

**RANCANG BANGUN SISTEM MIKROKONTROLER
AT89S51 SEBAGAI PENGENDALI KOMUNIKASI
SERIAL PC DENGAN MODEM QPSK UNTUK
POWER LINE COMMUNICATION**

TUGAS AKHIR

Oleh

DIA ADRIAN SYAH

06 06 04 243 0



**PROGRAM STUDI TEKNIK ELEKTRO
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/2008**

**RANCANG BANGUN SISTEM MIKROKONTROLER
AT89S51 SEBAGAI PENGENDALI KOMUNIKASI
SERIAL PC DENGAN MODEM QPSK UNTUK
POWER LINE COMMUNICATION**

TUGAS AKHIR

Oleh

DIA ADRIAN SYAH
06 06 04 243 0



**TUGAS AKHIR INI DIAJUKAN UNTUK MELENGKAPI
SEBAGIAN PERSYARATAN MENJADI SARJANA TEKNIK**

**PROGRAM STUDI TEKNIK ELEKTRO
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/2008**

PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa tugas akhir dengan judul :

**RANCANG BANGUN SISTEM MIKROKONTROLER
AT89S51 SEBAGAI PENGENDALI KOMUNIKASI
SERIAL PC DENGAN MODEM QPSK UNTUK
POWER LINE COMMUNICATION**

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. Tulisan ini bukan merupakan tiruan atau duplikasi dari tugas akhir yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 8 Juli 2008

Dia Adrian Syah

NPM 06 060 04 243 0

PENGESAHAN

Tugas akhir dengan judul :

**RANCANG BANGUN SISTEM MIKROKONTROLER
AT89S51 SEBAGAI PENGENDALI KOMUNIKASI
SERIAL PC DENGAN MODEM QPSK UNTUK
POWER LINE COMMUNICATION**

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. Tugas akhir ini telah diujikan pada sidang tanggal 7 Juli 2008 dan dinyatakan memenuhi syarat/sah sebagai tugas akhir pada Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Depok, 8 Juli 2008

Dosen Pembimbing

Dr.Ir. Purnomo Sidi Priambodo, MSc

NIP. 0407050192

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada :

Dr.Ir. Purnomo Sidi Priambodo, MSc

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan pengarahan, diskusi, bimbingan serta persetujuan sehingga penelitian tugas akhir ini dapat selesai dengan baik dan tepat waktu.

Dia Adrian Syah
NPM 06 06 04 243 0
Departemen Teknik Elektro

Dosen Pembimbing
Dr.Ir. Purnomo Sidi Priambodo, MSc

**RANCANG BANGUN SISTEM MIKROKONTROLER
AT89S51 SEBAGAI PENGENDALI KOMUNIKASI
SERIAL PC DENGAN MODEM QPSK UNTUK
POWER LINE COMMUNICATION**

ABSTRAK

Tugas akhir ini adalah membangun perangkat keras dan lunak sistem mikrokontroler *AT89S51* yang digunakan sebagai sistem kendali aliran data antara *PC* dengan *modem QPSK* atau sebaliknya yang dipergunakan untuk sistem *Power Line Communication*. Mikrokontroler *AT89S51* dipilih, karena murah dan diperhitungkan mampu untuk menangani aliran data tersebut.

Aliran data antara *PC* dan *modem QPSK* menggunakan protokol *asynchronous* yang merupakan standar komunikasi *RS-232C* pada *PC*. Ada dua metode yang dikembangkan terkait dengan kendali aliran data tersebut yaitu : *serial asynchronous* dengan metode *non-handshaking* dan *serial asynchronous* dengan metode *handshaking*. Metode *non-handshaking* dipergunakan untuk pengiriman informasi dengan kecepatan rendah, misalnya : transfer karakter (*typing*), pengiriman hasil pengukuran meter listrik, hasil pengukuran suhu ruang dan status pintu pada aplikasi *office security*. Sedangkan metode *handshaking* dipergunakan untuk pengiriman data yang besar dengan kecepatan tinggi, misalnya : suara, *video, file* dan sebagainya.

