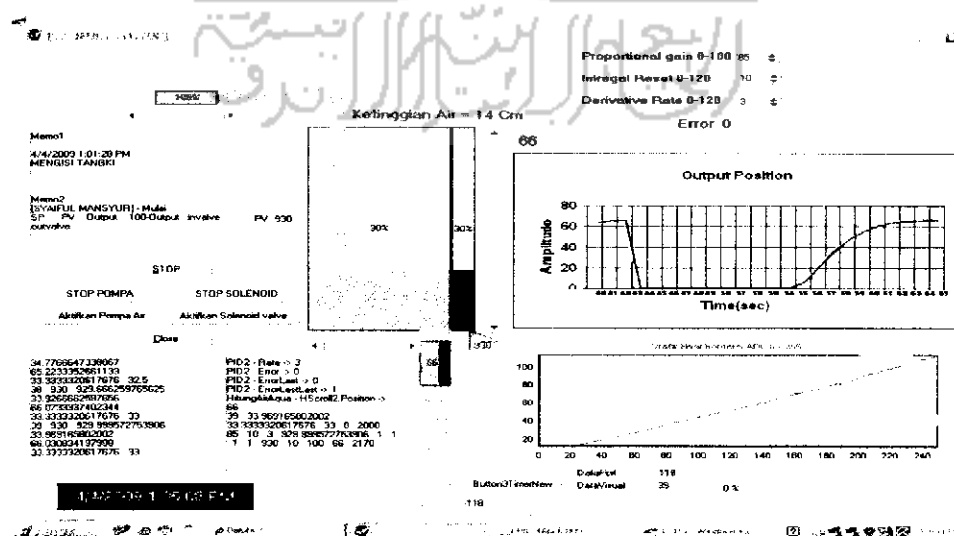
Setelah melakukan percobaan dengan melakukan perubahan pada parameter-parameter kontrol K_p , K_i , dan K_d maka didapatkan parameter kontrol yang paling optimal adalah $K_p=85$, $K_i=10$, dan $K_d= 3$. Selanjutnya dalam pengujian ini, dilakukan percobaan dengan parameter konstanta kontrol sama dan *setting point* yang berbeda. Tujuan dari percobaan ini untuk mengetahui kehandalan pengendali PID.

Tabel 4.6 Hasil Pengujian terhadap *setting point* PID

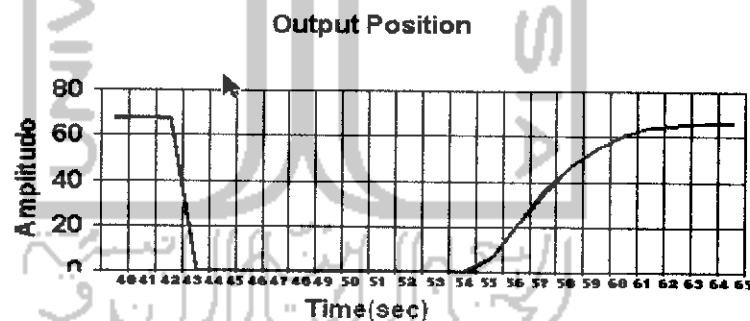
Setting Point (SP)	Konstanta PID			Data ADC	Rise Time (Sec)	Steady State (Sec)	PV
	Kp	Ki	Kd	Bit			
30%	85	10	3	118	54.8	63	930
35%	85	10	3	126	58.8	67	1035
40%	85	10	3	141	63.6	72	1240
45%	85	10	3	153	67.6	75	1395
50%	85	10	3	168	71.9	80	1550

Pada tabel 4.6 diatas adalah hasil pengujian dengan parameter yang sama namun dengan *setting point* yang berbeda. Hal ini dilakukan untuk mengetahui sejauh mana parameter tersebut handal pada setiap *setting point*, maka dari tabel diatas diketahui semua *setting point* pada sistem tersebut handal dengan parameter tersebut.

Nilai *setting point* ketinggian air 930 dan menunjukkan ketinggian air 30% yang dilihat dari visual sedangkan di keadaan sebenarnya ketinggian air 14cm.

Gambar 4.6 Tampilan GUI kontrol PID setelah di *Run*

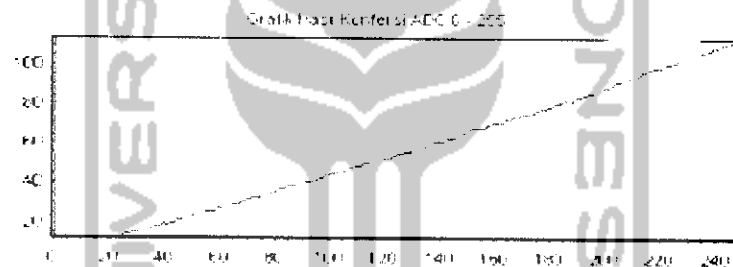
Gambar 4.6 menunjukkan hasil visualisasi kontrol PID yang mulai di proses setelah tombol aktifkan pompa air di “klik”. Pada awalnya semua parameter kontrol di *setting* $K_p = 85$, $K_i = 10$, dan $K_d = 3$. Pada tangki air visual akan terjadi proses hingga *setting point* yang telah di tetapkan yaitu 930 atau sebesar 30%. Pada tangki air yang sebenarnya air akan mengisi masuk hingga 14 cm. Ketika proses pengisian pada tangki air visual dan pada tangki sebenarnya telah mencapai *setting point* maka *output* sistem akan terbuka sebesar 66% dan pada *solenoid valve* yang telah di *setting* keluarannya akan terbuka sebesar 70 Heksa yang menyesuaikan *output* 66% , dimana *output* sistem inilah yang menjadi *output* dari PID.



Gambar 4.7 Respon output sistem dengan $K_p=85$, $K_i=10$, $K_d=3$

Pada gambar 4.4 diatas, waktu naik (*rise time*) terjadi pada detik ke-54,8. Dengan memasukan harga K_p yang besar akan memperkecil *overshoot* dan mempercepat pencapaian *steady state*. Pencapaian *steady state* untuk mendapatkan $error = 0$, terjadi pada detik ke- 63.

Untuk membuat tangki air visual dan tangki air sebenarnya terkoneksi adalah melalui data dari sensor potensiometer yang telah di konversi oleh ADC 0804. Untuk percobaan ini kita *setting point* sebesar 930 maka tangki air sebenarnya akan diisi hingga data dari sensor terbaca 118 bit. Saat *setting point* yang di berikan berbeda-beda maka data dari sensor untuk mencapai *setting point* tersebut juga bervariasi. Hal ini akan didapatkan di percobaan selanjutnya. Berikut gambar grafik dari ADC pada percobaan pertama ini,

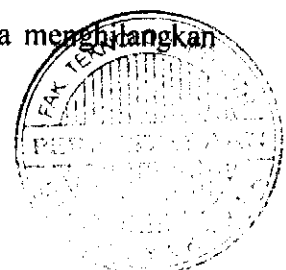


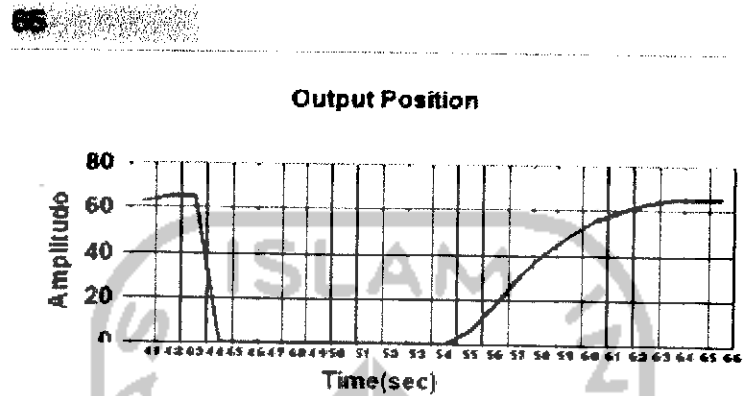
Gambar 4.8 Grafik ADC setelah di Run

Pada *setting point* selanjutnya maka diberikan masukkan nilai *setting point* yang berbeda hingga 50%, maka pengaruh dari *output* terdapat di semua kondisi yaitu pada data ADC, *steady state*, dan *rise time*. Semakin besar *setting point* yang kita masukkan maka semakin besar pula kondisi-kondisi yang dialami sistem.

4.3.2 Pengujian Sistem Kontrol PI

Pengujian yang dilakukan dengan menggunakan kontrol PI ini untuk melihat seberapa besar pengaruh kontrol PI terhadap sistem tanpa menambahkan kontrol D (*derivative*) didalam sistem tersebut. *Integral Controller* memiliki karakteristik mengurangi *rise time*, menambah *overshoot* dan *setling time*, serta menghilangkan





Gambar 4.9 Respon output sistem dengan $K_p=70$, $K_i=8$

Untuk membuat sistem stabil tanpa menambahkan kontrol K_d dipercobaan ini maka parameter K_p haruslah lebih besar daripada parameter K_i , apabila pemilihan K_i dan K_p tidak sesuai maka akan membuat sistem akan lambat mencapai $error = 0$ atau bahkan akan membuat sistem sedikit berisolasi.

Pengaruh kontrol PID pada valve keluaran air adalah ketika persamaan kontrol PID yang telah di masukkan pada sistem ini misalnya, sensor potensiometer menghitung setiap 1 detik atau lebih pendek, sehingga setiap 1 detik akan didapatkan posisi kontrol valve yang baru. Error pada persamaan PID dihitung dari perbedaan *set-point* dengan proses variabel. Jadi setiap 1 detik didapatkan *error* yang baru untuk dihitung oleh PID untuk menghasilkan manipulated variabel yang baru.

Manipulated variabel (MV) = aliran air keluaran yang diatur

Transmitter = pelampung untuk mengukur tinggi air

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa sistem kontrol PID yang dibuat mampu mengontrol level ketinggian air. Secara umum untuk aplikasi yang telah dibuat, semakin besar harga K_p maka sistem lebih cepat mencapai *steady state* dan juga *rise time* semakin kecil. Semakin besar harga K_i akan membuat respon sistem berisikasi dan lebih lambat mencapai keadaan *steady state*. Semakin besar harga K_d dapat membuat *overshoot* semakin kecil. Parameter kontrol PID yang optimum untuk pengontrolan ketinggian air ini adalah, $K_p = 80$, $K_i = 10$, $K_d = 3$ dan pada tangki air terbaca 14cm. Pada pengontrolan kontrol PI maka parameter yang tepat di sistem ini adalah, $K_p = 70$, $K_i = 8$ dan pada tangki air terbaca 15cm.

5.2 Saran

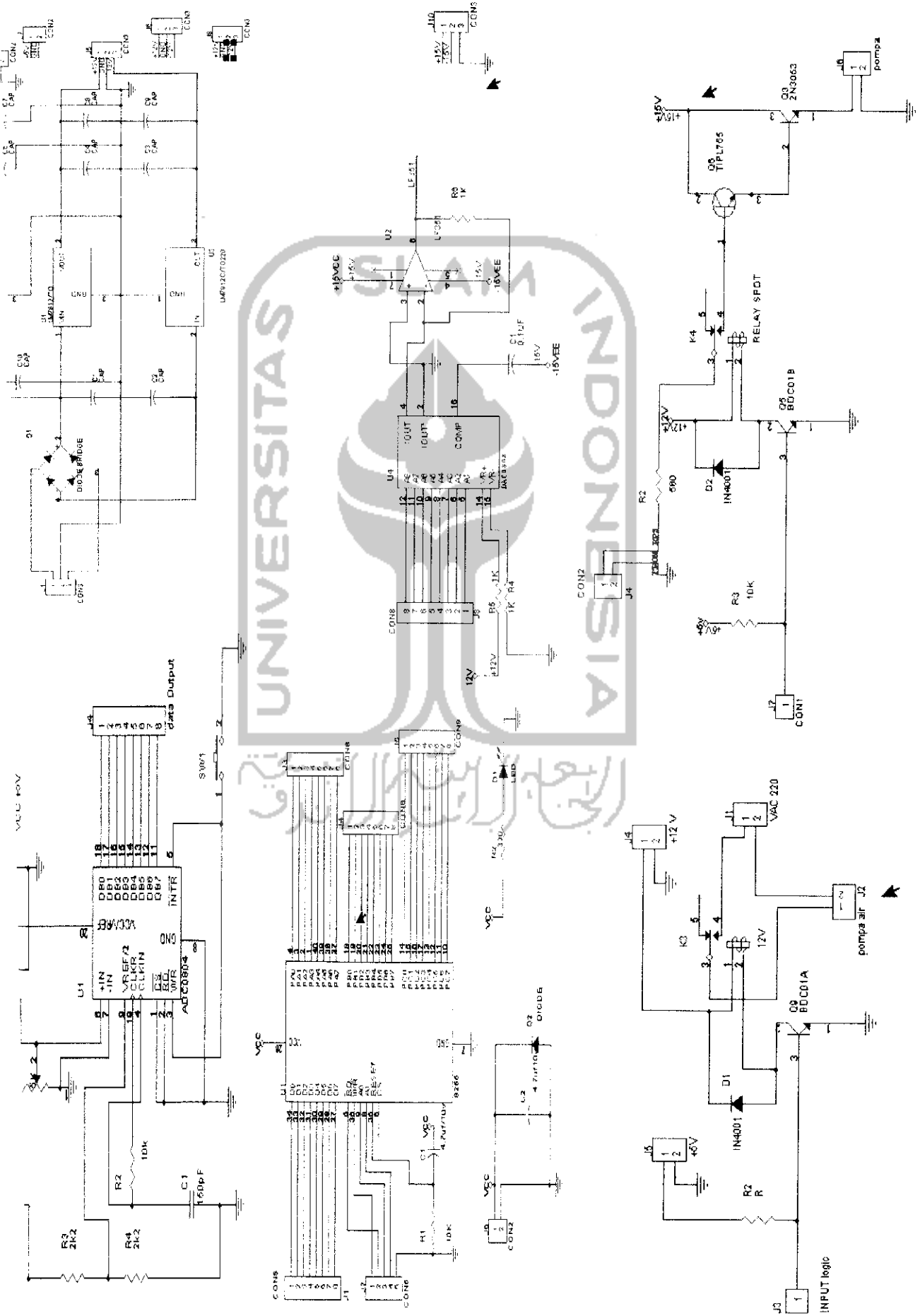
Dari perancangan sistem yang telah direalisasikan pada Tugas Akhir ini diharapkan dapat menjadi dasar penelitian lebih lanjut, mengingat banyaknya keterbatasan yang dihadapi maka diusulkan beberapa saran pengembangan, yaitu :

1. Penggantian sensor dari potensiometer menjadi sensor yang lebih presisi dan memiliki sedikit *noise* untuk mengukur level ketinggian air.

2. *Solenoid valve* yang digunakan selanjutnya lebih baik agar dapat mengukur keluaran air dengan pasti, misalnya menggunakan motor servo sebagai *interfacing* nya.
3. Pengujian sistem dengan memasukkan gangguan-gangguan yang mungkin terjadi.
4. Memperbaiki dan membandingkan sistem dengan pengontrol Fuzzy atau gabungan antara Fuzzy dan PID.
5. Pada *software* dapat dikembangkan lagi agar dapat menganalisis ketinggian air lebih tepat sehingga lebih sistem secara keseluruhan akan lebih lengkap.

LAMPIRAN





Gambar Rangkaian Lengkap

// Adapted from Max Seim's BV sample mlseim@mmm.com converted to Delphi5 and added a second type of PID algorithm.