Penelitian ini difokuskan pada rancang bangun perangkat keras dan perangkat lunak sistem mikrokontroler *AT89S51* untuk mengendalikan aliran data antara *PC* dan *modem QPSK* atau sebaliknya, termasuk segala aspek yang terkait dengannya seperti kecepatan *transfer*, optimasi *buffer*, *flexibility* dalam *setup modem*. Dalam tugas akhir ini segala aspek dijelaskan secara rinci.

Kata Kunci : Komunikasi Data Serial, Komputer PC, PLC, Serial Port, RS-232C, Mikrokontroler AT89S51, Modem, QPSK, Handshaking, Serial Asynchronous

Dia Adrian Syah
NPM 06 06 04 243 0
Electrical Engineering Department

Counsellor
Dr.Ir. Purnomo Sidi Priambodo, MSc

DEVELOPMENT OF AT89S51 MICROCONTROLLER SYSTEM TO CONTROL PC SERIAL COMMUNICATION WITH QPSK MODEM DESIGNED FOR POWERLINE COMMUNICATION

ABSTRACT

This final project is the development of hardware and software of AT89S51 microcontroller system to control PC serial communication with QPSK modem designed for powerline communication (PLC). AT89S51 microcontroller has been chosen due to its low cost and its capabilities of broad applications.

Standard serial communication asynchronous protocol RS-232C is used between PC and QPSK modem designed for PLC. Two methods used in conjunction to the serial communication, i.e., non-handshaking and handshaking methods. Non-handshaking method is mostly used for data transfer with low speed, for example: character transfer for hyper terminal chatting and small data transfer such as for utilities measurements (electric and water usage), room temperature measurement, states of the doors in home or office security and automation system. In other side, handshaking method is mostly used for very large and high speed data transfer, i.e., voice, video and files.

This research is focused on the development of hardware and software of AT89S51 microcontroller system to control PC serial communication with QPSK modem designed for powerline communication (PLC). It includes all aspect related to data transfer control algorithm, buffer memory optimization and modem setup user interface.

Keywords : Serial Communication, RS-232C, Powerline Communication, PLC, Personal Computer, PC, Microcontroller AT89S51, Modem, QPSK, Serial Asynchronous Handshaking

DAFTAR ISI

	Halaman
PERNYATAAN KEASLIAN TUGAS AKHIR	ii
PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
DAFTAR SINGKATAN	xiii
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 BATASAN MASALAH	1
1.3 TUJUAN PENULISAN	1
1.4 SISTEMATIKA PENULISAN	2
BAB II DASAR TEORI	3
2.1 KOMUNIKASI SERIAL	3
2.1.1 Asynchronous	4
2.1.2 Synchronous	6
2.1.3 Kecepatan Pengiriman Data	7
2.2 STANDAR KOMUNIKASI	7
2.2.1 RS-232C	8
2.3 MEMORI	10
2.4 PPI 82C55A	12
2.4.1 Operasi Mode 0	13
2.5 MIKROKONTROLER AT89S51	14
2.5.1 Operasi Serial Port	15
2.5.1.1 Register Kontrol Serial Port	16

2.5.1.2	<i>Mode Operasi Serial Port</i>	16
2.5.1.3	<i>Timer 1 Sebagai Baud Rate Clock</i>	18
BAB III	PERANCANGAN, PENGOPERASIAN DAN	
	PEMROGRAMAN	19
3.1	HARDWARE	19
3.1.1	Rangkaian Blok Modem PLC	19
3.1.2	Rangkaian Blok Ekspansi Board DT-51	20
3.1.3	Rancangan Pengalamatan Komponen Sistem Mikrokontroler	22
3.1.4	Handshaking	23
3.2	SOFTWARE	25
3.2.1	Flowchart Pengoperasian RS-232C	26
3.2.2	Flowchart Pengoperasian Software Serial Port Pada Komputer <i>PC</i>	27
3.2.3	Flowchart Komunikasi Data Serial pada Mikrokontroler AT89S51	27
3.2.4	Flowchart Komunikasi Data Paralel ke Serial pada Mikrokontroler AT89C2051	29
3.2.5	Flowchart Komunikasi Data Serial ke Paralel pada Mikrokontroler AT89C2051	29
3.2.6	Inisialisasi Serial Port AT89S51	30
3.2.7	Baca dan Tulis Memori Eksternal WS6264	30
3.2.8	Inisialisasi PPI 82C55A	31
BAB IV	HASIL EKSPERIMEN DAN ANALISA	32
4.1	KOMUNIKASI DATA SERIAL TANPA METODE HANDSHAKING	33
4.2	KOMUNIKASI DATA SERIAL DENGAN METODE HANDSHAKING	36
BAB V	KESIMPULAN	38
5.1	KESIMPULAN	38
	DAFTAR ACUAN	39
	DAFTAR PUSTAKA	40
	LAMPIRAN	41