// Andrew R Clayton 12-30-07 AndrewClayton@carolina.rr.com

// syaiful M 2-9-2008 Cukup_ipul_067@yahoo.co.id

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, ExtCtrls, ComCtrls, Spin, Gauges,
Series, Chart, Buttons, SmallPort, TeEngine, TeeProcs, OleCtrls, VCFL,
DB, DBTables;

type

TForm1 = class(TForm)

VScroll1: TScrollBar;

Text1: TLabel;

StaticText1: TStaticText;

HScroll2: TScrollBar;

GainSpinEdit: TSpinEdit;

IntegralResetSpinEdit: TSpinEdit;

DerivativeRateSpinEdit: TSpinEdit;

StaticText2: TStaticText;

StaticText3: TStaticText;

StaticText4: TStaticText;

HScroll1: TScrollBar;

Label12: TLabel;

Label30: TLabel;

Gauge1: TGauge;

Gauge2: TGauge;

VtChart1: TVtChart;

```

StaticText5: TStaticText;
Shape1: TShape;
Shape2: TShape;
Shape5: TShape;
Shape6: TShape;
Shape7: TShape;
Gauge3: TGauge;
Gauge4: TGauge;
Button2: TButton;
Chart1: TChart;
Series1: TLineSeries;
Panel1: TPanel;
PompaAir: TButton;
Solenoid: TButton;
Edit1: TEdit;
Button1: TButton;
Panel2: TPanel;
Button3: TButton;
Memo1: TMemo;
Button4: TButton;
SmallPort1: TSmallPort;
Memo2: TMemo;
Button3TimerNew: TButton;
TimerNew: TTimer;
Edit2: TEdit;
Memo3: TMemo;
Memo4: TMemo;
Memo5: TMemo;
Label1: TLabel;

```



```

Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
LabelTinggiAir: TLabel;
Label8: TLabel;
Panel3: TPanel;
Image1: TImage;
procedure HScroll1Change(Sender: TObject);
procedure VScroll1Change(Sender: TObject);
procedure HScroll2Change(Sender: TObject);
procedure strobe1;
procedure strobe2;
procedure PID(Sp,Pv:single; Var output : single);
procedure PID2(Sp,Pv:single; Var output : single);
procedure Tank();
Procedure Chart();
procedure Button2Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure PompaAirClick(Sender: TObject);
procedure SolenoidClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
function CariTinggiAir(Data1: integer): integer;
procedure HitungPVAir;
procedure Edit2KeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
procedure ProsesTinggiAir(DataVisual: byte);
procedure ProsesTunda(Lama: longint);
procedure TimerNewTimer(Sender: TObject);

```

```

procedure Button3TimerNewClick(Sender: TObject);
procedure ProsedurSolenoid(Persen: integer);
procedure BikinLabelChart(Sender: TObject);
private
    { Private declarations }
public
procedure tulis8255(port,data:byte);
    { Public declarations }
end;
const
PA=0;
PB=1;
PC=2;
PCW=3;
invalve : Single = 1.00;
outvalve : Single = 1.00;
mode   : integer = 1;
error  : integer = 1;
supply : integer = 1;
stability : integer = 1;
x      : integer = 1;
y      : integer = 1;
gain   : single = 1.00; // Proportional Gain
reset  : integer = 1; // Intregal reset
rate   : integer = 1; // Derivative rate
output : single = 1.00;
PV     : Single = 10.00;
s1     : integer = 1;
s2     : integer = 1;

```



```

inputD : single = 1.00;
inputDF : single = 1.00;
inputlast : single = 1.00;
feedbackfilter : integer = 1;
feedback : Single = 1.00;
SP : integer = 1;
PV_DIV2: integer = 1550;
dfilter : integer = 10;
lpt_data=$378;
lpt_status=$379;
lpt_kontrol=$37A;
var
  Form1: TForm1;
  wr1,wr0:byte;
  a,tampung:byte;
  i:integer;
  selesai:boolean
var TinggiAir: single;
  hh, hhh: integer;
  dataPort,dataVisual:byte;
  chart1ADC: integer;
  AOwner: TComponent;
  Lab: array[1..30] of TLabel;
  procedure Out32(waddr:word;bOut:byte);stdcall;external'inpout32.dll'
implementation
  procedure tform1.tulis8255(port,data:byte);
begin
case port of

```



```

PA:
begin
wr1:=$08;
wr0:=$09;
end;

PB:
begin
wr1:=$0C;
wr0:=$0D;
end;

PC:
begin
wr1:=$00;
wr0:=$01;
end;

PCW:
begin
wr1:=$04;
wr0:=$05;
end;
end;

Out32($37A,wr1);
Out32($37A,wr0);
Out32($378,data);
Out32($37A,wr1);
end;

procedure delay(lama:longint);
var
ref:longint;

```



```

begin
ref:=GetTickCount;
repeat
Application.ProcessMessages;
until((GetTickCount-ref)>=lama)
end;

procedure TForm1.strobe1;
begin
tampung:=SmallPort1.ReadByte(lpt_kontrol);
tampung:=tampung or $01;
SmallPort1.WriteByte(lpt_kontrol,tampung);
end;

procedure TForm1.strobe2;
begin
tampung:=SmallPort1.ReadByte(lpt_kontrol);
tampung:=tampung and $FE;
SmallPort1.WriteByte(lpt_kontrol,tampung);
end;
{$R *.DFM}

procedure TForm1.HScroll1Change(Sender: TObject);
begin
Invalve := (Hscroll1.position * (supply /100))/60;
Gauge3.progress := Hscroll1.position;
If Gauge3.progress = 0 then shape5.brush.color := clwhite else
    shape5.brush.color := claqua;
If Gauge3.progress = 0 then shape2.brush.color := clwhite else
    shape2.brush.color := claqua;
end;

```

```

procedure TForm1.VScroll1Change(Sender: TObject);
begin
Label12.caption := inttostr( (VScroll1.position - (PV_DIV2 * 2)) );
Gauge2.progress := (PV_DIV2 * 2) - VScroll1.position ;
Label12.caption := inttostr( (VScroll1.position - (PV_DIV2 * 2)) *-1);
sp := VScroll1.position;
end;
procedure TForm1.HScroll2Change(Sender: TObject);
begin
Outvalve := (HScroll2.position * 30)/60 ;
    Gauge4.progress := Hscroll2.position;
If Gauge4.progress = 0 then shape6.brush.color := clwhite else
    shape6.brush.color := claqua;
end;
procedure TForm1.Tank();
Begin
    Gauge1.Progress := round(PV);
End;
Procedure TForm1.Chart();
Var
Row: Integer;
temp : widestring;
Begin
VtChart1.repaint := false;
With Form1.VtChart1
Do
    VtChart1.Column := 1;
    VtChart1.Row := 24 ;
    VtChart1.Data := inttostr(Round(100- Output)) ;

```

For row := 1 to 26 Do

Begin

VtChart1.Column := 1;

VtChart1.row := (row + 1);

temp := VtChart1.Data;

VtChart1.row := row;

VtChart1.Data := Temp;

Edit1.Text := temp;

Lab[row].Caption := inttostr(row+hhh);

end;

VtChart1.repaint := true;

End;

procedure TForm1.PID(Sp,Pv:single; Var output : single); // no feedback

Begin

dfilter := 10; // Filter value to scale down derivative effect.

inputd := pv + (inputlast - pv) * (rate / 60); // Derivative Rate 0 to 120

inputlast := pv ;

inputdf := inputdf + (inputd - inputdf) * dfilter / 60 ;

output := (sp - inputdf) * (gain / 100) + feedback; // Proportional Gain

If output > 100 Then output := 100 ; // clamp output valve between 0 and 100%

If output < 0 Then output := 0 ;

feedback := feedback - (feedback - output) * reset / 60; // Integral Reset

End;

procedure TForm1.PID2(Sp,Pv:single; Var output : single); // no feedback

Const

```

ErrorLast : Single = 1.00;
ErrorlastLast : Single = 1.00;
Begin
if hhh = 1 then
begin
    ErrorLastLast := 1;
    ErrorLast := 1;
end;
Memo4.Lines.Append('PID2 - Gain -> ' + floattostr(Gain));
Memo4.Lines.Append('PID2 - Reset -> ' + floattostr(Reset));
Memo4.Lines.Append('PID2 - Rate -> ' + floattostr(Rate));
Memo4.Lines.Append('PID2 - Error -> ' + floattostr(Error));
Memo4.Lines.Append('PID2 - ErrorLast -> ' + floattostr(ErrorLast));
Memo4.Lines.Append('PID2 - ErrorLastLast -> ' + floattostr(ErrorLastLast));
output := Output + (Gain/100) * (Error - ErrorLast + (Reset / 60) * Error + (Rate/60) * (error - ErrorLast
* 2 + ErrorLastLast));
ErrorLastLast := ErrorLast;
ErrorLast := Error;
If output > 100 Then output := 100 ;// clamp output valve between 0 and 100%
If output < 0 Then output := 0 ;
End;
procedure TForm1.Button2Click(Sender: TObject);
begin
    selesai:=true;
    tulis8255(PC,$00);
    tulis8255(PB,$00);
    Shape1.Brush.Color:=clwhite;
    HScroll1.Position:=0;
    HScroll2.Position:= 0 ;

```

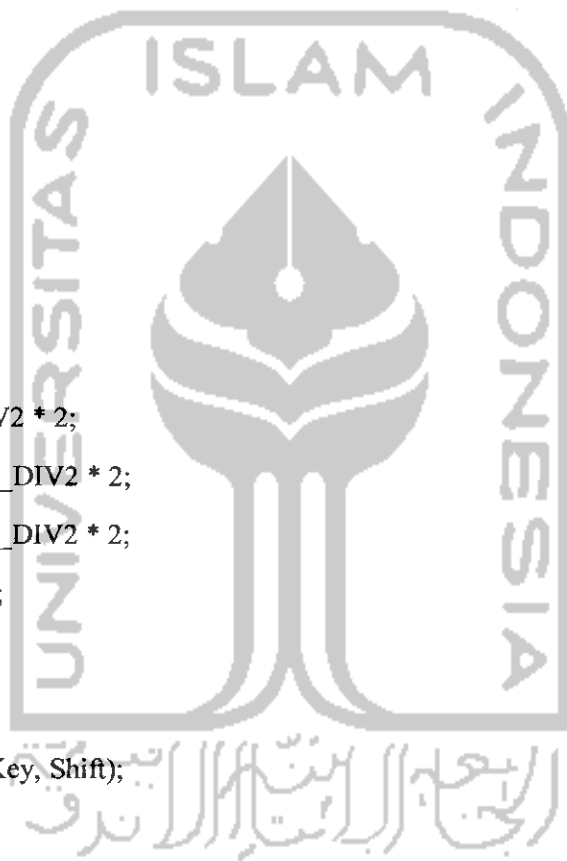
```

end;

procedure TForm1.FormCreate(Sender: TObject);
var Shift: TShiftState;
    Key: word;
    x: integer;
begin
    tulis8255(PCW,$80);
    tulis8255(PB,$FF);
    tulis8255(PC,$00);
    tulis8255(PA,$ff);
    HScroll2.Position := 0;
    Text1.Caption := "";
    VScroll1.Max := PV_DIV2 * 2;
    Gauge2.MaxValue := PV_DIV2 * 2;
    Gauge1.MaxValue := PV_DIV2 * 2;
    BikinLabelChart(Sender);
    Edit2.Text := '255';
    Key := 13;
    Edit2KeyDown(Sender, Key, Shift);
    for x := 1 to 26 do
        Lab[x].Caption := inttostr(x);
    Gauge1.Progress := 0;
    Edit2.Clear;
end;

procedure TForm1.PompaAirClick(Sender: TObject);
var
    dd: TDateTime;
    tt: TTime;
begin

```



```

tulis8255(PB,$00);
TimerNew.Enabled := true;
chart1ADC := 0;
dd := Date;
tt := Time;
Memo2.Lines.Append(Caption + ' - Mulai');
Memo2.Lines.Append('SP    PV    Output    100-Output    invalve    outvalve');
Memo1.Lines.Append("");
Memo1.Lines.Append(DateToStr(dd) + ' ' + TimeToStr(tt));
Memo1.Lines.Append('MENGENGI TANGKI');
begin
Label12.Caption := inttostr(VScroll1.Position);
supply := 2000;
HScroll1.Position := 100;
invalve := (HScroll1.Position * (supply / 100)) / 60 ;
outvalve := 0 ;
Text1.Caption := inttostr(Round( (PV_DIV2 * 2) -pv));
mode := 1;
HScroll2.Enabled := true;
gain := GainSpinedit.value;
reset := IntegralResetSpinedit.value;
rate := DerivativeRateSpinedit.value;
Label12.Caption := inttostr( VScroll1.Position);
sp := VScroll1.Position;
Shape2.Brush.Color:=clAqua;
Shape1.Brush.Color:=clAqua;
selesai:=false;
Series1.Clear;

```

```

end;

end;

procedure TForm1.SolenoidClick(Sender: TObject);
begin
    tulis8255(PC,$00);
    tulis8255(PA,$64);
end;

procedure TForm1.Button1Click(Sender: TObject);
var
    Nm: string;
    dd: TDateTime;
    tt: TTime;
begin
    dd := Date;
    tt := Time;
    Nm := DateToStr(dd) + ' ' + TimeToStr(tt);
    Memo1.Lines.Append("");
    Memo1.Lines.Append(DateToStr(dd) + ' ' + TimeToStr(tt));
    tulis8255(PC,$ff);
    tulis8255(PA,$00);
    Application.Terminate;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    tulis8255(PB,$FF);
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
    Memo1.Lines.Append("");

```

```

Memo1.Lines.Append('MENGOSONGKAN TANGKI');
Memo2.Lines.Append("");
Memo2.Lines.Append('MENGOSONGKAN TANGKI');
tulis8255(PC,$00);
tulis8255(PA,$64);
end;
function TForm1.CariTinggiAir(Data1: integer):integer;
var Hasil: integer;
begin
    Hasil :=0;
    if (Data1 >= 6) and (Data1 <= 13) then Hasil := 0;
    if (Data1 >= 14) and (Data1 <= 21) then Hasil :=1;
    if (Data1 >= 22) and (Data1 <= 30) then Hasil := 2;
    if (Data1 >= 31) and (Data1 <= 38) then Hasil := 3;
    if (Data1 >= 39) and (Data1 <= 45) then Hasil := 4;
    if (Data1 >= 46) and (Data1 <= 52) then Hasil := 5;
    if (Data1 >= 53) and (Data1 <= 60) then Hasil := 6;
    if (Data1 >= 61) and (Data1 <= 67) then Hasil := 7;
    if (Data1 >= 68) and (Data1 <= 74) then Hasil := 8;
    if (Data1 >= 75) and (Data1 <= 81) then Hasil := 9;
    if (Data1 >= 82) and (Data1 <= 89) then Hasil := 10;
    if (Data1 >= 90) and (Data1 <= 97) then Hasil := 11;
    if (Data1 >= 98) and (Data1 <= 104) then Hasil := 12;
    if (Data1 >= 105) and (Data1 <= 111) then Hasil := 13;
    if (Data1 >= 112) and (Data1 <= 118) then Hasil := 14;
    if (Data1 >= 119) and (Data1 <= 126) then Hasil := 15;
    if (Data1 >= 127) and (Data1 <= 134) then Hasil := 16;
    if (Data1 >= 135) and (Data1 <= 143) then Hasil := 17;
    if (Data1 >= 144) and (Data1 <= 151) then Hasil := 18;

```

```

if (Data1 >= 152) and (Data1 <= 159) then Hasil := 19;
if (Data1 >= 160) and (Data1 <= 166) then Hasil := 20;
if (Data1 >= 167) and (Data1 <= 176) then Hasil := 21;
if (Data1 >= 181) and (Data1 <= 187) then Hasil := 22;
if (Data1 >= 188) and (Data1 <= 196) then Hasil := 23;
if (Data1 >= 197) and (Data1 <= 205) then Hasil := 24;
if (Data1 >= 206) and (Data1 <= 214) then Hasil := 25;
if (Data1 >= 215) and (Data1 <= 222) then Hasil := 26;
if (Data1 >= 223) and (Data1 <= 229) then Hasil := 27;
if (Data1 >= 230) and (Data1 <= 238) then Hasil := 28;
if (Data1 >= 239) and (Data1 <= 247) then Hasil := 29;
if (Data1 >= 248) and (Data1 <= 255) then Hasil := 30;
Result := hasil;
end;
// ----- BARU 23-FEB-2009 -----
procedure TForm1.Edit2KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
if Key = 13 then
begin
  dataPort := strtoint(edit2.Text);
  case dataPort of
    0 : begin
      Gauge1.Progress := 0;
      dataVisual := 0;
      end;
    244..255:
      dataVisual := 81;
    else

```

```

    dataVisual := (dataPort) div 3;
end;
Label2.Caption := IntToStr(dataPort); // DataPort Asli
Label4.Caption := IntToStr(dataVisual); // DataPort Visual
ProsesTinggiAir(DataVisual);
// Ketinggian Air
LabelTinggiAir.Caption := 'Ketinggian Air = ' + IntToStr(CariTinggiAir(dataPort)) + ' Cm';
// Grafik ADC
with Chart1 do
    with Series1 do
        addxy(chart1ADC,dataPort,"clBlue");
        inc(chart1ADC);
        ProsedurSolenoid(error);
end;
end;
procedure TForm1.HitungPVAir;
Const
Toggle : boolean = false;
begin
    // 16-02-2009 23-15-15
    hhh := hhh + 1;
    Toggle := not toggle;
    sp := (PV_DIV2 * 2) - VScroll1.Position;
    Gain := GainSpinEdit.value; // 0 to 100
    Reset := IntegralResetSpinEdit.value; // 0 to 120 secs
    Rate := DerivativeRateSpinEdit.value; // 0 to 120 secs

    If pv < (PV_DIV2 * 2) Then pv := pv + invalve ; // add and subtract product to tank