DAFTAR GAMBAR

	Halaman	
Gambar 2.1	Transmisi data <i>serial</i> pada jalur tunggal	4
Gambar 2.2	Komunikasi data <i>serial asynchronous</i>	5
Gambar 2.3	Komunikasi data <i>serial synchronous</i>	6
Gambar 2.4	Karakter <i>ASCII asynchronous</i> dengan <i>start bit</i> dan <i>stop bit</i>	7
Gambar 2.5	Sinyal <i>RS-232C</i> yang dimulai dengan <i>start bit</i> dan diakhiri dengan <i>stop bit</i>	9
Gambar 2.6	Konektor <i>RS-232C DB-9P male</i>	9
Gambar 2.7	Konektor <i>RS-232C DB-9P female</i>	10
Gambar 2.8	Blok diagram memori <i>SRAM WS6264</i>	11
Gambar 2.9	Blok diagram <i>PPI 82C55A</i>	12
Gambar 2.10	Format <i>mode PPI</i>	13
Gambar 2.11	Blok diagram <i>AT89S51</i>	15
Gambar 2.12	Pewaktu <i>UART mode 1</i>	17
Gambar 3.1	Blok diagram <i>modem PLC</i>	20
Gambar 3.2	Tata letak komponen <i>DT-51</i>	21
Gambar 3.3	Blok diagram komunikasi data <i>serial</i> antara komputer <i>PC</i> dengan ekspansi <i>board DT-51</i> melalui <i>interface RS-232C</i>	22
Gambar 3.4	Peta memori	23
Gambar 3.5	Konektor <i>control header pin</i>	24
Gambar 3.6	<i>Reads51</i>	25
Gambar 3.7	<i>Advanced serial port terminal 5.5</i>	26
Gambar 3.8	<i>Flowchart</i> pengoperasian <i>RS-232C</i>	26
Gambar 3.9	<i>Flowchart software serial port</i> pada komputer <i>PC</i>	27
Gambar 3.10	<i>Flowchart</i> komunikasi data <i>serial</i> pada mikrokontroler <i>AT89S51</i>	28
Gambar 3.11	<i>Flowchart</i> komunikasi data paralel ke <i>serial</i> pada mikrokontroler <i>AT89C2051</i>	29

Gambar 3.12	<i>Flowchart</i> komunikasi data <i>serial</i> ke paralel pada mikrokontroler <i>AT89C2051</i>	29
Gambar 4.1	Status koneksi komputer <i>PC</i> dengan mikrokontroler <i>AT89S51</i>	32
Gambar 4.2	Pengiriman dan penerimaan data pada komputer <i>PC</i>	33
Gambar 4.3	Kerusakan pengiriman dan penerimaan data per blok <i>75 byte</i>	34
Gambar 4.4	Pengiriman dan penerimaan data per blok <i>64 byte</i> tanpa metode <i>handshaking</i>	35
Gambar 4.5	Pengiriman dan penerimaan data secara blok dengan ukuran <i>10.1 kbyte</i> tanpa metode <i>handshaking</i>	35
Gambar 4.6	Pengiriman dan penerimaan <i>file</i> data <i>10.1 kbyte</i> dengan metode <i>handshaking</i>	37

DAFTAR TABEL

	Halaman
Tabel 2.1 Standar baku <i>RS-232C</i>	10
Tabel 2.2 <i>Mode</i> operasi memori	11
Tabel 2.3 Operasi dasar <i>PPI</i>	12
Tabel 2.4 Definisi <i>port mode 0</i>	14
Tabel 2.5 <i>Register serial port control (SCON)</i>	16
Tabel 2.6 <i>Mode serial port</i>	17
Tabel 2.7 <i>Setting baud rate</i>	18
Tabel 3.1 Dekoder pengalamatan <i>PPI 82C55A</i>	23
Tabel 3.2 Dekoder pengalamatan memori	23
Tabel 3.3 Pengkabelan <i>RS-232C</i>	24

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 2.2.1 RS-232C	41
Lampiran 2 1. Skematik Diagram DT-51	42
2. Skematik Modul Ekspansi	43
3. Skematik Paralel ke <i>Serial</i> ke Paralel	44
4. Layout Modul Ekspansi	45
5. Layout Paralel ke <i>Serial</i> ke Paralel	46
6. Gambar Perangkat	46
Lampiran 3 <i>Listing Program :</i>	47
1. <i>Listing program AT89S51</i>	47
2. <i>Listing program paralel ke serial</i>	50
3. <i>Listing program serial ke paralel</i>	51
Lampiran 4 <i>Data Sheet IC :</i>	53
○ AT89S51	
○ AT89C2051	
○ ICL232	
○ AT28C64B	
○ 82C55A	
○ WS6264	
○ MAX232	

DAFTAR SINGKATAN

PC	Personal Computer
QPSK	Quadrature Phase Shift Keying
PLC	Power Line Communication
IC	Integerated Circuit
PPI	Programmable Peripheral Interface
I/O	Input / Output
CMOS	Complementary Metal Oxida Semiconductor
TTL	Transistor-transistor Logic
EEPROM	Electrically-Erasable and Programmable Read Only Memory
SRAM	Static Read Access Memory
DTR	Data Terminal Ready
RxD	Received Data
TxD	Transmitter Data
DSR	Data Set Ready
CTS	Clear To Send
DTE	Data Terminal Equipment
DCE	Data Communication Equipment
UART	Universal Asynchronous Receiver and Transmitter
USART	Universal Asynchronous Synchronous Receiver and Transmitter
COM	Asynchronous Communication Card
Bd	Baud
bps	Bit per second
cps	Character per second
ms	Milisecond
EIA	Electronic Industries Association
CCITT	Comitte Consultatif International Telephonique et Telegraphique



mW	Milliwatts
SBUF	Serial Buffer
SCON	Serial Control
TI	Transmit Interrupt
RI	Receive Interrupt
REN	Receiver Enable
FE	Framming Error
SMOD	Serial Mode
SFR	Special Function Register
RD	Read
WR	Write
MHz	Mega Hertz
uA	Mikro Ampere
RS	Recommended Standard

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Komunikasi data dengan *serial port* pada komputer *PC* dapat menghubungkan antara komputer *PC* dengan perangkat komunikasi yang lain. Komunikasi tersebut dapat mengirimkan data yang memiliki beragam format berupa teks, gambar, suara dan sebagainya. Komputer *PC* banyak dipergunakan guna mempermudah pekerjaan di kantor, di perguruan tinggi maupun di rumah. Fungsi komputer dapat digunakan untuk menganalisa, mengumpulkan dan memproses data yang diambil dari perangkat luar, selanjutnya sesuai dengan program yang ada kemudian akan menghasilkan *output* yang sesuai dengan kebutuhan. Agar komputer *PC* dapat berkomunikasi dengan jaringan *internet*, salah satunya diperlukan perangkat yang bernama *modem*. Dalam penelitian ini, penulis memfokuskan diri untuk merancang sistem mikrokontroler yang berfungsi untuk mengendalikan aliran data dari komputer *PC* ke *modem QPSK* atau sebaliknya. *Modem QPSK* ini dikembangkan sebagai *modem Power Line Communication (PLC)*. Penelitian ini merupakan bagian dari proyek Riset Unggulan Universitas Indonesia.

1.2 BATASAN MASALAH

Penelitian ini dibatasi pada perancangan perangkat lunak dan perangkat keras sistem mikrokontroler *AT89S51* yang dipergunakan untuk mengendalikan aliran data secara sinkron antara komputer *PC* dengan *modem QPSK*.