```



```

If pv > 0 Then pv := pv - outvalve;
Text1.Caption := IntToStr(Round( pv));
error := round(sp - pv) ;
Memo4.Lines.Append('Pv -> ' + floattostr(pv));
Memo4.Lines.Append('error -> ' + floattostr(error));
Label30.Caption := inttostr(error);
PID2(Sp,Pv, output); // feedback
HScroll2.Position := 100 - round( output) ;// Change slider value (AUTO MODE)
Memo4.Lines.Append('HitungAirAqua - HScroll2.Position -> ' + inttostr(HScroll2.Position));
Tank();
If not toggle then Chart(); // reduce flicker
Memo3.Lines.Append(inttostr(hhh) + ' ' + floattostr(Sp) + ' ' + floattostr(Pv) + ' ' + floattostr(Output)
+ ' ' + floattostr(100- Output) + ' ' + floattostr(invalve) + ' ' + floattostr(outvalve));

Memo4.Lines.Append(inttostr(hhh) + ' ' + floattostr(output) + ' ' + floattostr(invalve) + ' ' +
floattostr(outvalve) + ' ' + inttostr(error) + ' ' + inttostr(supply) + ' ' + floattostr(gain) + ' ' +
inttostr(reset) + ' ' + floattostr(rate) + ' ' + floattostr(PV) + ' ' + floattostr(inputD) + ' ' +
floattostr(inputDF) + ' ' + floattostr(inputlast) + ' ' + floattostr(feedback) + ' ' + inttostr(SP) + ' ' +
inttostr(dfilter) + ' ' + inttostr(HScroll1.position) + ' ' + inttostr(HScroll2.position) + ' ' +
inttostr(VScroll1.position));
Memo4.Lines.Append("");
end;
procedure TForm1.ProsesTinggiAir(DataVisual: byte);
var u: byte;
begin
Memo4.Clear;
Memo3.Clear;
HScroll2.Enabled := true;
HScroll1.Position := 100;
supply := 2000;
invalve := (HScroll1.Position * (supply / 100)) / 60 ;

```

```

outvalve := 0 ;
gain := GainSpinedit.value;
reset := IntregalResetSpinedit.value;
rate := DerivativeRateSpinedit.value;
sp := VScroll1.Position;
hhh := 0;
HScroll2.Position:=0;
PV    := 1.00;
output := 1.00;
inputD := 1.00;
inputDF := 1.00;
inputlast := 1.00;
feedback := 1.00;
for u := 1 to DataVisual do
begin
    HitungPVAir;
    ProsesTunda(30000);
end;
end;
procedure TForm1.ProsesTunda(Lama: longint);
var u: longint;
begin
    for u := 1 to Lama do
    begin
        Application.ProcessMessages;
    end;
end;
procedure TForm1.TimerNewTimer(Sender: TObject);
begin

```



```

Panel2.Caption := DateTimeToStr(now) ;
TimerNew.Enabled := false;
strobel;
a:=SmallPort1.ReadByte(lpt_status)and $F0;
a:=a shr 4;
strobe2;
a:=a or (SmallPort1.ReadByte(lpt_status) and $F0);
dataPort:=a xor $88;
// coz: saat DataPort 81, nilai PV sudah 1550
// padahal PV 1550 harusnya DataPort 255, maka  $255 / 81 = 3.148$ 
case dataPort of
244..255:
    dataVisual := 81;
else
    dataVisual := (dataPort) div 3;
end;
Label2.Caption := IntToStr(dataPort); // DataPort Asli
Label4.Caption := IntToStr(dataVisual); // DataPort Visual
edit2.Text := inttostr(dataPort);
ProsesTinggiAir(DataVisual);
ProsesTunda(50000);
// Ketinggian Air
LabelTinggiAir.Caption := 'Ketinggian Air = ' + IntToStr(CariTinggiAir(dataPort)) + ' Cm';
// Grafik ADC
Panel1.Caption:=IntToStr(dataPort);
with Chart1 do
    with Series1 do
        addxy(chart1ADC,dataPort,"clBlue");
        inc(chart1ADC);

```

```

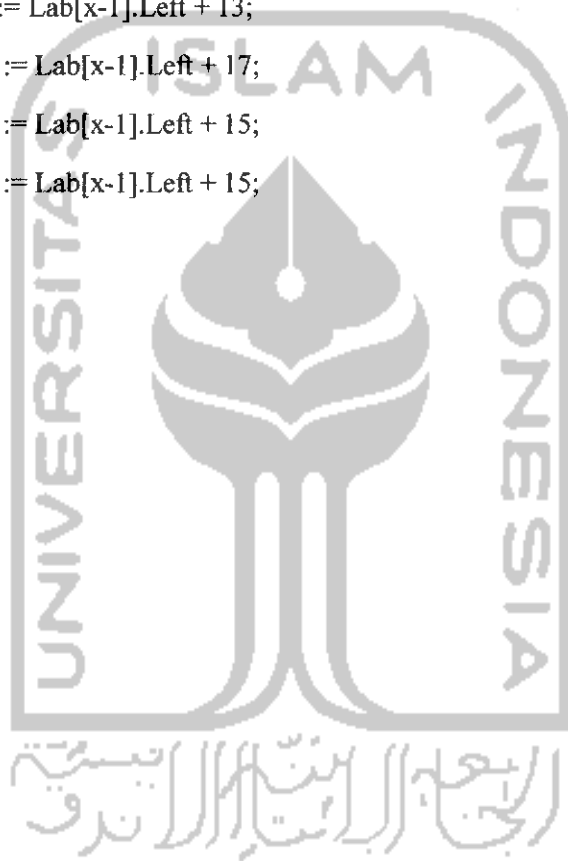
// ProsedurSolenoid
ProsedurSolenoid(error);
TimerNew.Enabled := true;
end;
procedure TForm1.Button3TimerNewClick(Sender: TObject);
begin
    TimerNew.Enabled := not TimerNew.Enabled;
end;
procedure TForm1.ProsedurSolenoid(Persen: integer);
var heksa: byte;
begin
    case Persen of
        0 : heksa := $00;
        1..1549 : heksa := $ff;
    end;
    Label8.Caption := IntToStr(Persen) + ' %';
    tulis8255(PC,$00);
    tulis8255(PA,heksa);
end;
procedure TForm1.BikinLabelChart(Sender: TObject);
var x: integer;
begin
    for x := 1 to 26 do
    begin
        Lab[x] := TLabel.Create(AOwner);
        Lab[x].Parent := Panel3;
        Lab[x].Font.Name := 'MS Serif';
        Lab[x].Font.Size := 6;
        Lab[x].Top := 0;
    end;
end;

```

```

if x = 1 then
  Lab[x].Left := 86//102
else
  begin
    case x of
      1..10 : Lab[x].Left := Lab[x-1].Left + 13;
      11..15 : Lab[x].Left := Lab[x-1].Left + 17;
      16..20 : Lab[x].Left := Lab[x-1].Left + 15;
      21..30 : Lab[x].Left := Lab[x-1].Left + 15;
    end;
  end;
end;
end;
end.

```



DAC0808 8-Bit D/A Converter

General Description

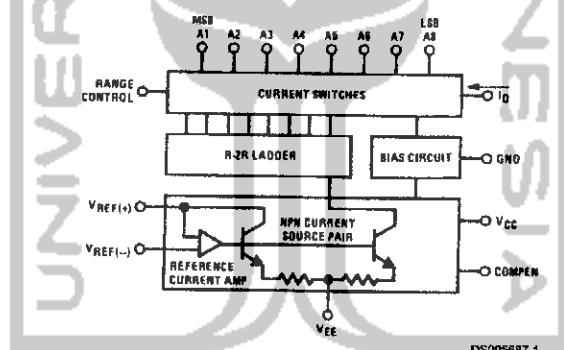
The DAC0808 is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with $\pm 5V$ supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ± 1 LSB of $255 I_{REF}/256$. Relative accuracies of better than $\pm 0.19\%$ assure 8-bit monotonicity and linearity while zero level output current of less than $4 \mu A$ provides 8-bit zero accuracy for $I_{REF} \geq 2$ mA. The power supply currents of the DAC0808 is independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

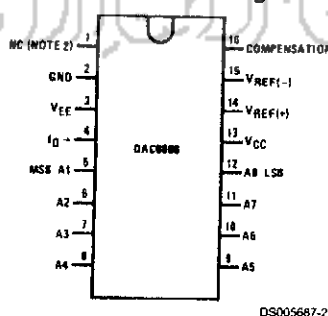
Features

- Relative accuracy: $\pm 0.19\%$ error maximum
- Full scale current match: ± 1 LSB typ
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/ μs
- Power supply voltage range: $\pm 4.5V$ to $\pm 18V$
- Low power consumption: 33 mW @ $\pm 5V$

Block and Connection Diagrams



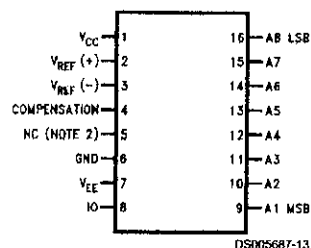
Dual-In-Line Package



Top View
Order Number DAC0808
See NS Package M16A or N16A

Block and Connection Diagrams (Continued)

Small-Outline Package



Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	N PACKAGE (N16A) (Note 1)		SO PACKAGE (M16A)
8-bit	$0^{\circ}\text{C} \leq T_A \leq +75^{\circ}\text{C}$	DAC0808LCN	MC1408P8	DAC0808LCM

Note 1: Devices may be ordered by using either order number.

E

Absolute Maximum Ratings (Note 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Power Supply Voltage

 V_{CC} +18 V_{DC} V_{EE} -18 V_{DC}Digital Input Voltage, V₅-V₁₂ -10 V_{DC} to +18 V_{DC}Applied Output Voltage, V_O -11 V_{DC} to +18 V_{DC}Reference Current, I₁₄ 5 mAReference Amplifier Inputs, V₁₄, V₁₅ V_{CC}, V_{EE}

Power Dissipation (Note 4) 1000 mW

ESD Susceptibility (Note 5) TBD

Storage Temperature Range

-65°C to +150°C

Lead Temp. (Soldering, 10 seconds)

Dual-In-Line Package (Plastic) 260°C

Dual-In-Line Package (Ceramic) 300°C

Surface Mount Package

Vapor Phase (60 seconds) 215°C

Infrared (15 seconds) 220°C

Operating Ratings

Temperature Range

 $T_{MIN} \leq T_A \leq T_{MAX}$

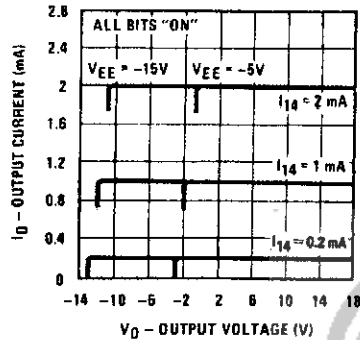
DAC0808

 $0 \leq T_A \leq +75^\circ\text{C}$ **Electrical Characteristics**(V_{CC} = 5V, V_{EE} = -15 V_{DC}, V_{REF}/R₁₄ = 2 mA, and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
E _r	Relative Accuracy (Error Relative to Full Scale I _O)	(Figure 4)				%
	DAC0808LC (LM1408-8)				±0.19	%
	Settling Time to Within ½ LSB (includes t _{PLH})	T _A = 25°C (Note 7), (Figure 5)		150		ns
t _{PLH} , t _{PHL}	Propagation Delay Time	T _A = 25°C, (Figure 5)		30	100	ns
TCI _O	Output Full Scale Current Drift			±20		ppm/°C
MSB	Digital Input Logic Levels	(Figure 3)				
V _{IH}	High Level, Logic "1"		2			V _{DC}
V _{IL}	Low Level, Logic "0"				0.8	V _{DC}
MSB	Digital Input Current	(Figure 3)				
	High Level	V _{IH} = 5V		0	0.040	mA
	Low Level	V _{IL} = 0.8V		-0.003	-0.8	mA
15	Reference Input Bias Current	(Figure 3)		-1	-3	µA
Typ	Output Current Range	(Figure 3)				
		V _{EE} = -5V	0	2.0	2.1	mA
Logic Input	Output Current	V _{REF} = 2.000V, R ₁₄ = 1000Ω, (Figure 3)	0	2.0	4.2	mA
	Output Current, All Bits Low	(Figure 3)	1.9	1.99	2.1	mA
I _L - LOGIC INPUT CURRENT (µA)	Output Voltage Compliance (Note 3)	E _r ≤ 0.19%, T _A = 25°C				
	V _{EE} = -5V, I _{REF} = 1 mA				-0.55, +0.4	V _{DC}
	V _{EE} Below -10V				-5.0, +0.4	V _{DC}
R _{IREF}	Reference Current Slew Rate	(Figure 6)	4	8		mA/µs
	Output Current Power Supply Sensitivity	-5V ≤ V _{EE} ≤ -16.5V		0.05	2.7	µA/V
	Power Supply Current (All Bits Low)	(Figure 3)				
				2.3	22	mA
				-4.3	-13	mA
	Power Supply Voltage Range	T _A = 25°C, (Figure 3)	4.5	5.0	5.5	V _{DC}
			-4.5	-15	-16.5	V _{DC}
	Power Dissipation					

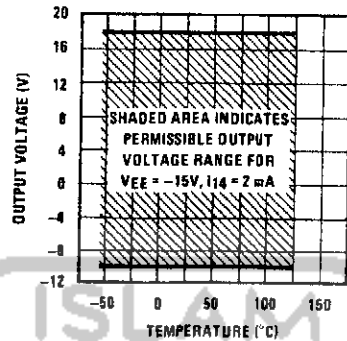
Typical Performance Characteristics $V_{CC} = 5V$, $V_{EE} = -15V$, $T_A = 25^\circ C$, unless otherwise noted (Continued)

Output Current vs Output Voltage (Output Voltage Compliance)



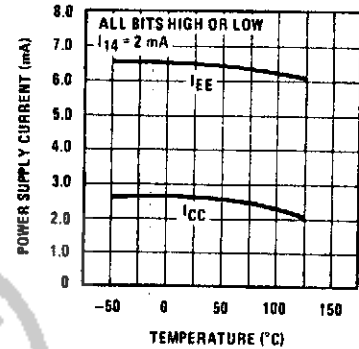
DS005687-17

Output Voltage Compliance vs Temperature



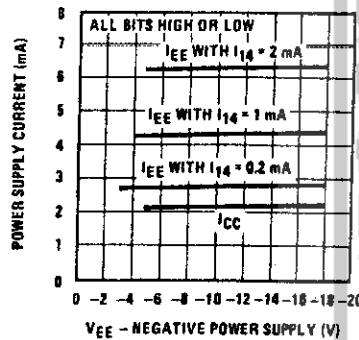
DS005687-18

Typical Power Supply Current vs Temperature



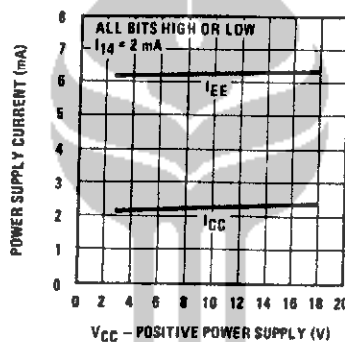
DS005687-19

Typical Power Supply Current vs V_{EE}



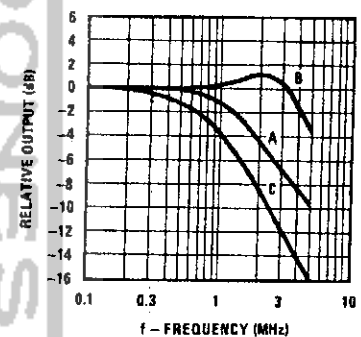
DS005687-20

Typical Power Supply Current vs V_{CC}



DS005687-21

Reference Input Frequency Response



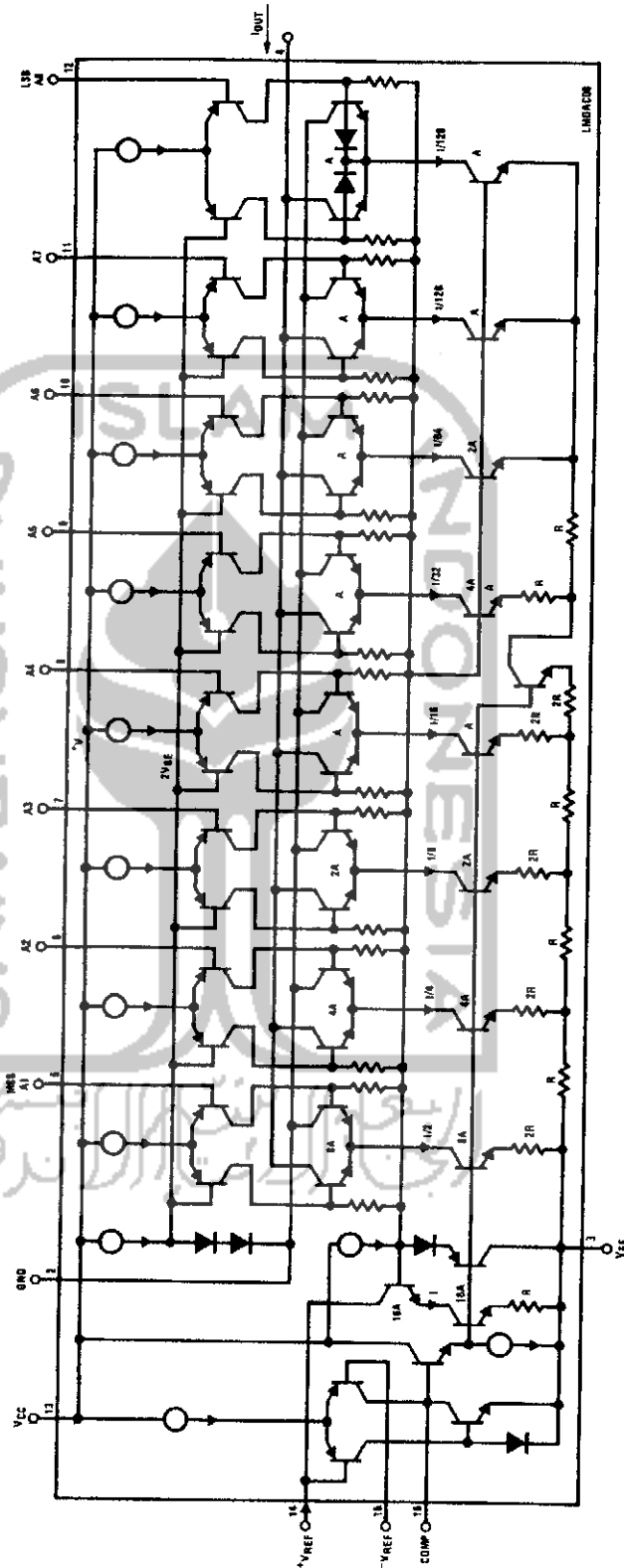
DS005687-22

Unless otherwise specified: $R_{14} = R_{15} = 1\text{ k}\Omega$, $C = 15\text{ pF}$, pin 16 to V_{EE} ; $R_L = 50\Omega$, pin 4 to ground.