1.3 TUJUAN PENULISAN

Penulisan ini bertujuan untuk mensistematikkan hasil penelitian berupa: perancangan perangkat lunak dan perangkat keras sistem mikrokontroler *AT89S51* yang selanjutnya dipergunakan untuk mengendalikan aliran data dari komputer *PC* ke *modem QPSK* atau dari *modem QPSK* ke komputer *PC*.

1.4 SISTEMATIKA PENULISAN

Penulisan ini terdiri dari lima bab, diuraikan sebagai berikut:

BAB I : PENDAHULUAN

Membahas latar belakang penelitian serta permasalahannya, batasan masalah, tujuan penulisan dan sistematika penulisan.

BAB II : DASAR TEORI

Penjelasan mengenai komunikasi *serial asynchronous*, *synchronous*, standar transmisi, *RS-232C*, memori *SRAM WS6264*, *PPI 82C55A*, dan mikrokontroler *AT89S51*.

BAB III : PERANCANGAN, PENGOPERASIAN DAN PEMROGRAMAN

Membahas secara lengkap *interface*, blok diagram *modem PLC*, *decoder* pengalamatan, memori eksternal, program aplikasi yang dipergunakan, pengkabelan *RS-232C*, *flowchart* pemrograman dan bahasa *assembler* pada *AT89S51*.

BAB IV : HASIL EKSPERIMEN DAN ANALISA

Menjelaskan hasil eksperimen yang telah dilakukan secara *loop* tertutup tanpa metode *handshaking* dan kemudian dibandingkan dengan menggunakan metode *handshaking*.

BAB V : KESIMPULAN

Kesimpulan dari penelitian.

BAB II

DASAR TEORI

Modem dipergunakan untuk menghubungkan komputer *PC* dengan komputer *PC* lainnya. Dalam komunikasi terdapat 3 parameter yang digunakan untuk transmisi data, yaitu sebagai berikut :

1). Transmisi Satu Arah (*Simplex*)

Pada sistem ini komunikasi terjadi hanya satu arah, dari perangkat pengirim (X) ke perangkat penerima (Y). Dimana perangkat penerima (Y) tidak dapat mengirimkan data ke perangkat (X). Komunikasi dalam satu arah terjadi seperti pada sistem pemancar radio dan penerima radio.

2). Transmisi bergantian (*Half-Duplex*)

Komunikasi hanya dapat dilakukan dalam satu arah diantara dua sistem pada waktu yang sama. Pada saat perangkat (X) mengirimkan data, perangkat (Y) hanya dapat menerima saja. Demikian sebaliknya ketika perangkat (Y) mengirimkan data, perangkat (X) hanya dapat menerima. Komunikasi *half-duplex* ini dapat dilihat pada komunikasi radio amatir, ketika terjadi pengiriman data (*transmitter on*) berupa *voice*/suara maka *receiver off* atau sebaliknya pada saat menerima, *transmitter off*.

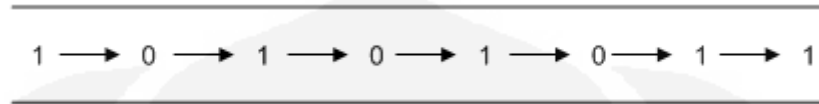
3). Transmisi dua arah (*Full-Duplex*)

Sistem ini dapat menerima dan mengirimkan data pada waktu bersamaan. Komunikasi *full-duplex* dapat dilihat seperti pada sistem *telephone*.

2.1 KOMUNIKASI SERIAL

Komunikasi data *serial* sangat berbeda dengan komunikasi data paralel, dimana pemindahan *byte* data dilakukan secara bersamaan/paralel pada waktu yang sama. Pada komunikasi *serial* pengiriman *byte* data tidak dilakukan sekaligus seperti pada komunikasi paralel tetapi setiap *bit* yang dikirimkan satu per satu melalui saluran tunggal. Komunikasi *serial* merupakan cara yang dipilih untuk transmisi data jarak jauh, namun lebih lambat dari komunikasi paralel. Oleh karena itu, untuk transmisi jarak dekat digunakan komunikasi paralel,

khususnya jika kecepatan merupakan hal yang penting misalnya pengiriman dari komputer ke *printer*. Ilustrasi dari komunikasi data *serial* dapat dilihat pada Gambar 2.1.