Curve A: Large Signal Bandwidth Method of Figure 7, $V_{REF} = 2\text{ Vp-p}$ offset 1V above ground.

Curve B: Small Signal Bandwidth Method of Figure 7, $R_L = 250\Omega$, $V_{REF} = 50\text{ mVp-p}$ offset 200 mV above ground.

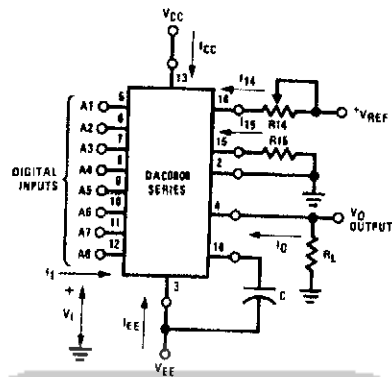
Curve C: Large and Small Signal Bandwidth Method of Figure 9 (no op amp, $R_L = 50\Omega$), $R_S = 50\Omega$, $V_{REF} = 2V$, $V_S = 100\text{ mVp-p}$ centered at 0V.



0805587-4

FIGURE 2. Equivalent Circuit of the DAC0808 Series (Note 8)

Test Circuits



DS005687-B

V_I and I_I apply to inputs A1–A8.

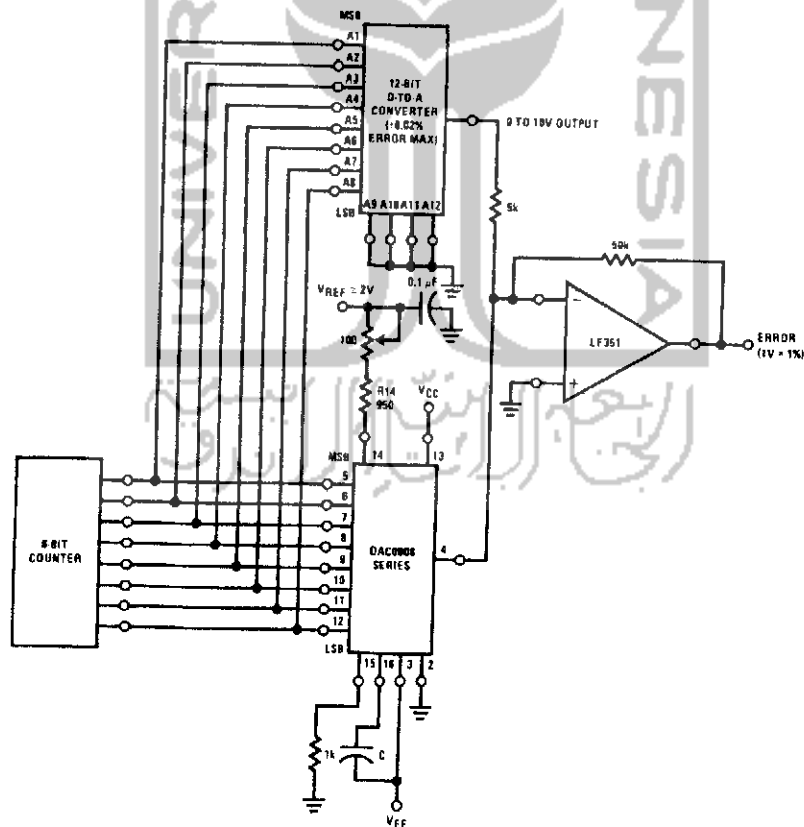
The resistor tied to pin 15 is to temperature compensate the bias current and may not be necessary for all applications.

$$I_O = K \left(\frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right)$$

$$\text{where } K \cong \frac{V_{REF}}{R_{14}}$$

and $A_N = "1"$ if A_N is at high level
 $A_N = "0"$ if A_N is at low level

FIGURE 3. Notation Definitions Test Circuit (Note 8)



DS005687-7

FIGURE 4. Relative Accuracy Test Circuit (Note 8)

Test Circuits (Continued)

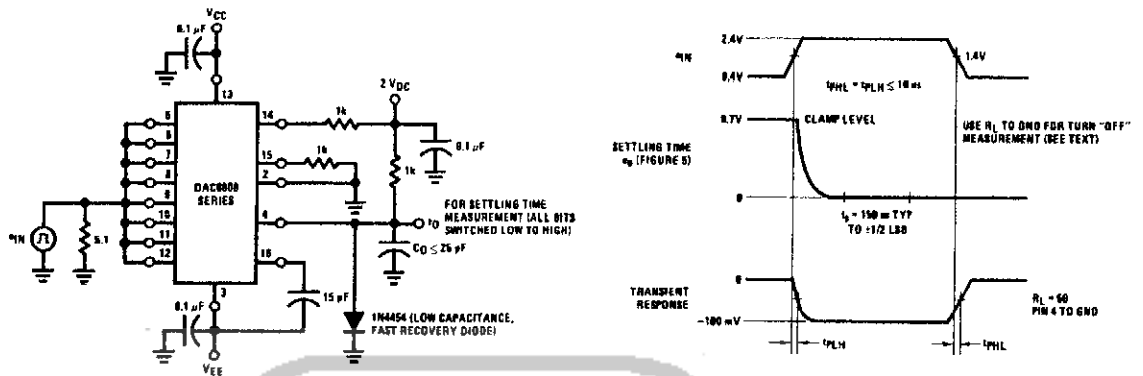


FIGURE 5. Transient Response and Settling Time (Note 8)

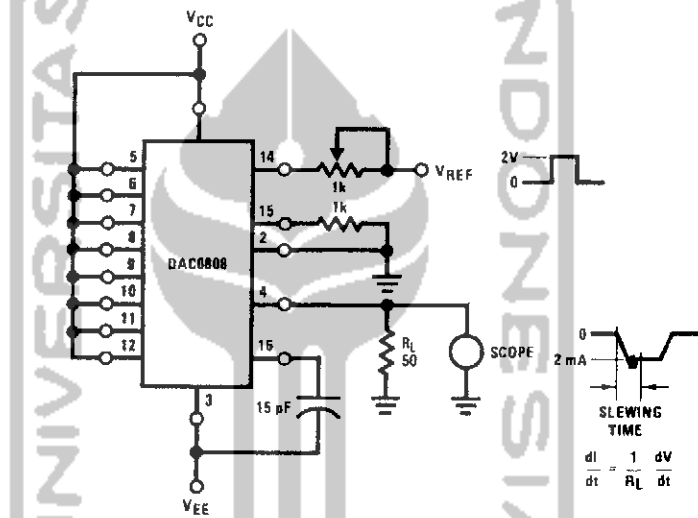
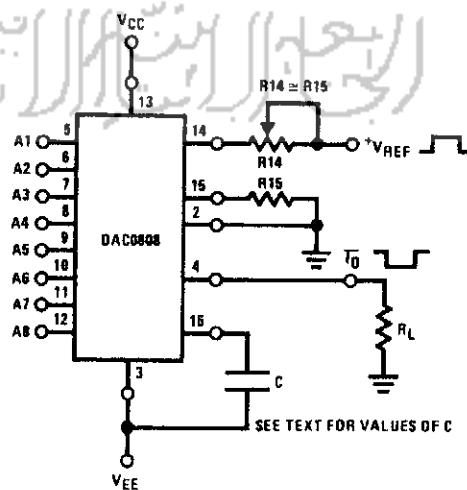
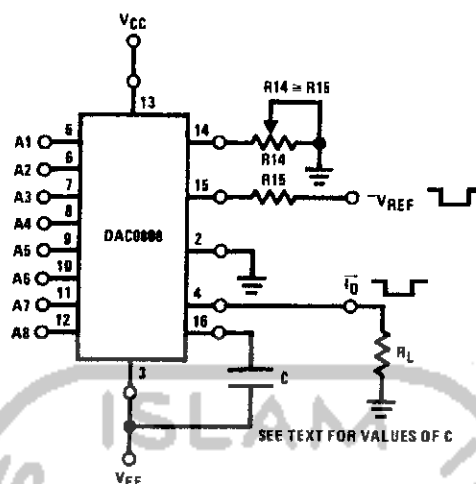


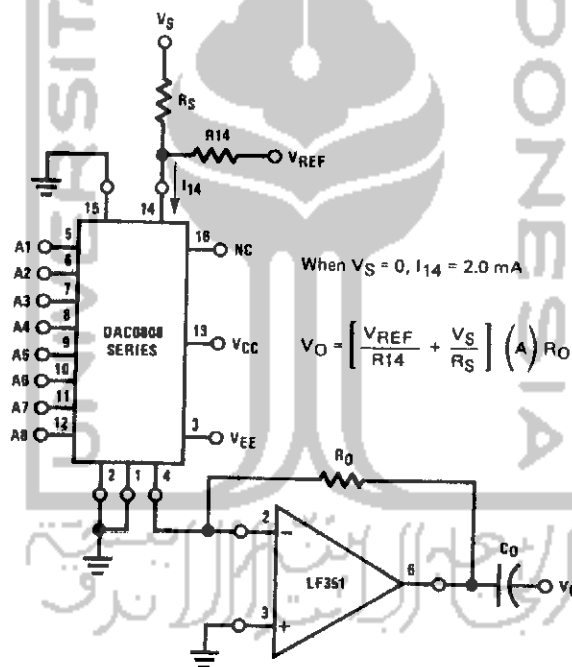
FIGURE 6. Reference Current Slew Rate Measurement (Note 8)

FIGURE 7. Positive V_{REF} (Note 8)

Test Circuits (Continued)



DS005687-11

FIGURE 8. Negative V_{REF} (Note 8)

DS005687-12

FIGURE 9. Programmable Gain Amplifier or Digital Attenuator Circuit (Note 8)

Application Hints

REFERENCE AMPLIFIER DRIVE AND COMPENSATION

The reference amplifier provides a voltage at pin 14 for converting the reference voltage to a current, and a turn-around circuit or current mirror for feeding the ladder. The reference amplifier input current, I_{14} , must always flow into pin 14, regardless of the set-up method or reference voltage polarity.

Connections for a positive voltage are shown in Figure 7. The reference voltage source supplies the full current I_{14} .

For bipolar reference signals, as in the multiplying mode, R_{15} can be tied to a negative voltage corresponding to the minimum input level. It is possible to eliminate R_{15} with only a small sacrifice in accuracy and temperature drift.

The compensation capacitor value must be increased with increases in R_{14} to maintain proper phase margin; for R_{14} values of 1, 2.5 and 5 k Ω , minimum capacitor values are 15, 37 and 75 pF. The capacitor may be tied to either V_{EE} or ground, but using V_{EE} increases negative supply rejection.

A negative reference voltage may be used if R_{14} is grounded and the reference voltage is applied to R_{15} as shown in Figure 8. A high input impedance is the main

Application Hints (Continued)

advantage of this method. Compensation involves a capacitor to V_{EE} on pin 16, using the values of the previous paragraph. The negative reference voltage must be at least 4V above the V_{EE} supply. Bipolar input signals may be handled by connecting R14 to a positive reference voltage equal to the peak positive input level at pin 15.

When a DC reference voltage is used, capacitive bypass to ground is recommended. The 5V logic supply is not recommended as a reference voltage. If a well regulated 5V supply which drives logic is to be used as the reference, R14 should be decoupled by connecting it to 5V through another resistor and bypassing the junction of the 2 resistors with 0.1 μ F to ground. For reference voltages greater than 5V, a clamp diode is recommended between pin 14 and ground.

If pin 14 is driven by a high impedance such as a transistor current source, none of the above compensation methods apply and the amplifier must be heavily compensated, decreasing the overall bandwidth.

OUTPUT VOLTAGE RANGE

The voltage on pin 4 is restricted to a range of -0.55 to 0.4V when $V_{EE} = -5V$ due to the current switching methods employed in the DAC0808.

The negative output voltage compliance of the DAC0808 is extended to -5V where the negative supply voltage is more negative than -10V. Using a full-scale current of 1.992 mA and load resistor of 2.5 k Ω between pin 4 and ground will yield a voltage output of 256 levels between 0 and -4.980V. Floating pin 1 does not affect the converter speed or power dissipation. However, the value of the load resistor determines the switching time due to increased voltage swing. Values of R_L up to 500 Ω do not significantly affect performance, but a 2.5 k Ω load increases worst-case settling time to 1.2 μ s (when all bits are switched ON). Refer to the subsequent text section on Settling Time for more details on output loading.

OUTPUT CURRENT RANGE

The output current maximum rating of 4.2 mA may be used only for negative supply voltages more negative than -8V, due to the increased voltage drop across the resistors in the reference current amplifier.

ACCURACY

Absolute accuracy is the measure of each output current level with respect to its intended value, and is dependent upon relative accuracy and full-scale current drift. Relative accuracy is the measure of each output current level as a fraction of the full-scale current. The relative accuracy of the DAC0808 is essentially constant with temperature due to the excellent temperature tracking of the monolithic resistor ladder.

The reference current may drift with temperature, causing a change in the absolute accuracy of output current. However, the DAC0808 has a very low full-scale current drift with temperature.

The DAC0808 series is guaranteed accurate to within $\pm 1/2$ LSB at a full-scale output current of 1.992 mA. This corresponds to a reference amplifier output current drive to the ladder network of 2 mA, with the loss of 1 LSB (8 μ A) which is the ladder remainder shunted to ground. The input current to pin 14 has a guaranteed value of between 1.9 and 2.1 mA, allowing some mismatch in the NPN current source pair. The accuracy test circuit is shown in Figure 4. The 12-bit converter is calibrated for a full-scale output current of 1.992 mA. This is an optional step since the DAC0808 accuracy is essentially the same between 1.5 and 2.5 mA. Then the DAC0808 circuits' full-scale current is trimmed to the same value with R14 so that a zero value appears at the error amplifier output. The counter is activated and the error band may be displayed on an oscilloscope, detected by comparators, or stored in a peak detector.

Two 8-bit D-to-A converters may not be used to construct a 16-bit accuracy D-to-A converter. 16-bit accuracy implies a total error of $\pm 1/2$ of one part in 65,536 or $\pm 0.00076\%$, which is much more accurate than the $\pm 0.019\%$ specification provided by the DAC0808.

MULTIPLYING ACCURACY

The DAC0808 may be used in the multiplying mode with 8-bit accuracy when the reference current is varied over a range of 256:1. If the reference current in the multiplying mode ranges from 16 μ A to 4 mA, the additional error contributions are less than 1.6 μ A. This is well within 8-bit accuracy when referred to full-scale.

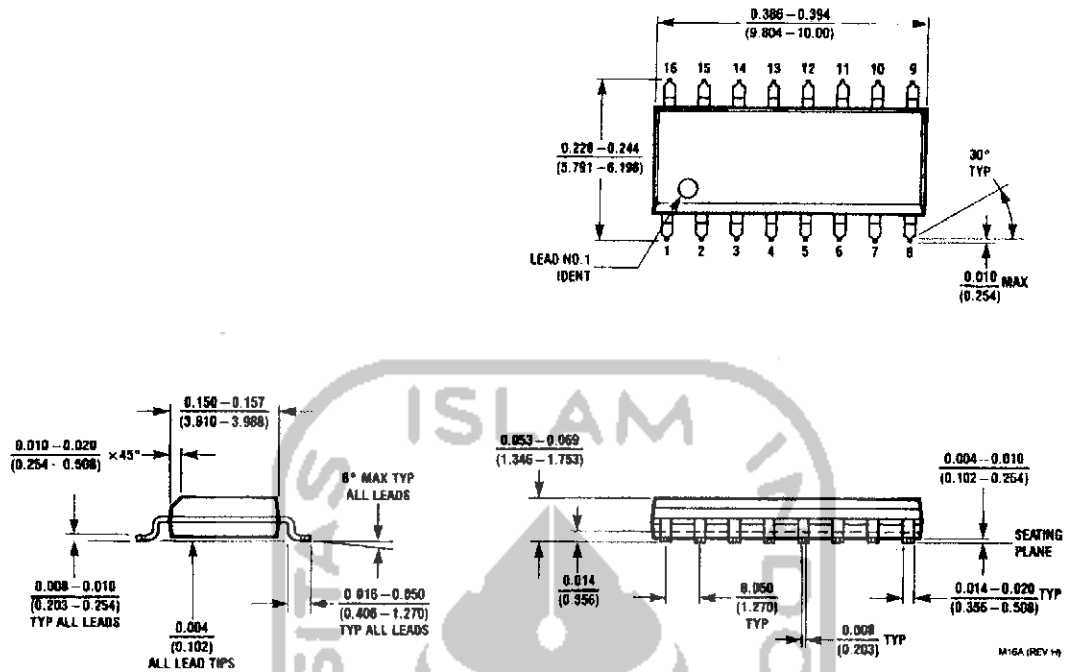
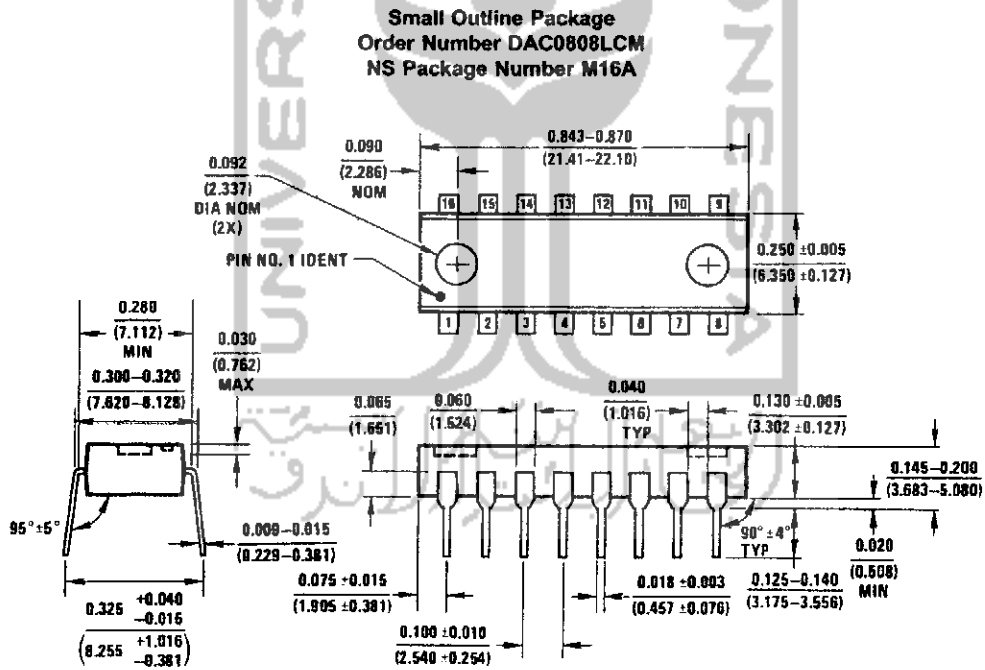
A monotonic converter is one which supplies an increase in current for each increment in the binary word. Typically, the DAC0808 is monotonic for all values of reference current above 0.5 mA. The recommended range for operation with a DC reference current is 0.5 to 4 mA.

SETTLING TIME

The worst-case switching condition occurs when all bits are switched ON, which corresponds to a low-to-high transition for all bits. This time is typically 150 ns for settling to within $\pm 1/2$ LSB, for 8-bit accuracy, and 100 ns to $1/2$ LSB for 7 and 6-bit accuracy. The turn OFF is typically under 100 ns. These times apply when $R_L \leq 500\Omega$ and $C_O \leq 25$ pF.

Extra care must be taken in board layout since this is usually the dominant factor in satisfactory test results when measuring settling time. Short leads, 100 μ F supply bypassing for low frequencies, and minimum scope lead length are all mandatory.

Physical Dimensions inches (millimeters) unless otherwise noted



Notes



LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor Corporation
Americas
Email: support@nsc.com

www.national.com

National Semiconductor Europe

Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

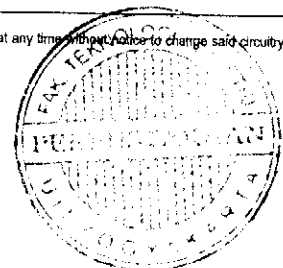
National Semiconductor Asia Pacific Customer Response Group

Tel: 65-2544466
Fax: 65-2504468
Email: ap.support@nsc.com

National Semiconductor Japan Ltd.

Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.

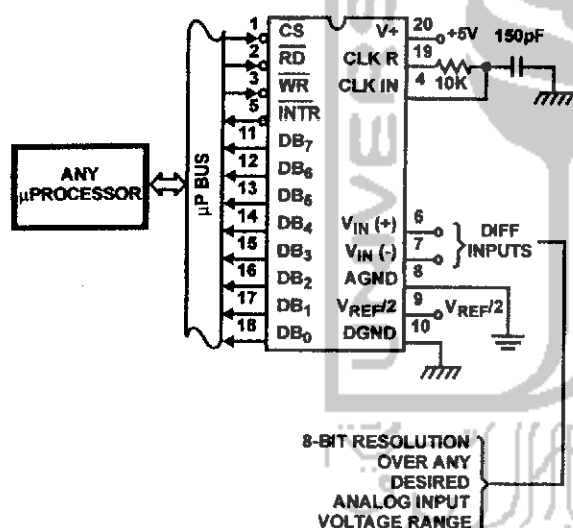


8-Bit, Microprocessor-Compatible, A/D Converters

The ADC080X family are CMOS 8-Bit, successive-approximation A/D converters which use a modified potentiometric ladder and are designed to operate with the 8080A control bus via three-state outputs. These converters appear to the processor as memory locations or I/O ports, and hence no interfacing logic is required.

The differential analog voltage input has good common-mode-rejection and permits offsetting the analog zero-input-voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

Typical Application Schematic

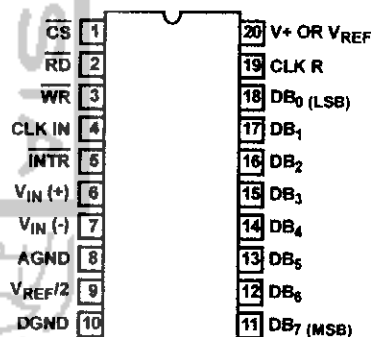


Features

- 80C48 and 80C80/85 Bus Compatible - No Interfacing Logic Required
- Conversion Time <100μs
- Easy Interface to Most Microprocessors
- Will Operate in a "Stand Alone" Mode
- Differential Analog Voltage Inputs
- Works with Bandgap Voltage References
- TTL Compatible Inputs and Outputs
- On-Chip Clock Generator
- Analog Voltage Input Range (Single + 5V Supply) 0V to 5V
- No Zero-Adjust Required
- 80C48 and 80C80/85 Bus Compatible - No Interfacing Logic Required

Pinout

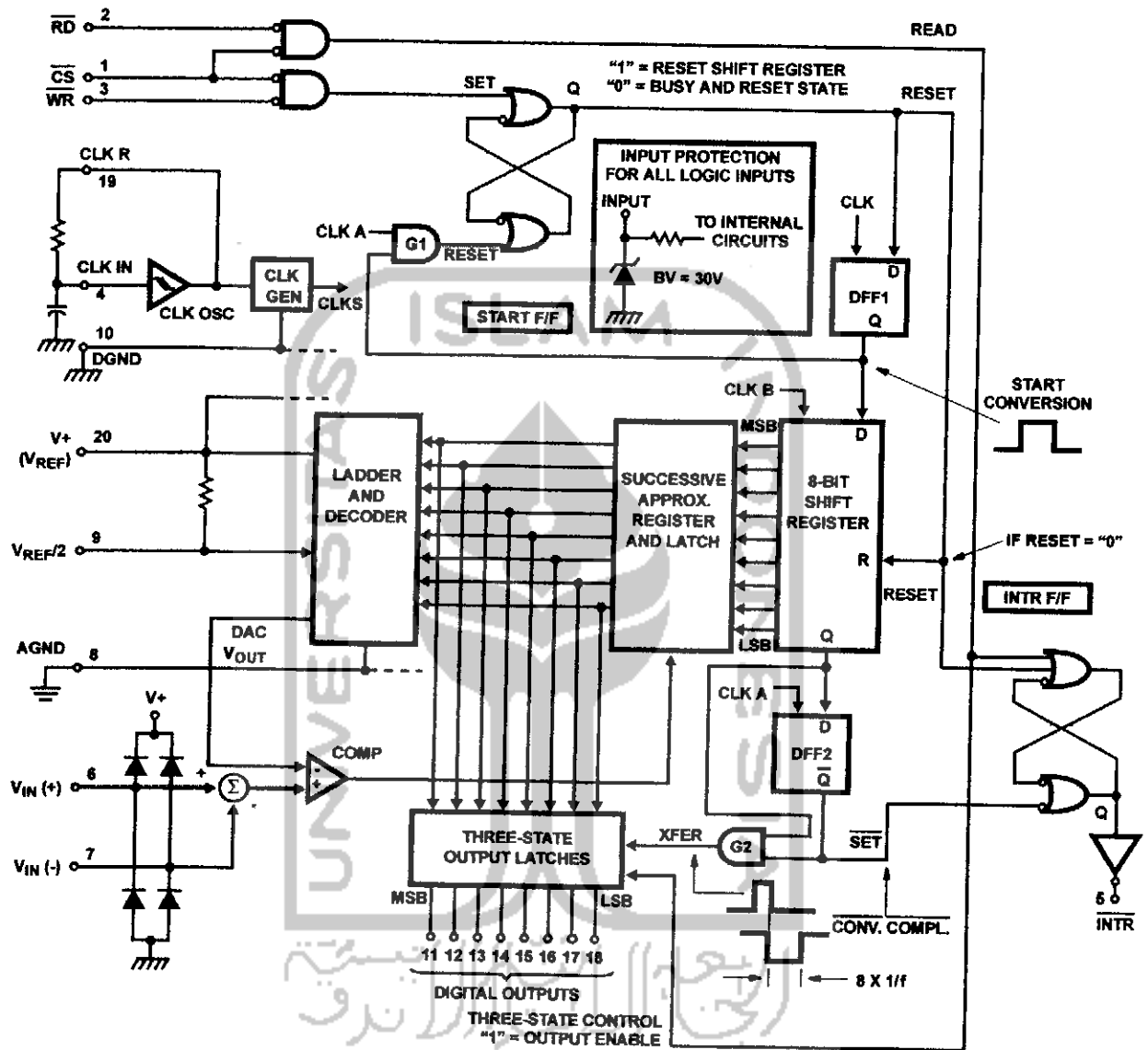
ADC0803, ADC0804
(PDIP)
TOP VIEW



Ordering Information

PART NUMBER	ERROR	EXTERNAL CONDITIONS	TEMP. RANGE (°C)	PACKAGE	PKG. NO
ADC0803LCN	$\pm 1/2$ LSB	$V_{REF/2}$ Adjusted for Correct Full Scale Reading	0 to 70	20 Ld PDIP	E20.3
ADC0804LCN	± 1 LSB	$V_{REF/2} = 2.500V_{DC}$ (No Adjustments)	0 to 70	20 Ld PDIP	E20.3

Functional Diagram



ADC0803, ADC0804

Absolute Maximum Ratings

Supply Voltage 6.5V
Voltage at Any Input -0.3V to ($V^+ + 0.3V$)

Operating Conditions

Temperature Range 0°C to 70°C

Thermal Information

Thermal Resistance (Typical, Note 1) θ_{JA} (°C/W) 80
PDIP Package
Maximum Junction Temperature
Plastic Package 150°C
Maximum Storage Temperature Range -65°C to 150°C
Maximum Lead Temperature (Soldering, 10s) 300°C

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

1. θ_{JA} is measured with the component mounted on a low effective thermal conductivity test board in free air. See Tech Brief TB379 for details.

Electrical Specifications (Notes 2, 8)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
CONVERTER SPECIFICATIONS $V^+ = 5V$, $T_A = 25^\circ C$ and $f_{CLK} = 640kHz$, Unless Otherwise Specified					
Total Unadjusted Error					
ADC0803	$V_{REF/2}$ Adjusted for Correct Full Scale Reading	-	-	$\pm 1/2$	LSB
ADC0804	$V_{REF/2} = 2.500V$	-	-	± 1	LSB
$V_{REF/2}$ Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	k Ω
Analog Input Voltage Range	(Note 3)	GND-0.05	-	$(V^+) + 0.05$	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	$\pm 1/16$	$\pm 1/8$	LSB
Power Supply Sensitivity	$V^+ = 5V \pm 10\%$ Over Allowed Input Voltage Range	-	$\pm 1/16$	$\pm 1/8$	LSB
CONVERTER SPECIFICATIONS $V^+ = 5V$, 0°C to 70°C and $f_{CLK} = 640kHz$, Unless Otherwise Specified					
Total Unadjusted Error					
ADC0803	$V_{REF/2}$ Adjusted for Correct Full Scale Reading	-	-	$\pm 1/2$	LSB
ADC0804	$V_{REF/2} = 2.500V$	-	-	± 1	LSB
$V_{REF/2}$ Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	k Ω
Analog Input Voltage Range	(Note 3)	GND-0.05	-	$(V^+) + 0.05$	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	$\pm 1/8$	$\pm 1/4$	LSB
Power Supply Sensitivity	$V^+ = 5V \pm 10\%$ Over Allowed Input Voltage Range	-	$\pm 1/16$	$\pm 1/8$	LSB
AC TIMING SPECIFICATIONS $V^+ = 5V$, and $T_A = 25^\circ C$, Unless Otherwise Specified					
Clock Frequency, f_{CLK}	$V^+ = 6V$ (Note 4)	100	640	1280	kHz
	$V^+ = 5V$	100	640	800	kHz
Clock Periods per Conversion (Note 5), t_{CONV}		62	-	73	Clocks/Conv
Conversion Rate in Free-Running Mode, CR	INTR tied to WR with $\overline{CS} = 0V$, $f_{CLK} = 640kHz$	-	-	8888	Conv/s
Width of WR Input (Start Pulse Width), $t_{W(WR)}$	$\overline{CS} = 0V$ (Note 6)	100	-	-	ns
Access Time (Delay from Falling Edge of RD to Output Data Valid), t_{ACC}	$C_L = 100pF$ (Use Bus Driver IC for Larger C_L)	-	135	200	ns
Three-State Control (Delay from Rising Edge of RD to HI-Z State), t_{1H} , t_{0H}	$C_L = 10pF$, $R_L = 10K$ (See Three-State Test Circuits)	-	125	250	ns
Delay from Falling Edge of WR to Reset of INTR, t_{WM} , t_{RI}		-	300	450	ns
Input Capacitance of Logic Control Inputs, C_{IN}		-	5	-	pF
Three-State Output Capacitance (Data Buffers), C_{OUT}		-	5	-	pF

ADC0803, ADC0804

Electrical Specifications (Notes 2, 8) (Continued)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
DC DIGITAL LEVELS AND DC SPECIFICATIONS $V^+ = 5V$, and T_{MIN} to T_{MAX} , Unless Otherwise Specified					
CONTROL INPUTS (Note 7)					
Logic "1" Input Voltage (Except Pin 4 CLK IN), V_{INH}	$V^+ = 5.25V$	2.0	-	V^+	V
Logic "0" Input Voltage (Except Pin 4 CLK IN), V_{INL}	$V^+ = 4.75V$	-	-	0.8	V
CLK IN (Pin 4) Positive Going Threshold Voltage, V^+_{CLK}		2.7	3.1	3.5	V
CLK IN (Pin 4) Negative Going Threshold Voltage, V^-_{CLK}		1.5	1.8	2.1	V
CLK IN (Pin 4) Hysteresis, V_H		0.6	1.3	2.0	V
Logic "1" Input Current (All Inputs), I_{INH}	$V_{IN} = 5V$	-	0.005	1	μA
Logic "0" Input Current (All Inputs), I_{INL}	$V_{IN} = 0V$	-1	-0.005	-	μA
Supply Current (Includes Ladder Current), I^+	$f_{CLK} = 640kHz$, $T_A = 25^\circ C$ and $\overline{CS} = HI$	-	1.3	2.5	mA
DATA OUTPUTS AND INTR					
Logic "0" Output Voltage, V_{OL}	$I_O = 1.6mA$, $V^+ = 4.75V$	-	-	0.4	V
Logic "1" Output Voltage, V_{OH}	$I_O = -360\mu A$, $V^+ = 4.75V$	2.4	-	-	V
Three-State Disabled Output Leakage (All Data Buffers), I_{LO}	$V_{OUT} = 0V$	-3	-	-	μA
	$V_{OUT} = 5V$	-	-	3	μA
Output Short Circuit Current, I_{SOURCE}	V_{OUT} Short to GND, $T_A = 25^\circ C$	4.5	6	-	mA
Output Short Circuit Current, I_{SINK}	V_{OUT} Short to V^+ , $T_A = 25^\circ C$	9.0	16	-	mA

NOTES:

- All voltages are measured with respect to GND, unless otherwise specified. The separate AGND point should always be wired to the DGND, being careful to avoid ground loops.
- For $V_{IN(-)} \geq V_{IN(+)}$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see Block Diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V^+ supply. Be careful, during testing at low V^+ levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct - especially at elevated temperatures, and cause errors for analog inputs near full scale. As long as the analog V_{IN} does not exceed the supply voltage by more than 50mV, the output code will be correct. To achieve an absolute 0V to 5V input voltage range will therefore require a minimum supply voltage of 4.950V over temperature variations, initial tolerance and loading.
- With $V^+ = 6V$, the digital logic interfaces are no longer TTL compatible.
- With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process.
- The $\overline{CS}$ input is assumed to bracket the $\overline{WR}$ strobe input so that timing is dependent on the $\overline{WR}$ pulse width. An arbitrarily wide pulse width will hold the converter in a reset mode and the start of conversion is initiated by the low to high transition of the $\overline{WR}$ pulse (see Timing Diagrams).
- CLK IN (pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately.
- None of these A/Ds requires a zero-adjust. However, if an all zero code is desired for an analog input other than 0V, or if a narrow full scale span exists (for example: 0.5V to 4V full scale) the $V_{IN(-)}$ input can be adjusted to achieve this. See the Zero Error description in this data sheet.

Timing Waveforms

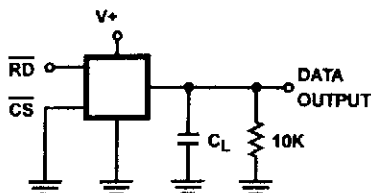


FIGURE 1A. t_{1H}

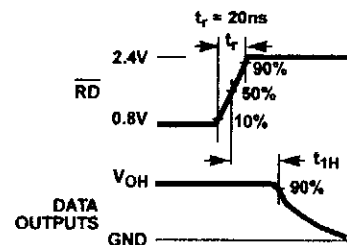


FIGURE 1B. t_{1H} , $C_L = 10pF$

Timing Waveforms (Continued)

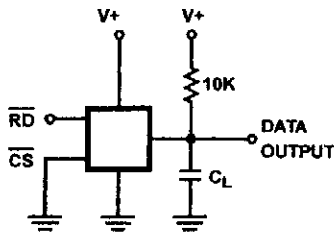


FIGURE 1C. t_{OH}

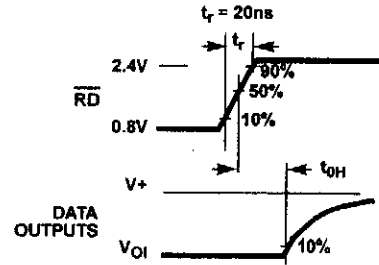


FIGURE 1D. t_{OH} , $C_L = 10\text{pF}$

FIGURE 1. THREE-STATE CIRCUITS AND WAVEFORMS

Typical Performance Curves

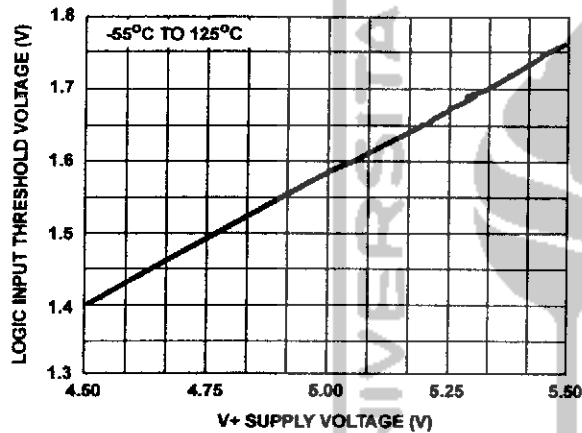


FIGURE 2. LOGIC INPUT THRESHOLD VOLTAGE vs SUPPLY VOLTAGE

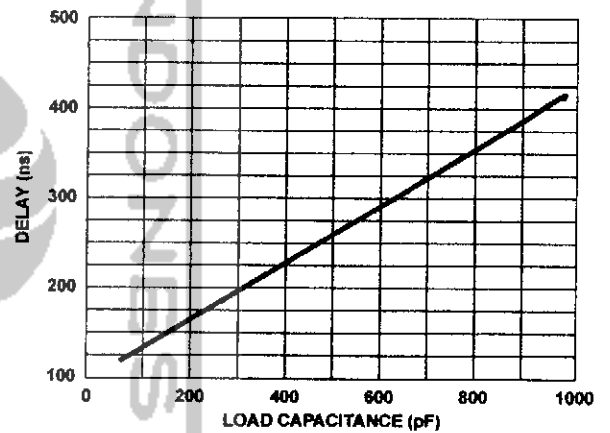


FIGURE 3. DELAY FROM FALLING EDGE OF $\overline{RD}$ TO OUTPUT DATA VALID vs LOAD CAPACITANCE

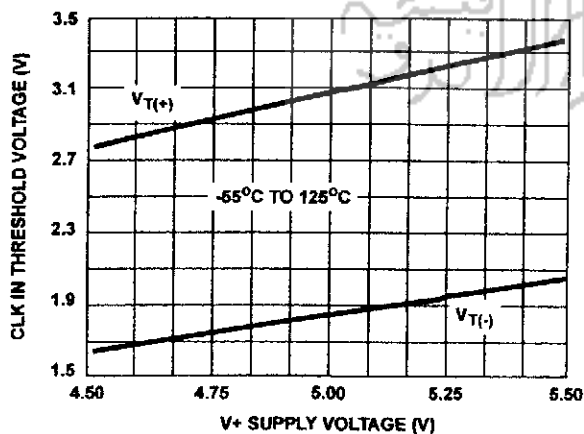


FIGURE 4. CLK IN SCHMITT TRIP LEVELS vs SUPPLY VOLTAGE

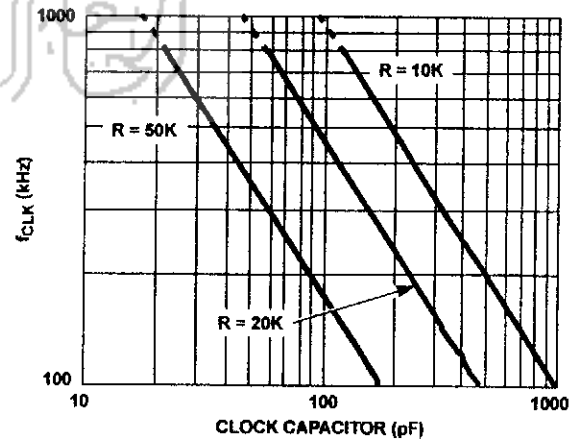


FIGURE 5. f_{CLK} vs CLOCK CAPACITOR

Typical Performance Curves (Continued)

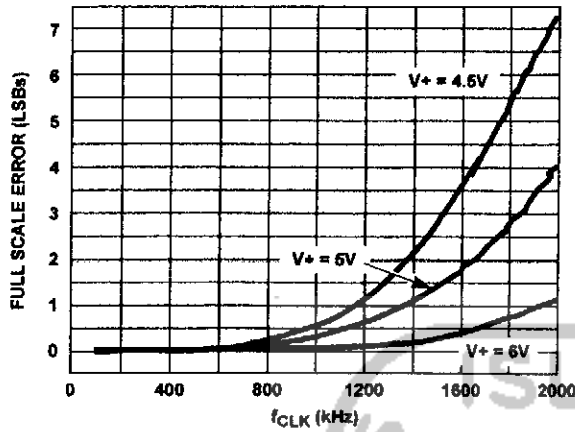


FIGURE 6. FULL SCALE ERROR vs f_{CLK}

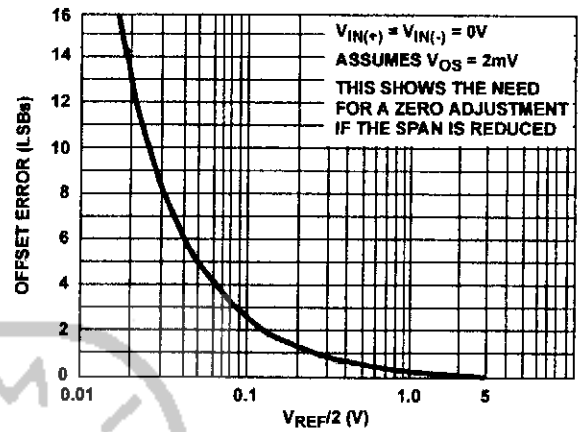


FIGURE 7. EFFECT OF UNADJUSTED OFFSET ERROR

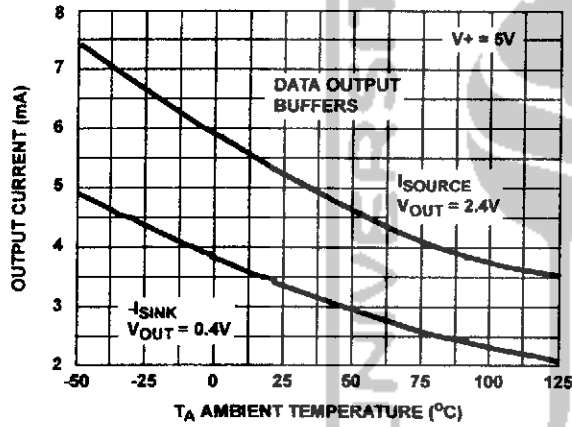


FIGURE 8. OUTPUT CURRENT vs TEMPERATURE

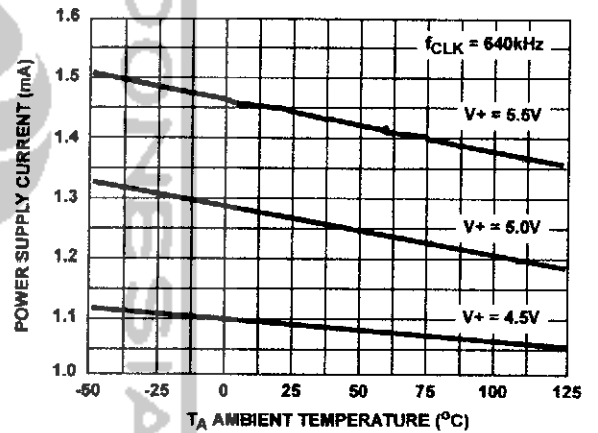


FIGURE 9. POWER SUPPLY CURRENT vs TEMPERATURE

Timing Diagrams

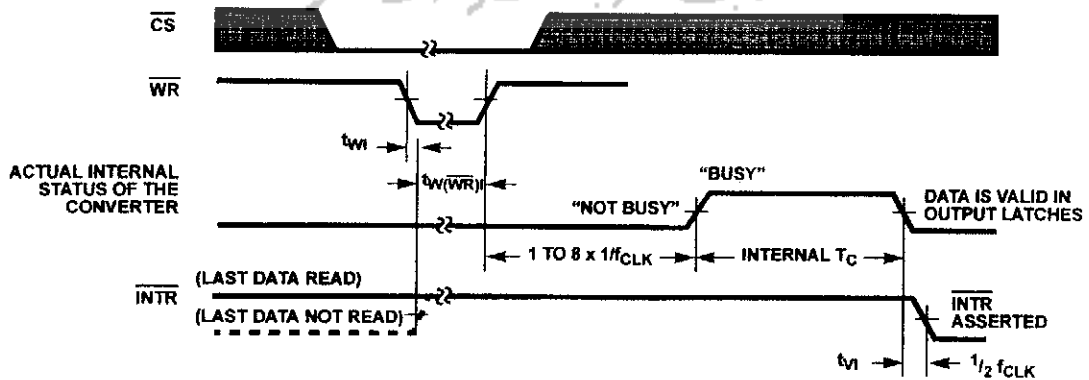


FIGURE 10A. START CONVERSION

Timing Diagrams (Continued)

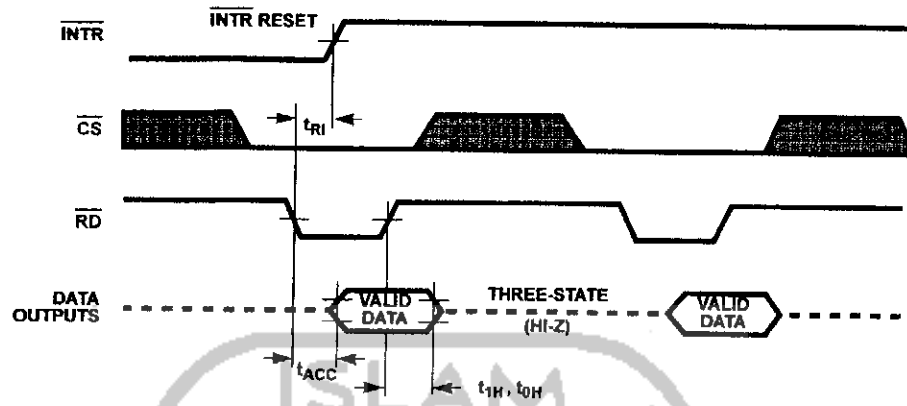


FIGURE 10B. OUTPUT ENABLE AND RESET INTR

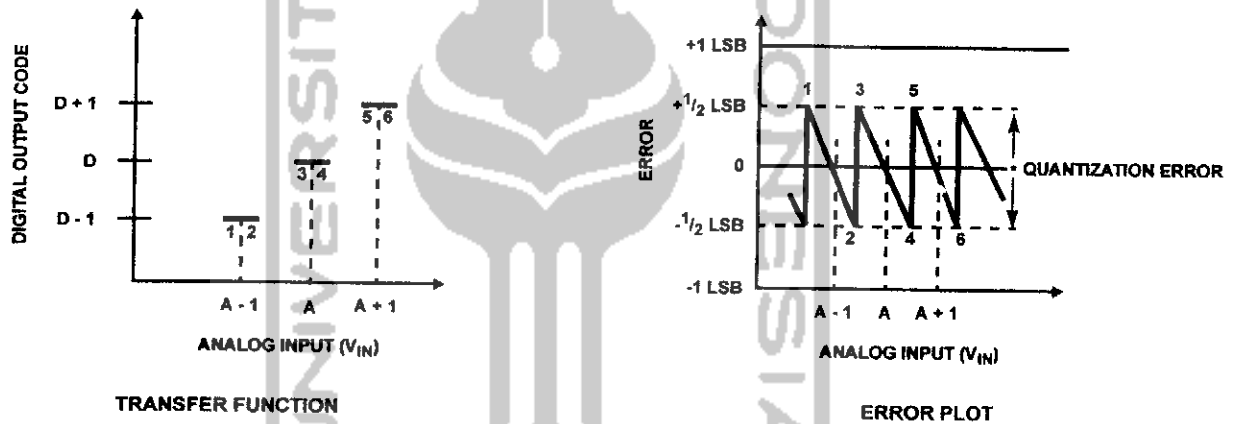


FIGURE 11A. ACCURACY = ± 0 LSB; PERFECT A/D

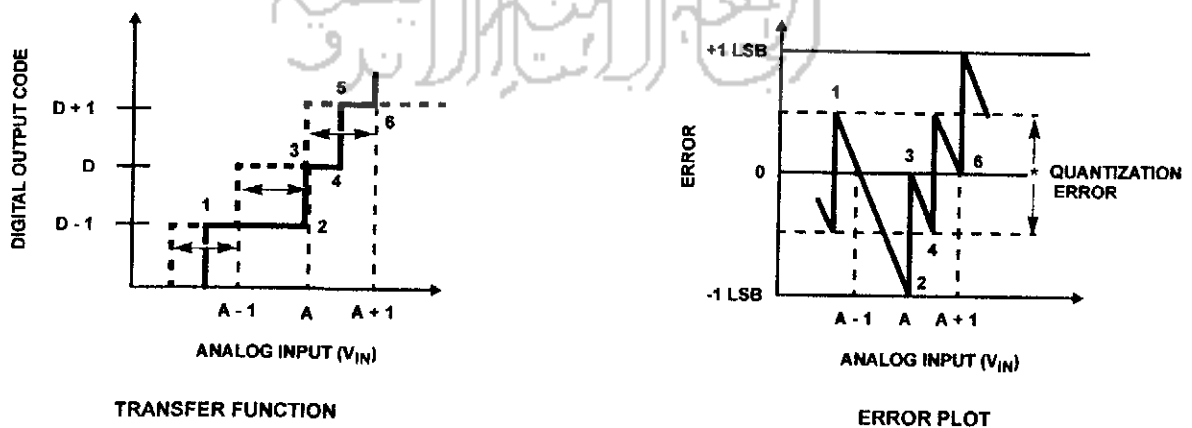


FIGURE 11B. ACCURACY = $\pm 1/2$ LSB

FIGURE 11. CLARIFYING THE ERROR SPECS OF AN A/D CONVERTER

Understanding A/D Error Specs

A perfect A/D transfer characteristic (staircase wave-form) is shown in Figure 11A. The horizontal scale is analog input voltage and the particular points labeled are in steps of 1 LSB (19.53mV with 2.5V tied to the $V_{REF}/2$ pin). The digital output codes which correspond to these inputs are shown as D-1, D, and D+1. For the perfect A/D, not only will center-value ($A - 1$, A, $A + 1$, ...) analog inputs produce the correct output digital codes, but also each riser (the transitions between adjacent output codes) will be located $\pm 1/2$ LSB away from each center-value. As shown, the risers are ideal and have no width. Correct digital output codes will be provided for a range of analog input voltages which extend $\pm 1/2$ LSB from the ideal center-values. Each tread (the range of analog input voltage which provides the same digital output code) is therefore 1 LSB wide.

The error curve of Figure 11B shows the worst case transfer function for the ADC080X. Here the specification guarantees that if we apply an analog input equal to the LSB analog voltage center-value, the A/D will produce the correct digital code.

Next to each transfer function is shown the corresponding error plot. Notice that the error includes the quantization uncertainty of the A/D. For example, the error at point 1 of Figure 11A is $+1/2$ LSB because the digital code appeared $1/2$ LSB in advance of the center-value of the tread. The error plots always have a constant negative slope and the abrupt upside steps are always 1 LSB in magnitude, unless the device has missing codes.

Detailed Description

The functional diagram of the ADC080X series of A/D converters operates on the successive approximation principle (see Application Notes AN016 and AN020 for a more detailed description of this principle). Analog switches are closed sequentially by successive-approximation logic until the analog differential input voltage [$V_{IN(+)} - V_{IN(-)}$] matches a voltage derived from a tapped resistor string across the reference voltage. The most significant bit is tested first and after 8 comparisons (64 clock cycles), an 8-bit binary code (1111 1111 = full scale) is transferred to an output latch.

The normal operation proceeds as follows. On the high-to-low transition of the WR input, the internal SAR latches and the shift-register stages are reset, and the INTR output will be set high. As long as the CS input and WR input remain low, the A/D will remain in a reset state. Conversion will start from 1 to clock periods after at least one of these inputs makes a low-to-high transition. After the requisite number of clock pulses to complete the conversion, the INTR pin will make a high-to-low transition. This can be used to interrupt a processor, or otherwise signal the availability of a new conversion. A RD operation (with CS low) will clear the INTR line high again. The device may be operated in the free-running mode by

connecting INTR to the WR input with CS = 0. To ensure start-up under all possible conditions, an external WR pulse is required during the first power-up cycle. A conversion-in-process can be interrupted by issuing a second start command.

Digital Operation

The converter is started by having CS and WR simultaneously low. This sets the start flip-flop (F/F) and the resulting "1" level resets the 8-bit shift register, resets the Interrupt (INTR) F/F and inputs a "1" to the D flip-flop, DFF1, which is at the input end of the 8-bit shift register. Internal clock signals then transfer this "1" to the Q output of DFF1. The AND gate, G1, combines this "1" output with a clock signal to provide a reset signal to the start F/F. If the set signal is no longer present (either WR or CS is a "1"), the start F/F is reset and the 8-bit shift register then can have the "1" clocked in, which starts the conversion process. If the set signal were to still be present, this reset pulse would have no effect (both outputs of the start F/F would be at a "1" level) and the 8-bit shift register would continue to be held in the reset mode. This allows for asynchronous or wide CS and WR signals.

After the "1" is clocked through the 8-bit shift register (which completes the SAR operation) it appears as the input to DFF2. As soon as this "1" is output from the shift register, the AND gate, G2, causes the new digital word to transfer to the Three-State output latches. When DFF2 is subsequently clocked, the $\bar{Q}$ output makes a high-to-low transition which causes the INTR F/F to set. An inverting buffer then supplies the INTR output signal.

When data is to be read, the combination of both $\overline{CS}$ and $\overline{RD}$ being low will cause the INTR F/F to be reset and the three-state output latches will be enabled to provide the 8-bit digital outputs.

Digital Control Inputs

The digital control inputs (CS, RD, and WR) meet standard TTL logic voltage levels. These signals are essentially equivalent to the standard A/D Start and Output Enable control signals, and are active low to allow an easy interface to microprocessor control busses. For non-microprocessor based applications, the CS input (pin 1) can be grounded and the standard A/D Start function obtained by an active low pulse at the WR input (pin 3). The Output Enable function is achieved by an active low pulse at the RD input (pin 2).

Analog Operation

The analog comparisons are performed by a capacitive charge summing circuit. Three capacitors (with precise ratioed values) share a common node with the input to an auto-zeroed comparator. The input capacitor is switched between $V_{IN(+)}$ and $V_{IN(-)}$, while two ratioed reference capacitors are switched between taps on the reference voltage divider string. The net charge corresponds to the weighted difference between the input and the current total value set by the

successive approximation register. A correction is made to offset the comparison by $1/2$ LSB (see Figure 11A).

Analog Differential Voltage Inputs and Common-Mode Rejection

This A/D gains considerable applications flexibility from the analog differential voltage input. The $V_{IN(-)}$ input (pin 7) can be used to automatically subtract a fixed voltage value from the input reading (tare correction). This is also useful in 4mA - 20mA current loop conversion. In addition, common-mode noise can be reduced by use of the differential input.

The time interval between sampling $V_{IN(+)}$ and $V_{IN(-)}$ is $4^{1/2}$ clock periods. The maximum error voltage due to this slight time difference between the input voltage samples is given by:

$$\Delta V_E(\text{MAX}) = (V_{\text{PEAK}})(2\pi f_{\text{CM}}) \left[\frac{4.5}{f_{\text{CLK}}} \right]$$

where:

ΔV_E is the error voltage due to sampling delay,

V_{PEAK} is the peak value of the common-mode voltage,

f_{CM} is the common-mode frequency.

For example, with a 60Hz common-mode frequency, f_{CM} , and a 640kHz A/D clock, f_{CLK} , keeping this error to $1/4$ LSB ($\sim 5\text{mV}$) would allow a common-mode voltage, V_{PEAK} , given by:

$$V_{\text{PEAK}} = \frac{[\Delta V_E(\text{MAX})(f_{\text{CLK}})]}{(2\pi f_{\text{CM}})(4.5)}$$

or

$$V_{\text{PEAK}} = \frac{(5 \times 10^{-3})(640 \times 10^3)}{(6.28)(60)(4.5)} \approx 1.9\text{V}$$

The allowed range of analog input voltage usually places more severe restrictions on input common-mode voltage levels than this.

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

Analog Input Current

The internal switching action causes displacement currents to flow at the analog inputs. The voltage on the on-chip capacitance to ground is switched through the analog differential input voltage, resulting in proportional currents entering the $V_{IN(+)}$ input and leaving the $V_{IN(-)}$ input. These current transients occur at the leading edge of the internal clocks. They rapidly decay and do not inherently cause errors as the on-chip comparator is strobed at the end of the clock period.

Input Bypass Capacitors

Bypass capacitors at the inputs will average these charges and cause a DC current to flow through the output resistances to the analog signal sources. This charge pumping action is worse for continuous conversions with the $V_{IN(+)}$ input voltage full scale. For a 640kHz clock frequency with the $V_{IN(+)}$

input at 5V, this DC current is at a maximum of approximately 5 μA . Therefore, **bypass capacitors should not be used at the analog inputs or the $V_{\text{REF}}/2$ pin** for high resistance sources ($>1\text{k}\Omega$). If input bypass capacitors are necessary for noise filtering and high source resistance is desirable to minimize capacitor size, the effects of the voltage drop across this input resistance, due to the average value of the input current, can be compensated by a full scale adjustment while the given source resistor and input bypass capacitor are both in place. This is possible because the average value of the input current is a precise linear function of the differential input voltage at a constant conversion rate.

Input Source Resistance

Large values of source resistance where an input bypass capacitor is not used will not cause errors since the input currents settle out prior to the comparison time. If a low-pass filter is required in the system, use a low-value series resistor ($\leq 1\text{k}\Omega$) for a passive RC section or add an op amp RC active low-pass filter. For low-source-resistance applications ($\leq 1\text{k}\Omega$), a 0.1 μF bypass capacitor at the inputs will minimize EMI due to the series lead inductance of a long wire. A 100 Ω series resistor can be used to isolate this capacitor (both the R and C are placed outside the feedback loop) from the output of an op amp, if used.

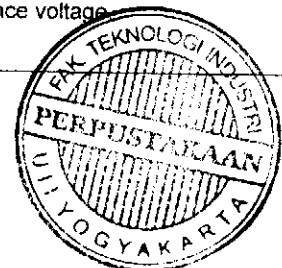
Stray Pickup

The leads to the analog inputs (pins 6 and 7) should be kept as short as possible to minimize stray signal pickup (EMI). Both EMI and undesired digital-clock coupling to these inputs can cause system errors. The source resistance for these inputs should, in general, be kept below 5k Ω . Larger values of source resistance can cause undesired signal pickup. Input bypass capacitors, placed from the analog inputs to ground, will eliminate this pickup but can create analog scale errors as these capacitors will average the transient input switching currents of the A/D (see Analog Input Current). This scale error depends on both a large source resistance and the use of an input bypass capacitor. This error can be compensated by a full scale adjustment of the A/D (see Full Scale Adjustment) with the source resistance and input bypass capacitor in place, and the desired conversion rate.

Reference Voltage Span Adjust

For maximum application flexibility, these A/Ds have been designed to accommodate a 5V, 2.5V or an adjusted voltage reference. This has been achieved in the design of the IC as shown in Figure 12.

Notice that the reference voltage for the IC is either $1/2$ of the voltage which is applied to the $V+$ supply pin, or is equal to the voltage which is externally forced at the $V_{\text{REF}}/2$ pin. This allows for a pseudo-ratiometric voltage reference using, for the $V+$ supply, a 5V reference voltage. Alternatively, a voltage less than 2.5V can be applied to the $V_{\text{REF}}/2$ input. The internal gain to the $V_{\text{REF}}/2$ input is 2 to allow this factor of 2 reduction in the reference voltage.



ADC0803, ADC0804

Such an adjusted reference voltage can accommodate a reduced span or dynamic voltage range of the analog input voltage. If the analog input voltage were to range from 0.5V to 3.5V, instead of 0V to 5V, the span would be 3V. With 0.5V applied to the $V_{IN(-)}$ pin to absorb the offset, the reference voltage can be made equal to $1/2$ of the 3V span or 1.5V. The A/D now will encode the $V_{IN(+)}$ signal from 0.5V to 3.5V with the 0.5V input corresponding to zero and the 3.5V input corresponding to full scale. The full 8 bits of resolution are therefore applied over this reduced analog input voltage range. The requisite connections are shown in Figure 13. For expanded scale inputs, the circuits of Figures 14 and 15 can be used.

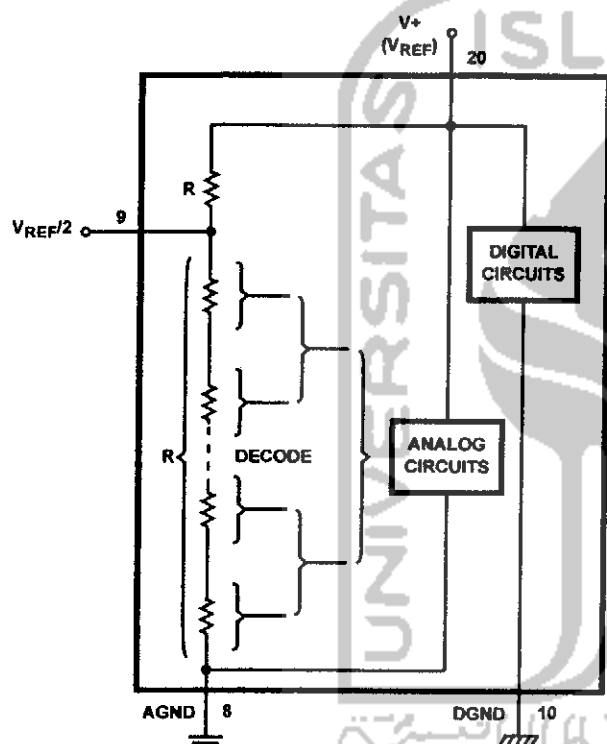


FIGURE 12. THE $V_{REFERENCE}$ DESIGN ON THE IC

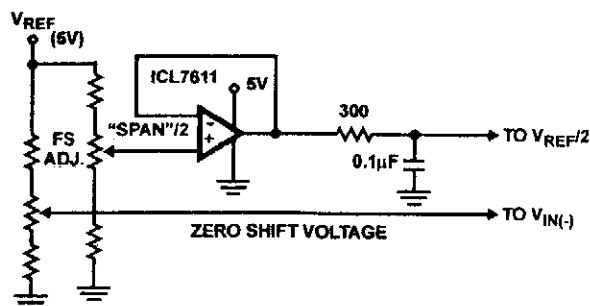


FIGURE 13. OFFSETTING THE ZERO OF THE ADC080X AND PERFORMING AN INPUT RANGE (SPAN) ADJUSTMENT

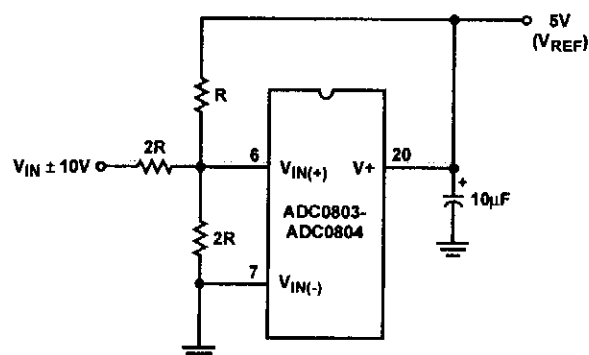


FIGURE 14. HANDLING $\pm 10V$ ANALOG INPUT RANGE

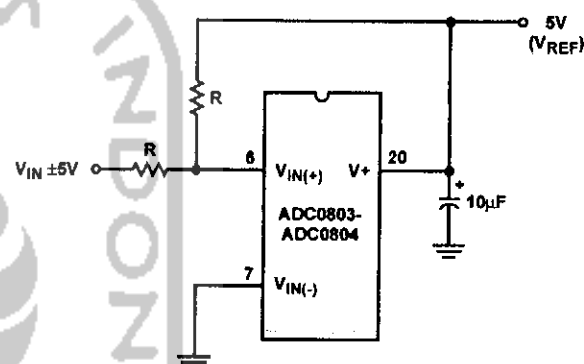


FIGURE 15. HANDLING $\pm 5V$ ANALOG INPUT RANGE

Reference Accuracy Requirements

The converter can be operated in a pseudo-ratiometric mode or an absolute mode. In ratiometric converter applications, the magnitude of the reference voltage is a factor in both the output of the source transducer and the output of the A/D converter and therefore cancels out in the final digital output code. In absolute conversion applications, both the initial value and the temperature stability of the reference voltage are important accuracy factors in the operation of the A/D converter. For $V_{REF}/2$ voltages of 2.5V nominal value, initial errors of $\pm 10mV$ will cause conversion errors of ± 1 LSB due to the gain of 2 of the $V_{REF}/2$ input. In reduced span applications, the initial value and the stability of the $V_{REF}/2$ input voltage become even more important. For example, if the span is reduced to 2.5V, the analog input LSB voltage value is correspondingly reduced from 20mV (5V span) to 10mV and 1 LSB at the $V_{REF}/2$ input becomes 5mV. As can be seen, this reduces the allowed initial tolerance of the reference voltage and requires correspondingly less absolute change with temperature variations. Note that spans smaller than 2.5V place even tighter requirements on the initial accuracy and stability of the reference source.

In general, the reference voltage will require an initial adjustment. Errors due to an improper value of reference voltage appear as full scale errors in the A/D transfer

ADC0803, ADC0804

function. IC voltage regulators may be used for references if the ambient temperature changes are not excessive.

Zero Error

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

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

Full Scale Adjust

The full scale adjustment can be made by applying a differential input voltage which is $1 1/2$ LSB down from the desired analog full scale voltage range and then adjusting the magnitude of the $V_{REF}/2$ input (pin 9) for a digital output code which is just changing from 1111 1110 to 1111 1111. When offsetting the zero and using a span-adjusted $V_{REF}/2$ voltage, the full scale adjustment is made by inputting V_{MIN} to the $V_{IN(-)}$ input of the A/D and applying a voltage to the $V_{IN(+)}$ input which is given by:

$$V_{IN(+)} f_{SADJ} = V_{MAX} - 1.5 \left[\frac{(V_{MAX} - V_{MIN})}{256} \right]$$

where:

V_{MAX} = the high end of the analog input range, and

V_{MIN} = the low end (the offset zero) of the analog range. (Both are ground referenced.)

Clocking Option

The clock for the A/D can be derived from an external source such as the CPU clock or an external RC network can be added to provide self-clocking. The CLK IN (pin 4) makes use of a Schmitt trigger as shown in Figure 16.

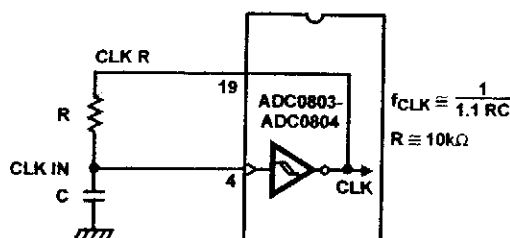


FIGURE 16. SELF-CLOCKING THE A/D

heavy capacitive or DC loading of the CLK R pin should be avoided as this will disturb normal converter operation.

Loads less than 50pF, such as driving up to 7 A/D converter clock inputs from a single CLK R pin of 1 converter, are allowed. For larger clock line loading, a CMOS or low power TTL buffer or PNP input logic should be used to minimize the loading on the CLK R pin (do not use a standard TTL buffer).

Restart During a Conversion

If the A/D is restarted ($\overline{CS}$ and $\overline{WR}$ go low and return high) during a conversion, the converter is reset and a new conversion is started. The output data latch is not updated if the conversion in progress is not completed. The data from the previous conversion remain in this latch.

Continuous Conversions

In this application, the $\overline{CS}$ input is grounded and the $\overline{WR}$ input is tied to the INTR output. This $\overline{WR}$ and INTR node should be momentarily forced to logic low following a power-up cycle to insure circuit operation. See Figure 17 for details.

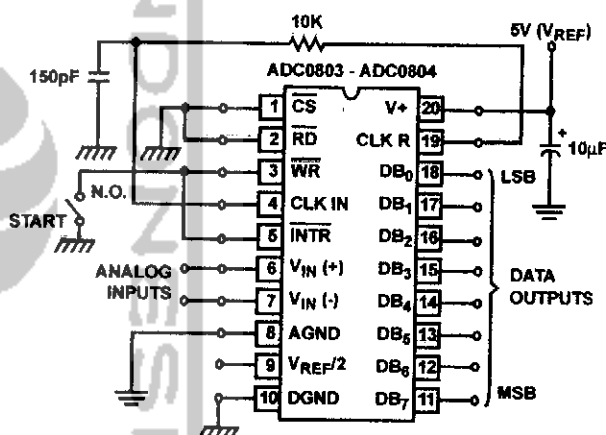


FIGURE 17. FREE-RUNNING CONNECTION

Driving the Data Bus

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

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

At higher CPU clock frequencies time can be extended for I/O reads (and/or writes) by inserting wait states (8080) or using clock-extending circuits (6800).

ADC0803, ADC0804

Finally, if time is short and capacitive loading is high, external bus drivers must be used. These can be three-state buffers (low power Schottky is recommended, such as the 74LS240 series) or special higher-drive-current products which are designed as bus drivers. High-current bipolar bus drivers with PNP inputs are recommended.

Power Supplies

Noise spikes on the V+ supply line can cause conversion errors as the comparator will respond to this noise. A low-inductance tantalum filter capacitor should be used close to the converter V+ pin, and values of 1µF or greater are recommended. If an unregulated voltage is available in the system, a separate 5V voltage regulator for the converter (and other analog circuitry) will greatly reduce digital noise on the V+ supply. An ICL7663 can be used to regulate such a supply from an input as low as 5.2V.

Wiring and Hook-Up Precautions

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

A single-point analog ground should be used which is separate from the logic ground points. The power supply bypass capacitor and the self-clocking capacitor (if used) should both be returned to digital ground. Any V_{REF/2} bypass capacitors, analog input filter capacitors, or input signal shielding should be returned to the analog ground point. A test for proper grounding is to measure the zero error of the A/D converter. Zero errors in excess of 1/4 LSB can usually be traced to improper board layout and wiring (see Zero Error for measurement). Further information can be found in Application Note AN018.

Testing the A/D Converter

There are many degrees of complexity associated with testing an A/D converter. One of the simplest tests is to apply a known analog input voltage to the converter and use LEDs to display the resulting digital output code as shown in Figure 18.

For ease of testing, the V_{REF/2} (pin 9) should be supplied with 2.560V and a V+ supply voltage of 5.12V should be used. This provides an LSB value of 20mV.

If a full scale adjustment is to be made, an analog input voltage of 5.090V (5.120 - 1/2 LSB) should be applied to the V_{IN(+)} pin with the V_{IN(-)} pin grounded. The value of the V_{REF/2} input voltage should be adjusted until the digital output code is just changing from 1111 1110 to 1111 1111. This value of V_{REF/2} should then be used for all the tests.

The digital-output LED display can be decoded by dividing the 8 bits into 2 hex characters, one with the 4 most-

significant bits (MS) and one with the 4 least-significant bits (LS). The output is then interpreted as a sum of fractions times the full scale voltage:

$$V_{OUT} = \left(\frac{MS}{16} + \frac{LS}{256} \right) (5.12) V$$

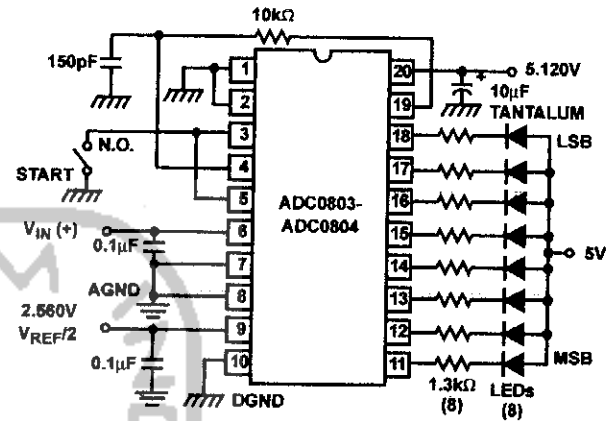


FIGURE 18. BASIC TESTER FOR THE A/D

For example, for an output LED display of 1011 0110, the MS character is hex B (decimal 11) and the LS character is hex 6 (and decimal) 6, so:

$$V_{OUT} = \left(\frac{11}{16} + \frac{6}{256} \right) (5.12) = 3.64V$$

Figures 19 and 20 show more sophisticated test circuits.

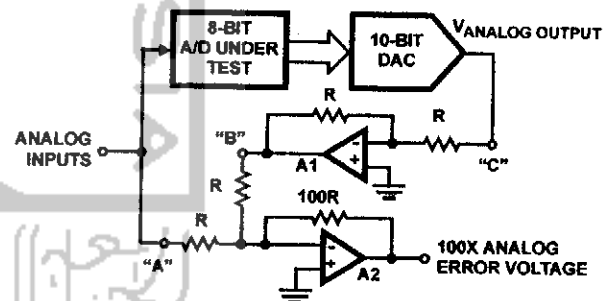


FIGURE 19. A/D TESTER WITH ANALOG ERROR OUTPUT. THIS CIRCUIT CAN BE USED TO GENERATE "ERROR PLOTS" OF FIGURE 11.

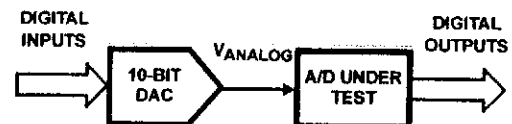


FIGURE 20. BASIC "DIGITAL" A/D TESTER

Typical Applications

Interfacing 8080/85 or Z-80 Microprocessors

This converter has been designed to directly interface with 8080/85 or Z-80 Microprocessors. The three-state output capability of the A/D eliminates the need for a peripheral interface device, although address decoding is still required to generate the appropriate $\overline{CS}$ for the converter. The A/D can be mapped into memory space (using standard memory-address decoding for $\overline{CS}$ and the $\overline{MEMR}$ and $\overline{MEMW}$ strobes) or it can be controlled as an I/O device by using the $\overline{IOR}$ and $\overline{IOW}$ strobes and decoding the address bits A0 → A7 (or address bits A8 → A15, since they will contain the same 8-bit address information) to obtain the $\overline{CS}$ input. Using the I/O space provides 256 additional addresses and may allow a simpler 8-bit address decoder, but the data can only be input to the accumulator. To make use of the additional memory reference instructions, the A/D should be mapped into memory space. See AN020 for more discussion of memory-mapped vs I/O-mapped interfaces. An example of an A/D in I/O space is shown in Figure 21.

The standard control-bus signals of the 8080 ($\overline{CS}$, $\overline{RD}$ and $\overline{WR}$) can be directly wired to the digital control inputs of the A/D, since the bus timing requirements, to allow both starting the converter, and outputting the data onto the data bus, are met. A bus driver should be used for larger microprocessor systems where the data bus leaves the PC board and/or must drive capacitive loads larger than 100pF.

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

Interfacing the Z-80 and 8085

The Z-80 and 8085 control buses are slightly different from that of the 8080. General $\overline{RD}$ and $\overline{WR}$ strobes are provided and separate memory request, $\overline{MREQ}$, and I/O request, $\overline{IORQ}$, signals have to be combined with the generalized strobes to provide the appropriate signals. An advantage of operating the A/D in I/O space with the Z-80 is that the CPU will automatically insert one wait state (the $\overline{RD}$ and $\overline{WR}$ strobes are extended one clock period) to allow more time for the I/O devices to respond. Logic to map the A/D in I/O space is shown in Figure 22. By using $\overline{MREQ}$ in place of $\overline{ORQ}$, a memory-mapped configuration results.

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

The 8085 also provides a generalized $\overline{RD}$ and $\overline{WR}$ strobe, with an $\overline{IO/M}$ line to distinguish I/O and memory requests. The circuit in Figure 22 can again be used, with $\overline{IO/M}$ in place of $\overline{IORQ}$ for memory-mapped interface, and an extra inverter (or the logic equivalent) to provide $\overline{IO/M}$ for an I/O-mapped connection.

Interfacing 6800 Microprocessor Derivatives (6502, etc.)

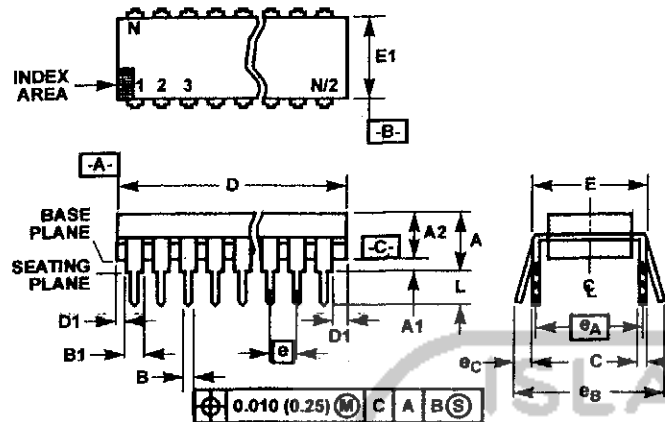
The control bus for the 6800 microprocessor derivatives does not use the $\overline{RD}$ and $\overline{WR}$ strobe signals. Instead it employs a single R/W line and additional timing, if needed, can be derived from the $\phi 2$ clock. All I/O devices are memory-mapped in the 6800 system, and a special signal, VMA, indicates that the current address is valid. Figure 23 shows an interface schematic where the A/D is memory-mapped in the 6800 system. For simplicity, the CS decoding is shown using $1/2$ DM8092. Note that in many 6800 systems, an already decoded $4/5$ line is brought out to the common bus at pin 21. This can be tied directly to the $\overline{CS}$ pin of the A/D, provided that no other devices are addressed at HEX ADDR: 4XXX or 5XXX.

In Figure 24 the ADC080X series is interfaced to the MC6800 microprocessor through (the arbitrarily chosen) Port B of the MC6820 or MC6821 Peripheral Interface Adapter (PIA). Here the $\overline{CS}$ pin of the A/D is grounded since the PIA is already memory-mapped in the MC6800 system and no CS decoding is necessary. Also notice that the A/D output data lines are connected to the microprocessor bus under program control through the PIA and therefore the A/D $\overline{RD}$ pin can be grounded.

Application Notes

NOTE #	DESCRIPTION
AN016	"Selecting A/D Converters"
AN018	"Do's and Don'ts of Applying A/D Converters"
AN020	"A Cookbook Approach to High Speed Data Acquisition and Microprocessor Interfacing"
AN030	"The ICL7104 - A Binary Output A/D Converter for Microprocessors"

Dual-In-Line Plastic Packages (PDIP)



NOTES:

1. Controlling Dimensions: INCH. In case of conflict between English and Metric dimensions, the inch dimensions control.
2. Dimensioning and tolerancing per ANSI Y14.5M-1982.
3. Symbols are defined in the "MO Series Symbol List" in Section 2.2 of Publication No. 95.
4. Dimensions A, A1 and L are measured with the package seated in JEDEC seating plane gauge GS-3.
5. D, D1, and E1 dimensions do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010 inch (0.25mm).
6. E and e_A are measured with the leads constrained to be perpendicular to datum $-C-$.
7. e_B and e_C are measured at the lead tips with the leads unconstrained. e_C must be zero or greater.
8. B1 maximum dimensions do not include dambar protrusions. Dambar protrusions shall not exceed 0.010 inch (0.25mm).
9. N is the maximum number of terminal positions.
10. Corner leads (1, N, N/2 and N/2 + 1) for E8.3, E16.3, E18.3, E28.3, E42.6 will have a B1 dimension of 0.030 - 0.045 inch (0.76 - 1.14mm).

E20.3 (JEDEC MS-001-AD ISSUE D)
20 LEAD DUAL-IN-LINE PLASTIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN	MAX	MIN	MAX	
A	-	0.210	-	5.33	4
A1	0.015	-	0.39	-	4
A2	0.115	0.195	2.93	4.95	-
B	0.014	0.022	0.356	0.558	-
B1	0.045	0.070	1.55	1.77	8
C	0.008	0.014	0.204	0.355	-
D	0.980	1.060	24.89	26.9	5
D1	0.005	-	0.13	-	5
E	0.300	0.325	7.62	8.25	6
E1	0.240	0.280	6.10	7.11	5
e	0.100 BSC		2.54 BSC		-
e_A	0.300 BSC		7.62 BSC		6
e_B	-	0.430	-	10.92	7
L	0.115	0.150	2.93	3.81	4
N	20		20		9

Rev. 0 12/93

All Intersil products are manufactured, assembled and tested utilizing ISO9000 quality systems.
Intersil Corporation's quality certifications can be viewed at website www.intersil.com/quality/iso.asp.

Intersil products are sold by description only. Intersil Corporation reserves the right to make changes in circuit design and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No sense is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.

For information regarding Intersil Corporation and its products, see web site www.intersil.com