Gambar 2.1. Transmisi data *serial* pada jalur tunggal [1].

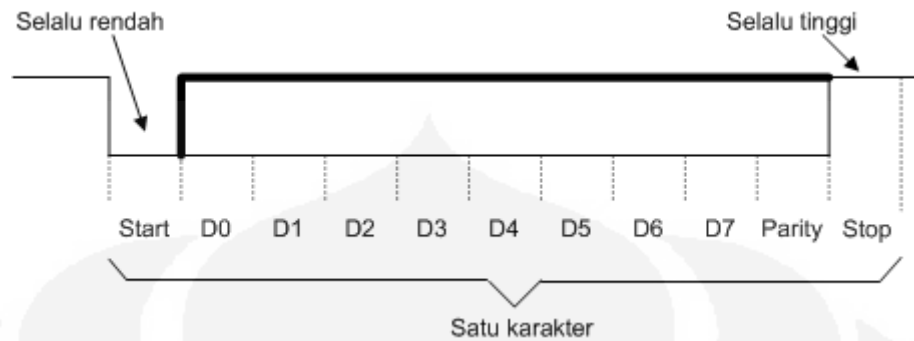
Tipe komunikasi data *serial* terdapat 2 macam, yaitu : *Asynchronous* dan *Synchronous*.

2.1.1 Asynchronous

Komunikasi data *serial asynchronous* terjadi jika *clock* yang terdapat pada masing-masing perangkat yang berkomunikasi tidak tersinkronisasi. Agar komunikasi terjalin dengan baik tanpa ada kerusakan data, dibutuhkan suatu tanda yang disisipkan pada data yang akan dikirimkan agar dapat diterima secara normal oleh penerima pada komunikasi data *serial asynchronous*. Tanda-tanda (indikator) tersebut yaitu :

- 1). Jumlah *bit* tiap karakter terdiri dari 5 sampai 8 *bit*,
- 2). *Parity bit* yang digunakan untuk mendeteksi kesalahan/*error* yaitu; ganjil (*odd*), genap (*even*) atau tanpa paritas (*no parity*),
- 3). Jumlah *stop bit* yang terdiri atas 1 *bit*, 1,5 *bit*, atau 2 *bit* sedangkan *start bit* umumnya 1 *bit*,
- 4). *Baud rate* atau kecepatan data (*bps*).

Meskipun disebut *asynchronous*, agar penerima mengetahui kapan data *byte* dikirim oleh pengirim maka data karakter yang akan dikirimkan, harus didahului oleh kondisi tinggi (*high*) ke rendah (*low*) yang dinamakan *start bit*, yang digunakan untuk mensinkronkan antara pengirim dan penerima. Setelah *start bit*, selanjutnya berisi karakter yang disertai dengan *parity bit* dan diakhiri oleh *stop bit* yang merupakan indikator yang memberitahukan bahwa pengiriman data satu karakter telah lengkap. Format komunikasi data *serial asynchronous* dapat dilihat pada Gambar 2.2.



Gambar 2.2. Komunikasi data *serial asynchronous* [1].

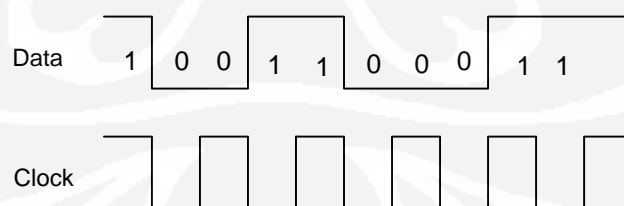
Dalam komunikasi data *serial asynchronous* dapat terjadi *error* (kesalahan) yang disebabkan karena adanya perubahan dari logika '0' menjadi logika '1' atau dari logika '1' menjadi logika '0'. Ada beberapa kesalahan dalam pengiriman data *digital* tersebut, yaitu sebagai berikut :

- 1). *Framming error* (kesalahan *frame*), kesalahan ini terjadi karena penerima tidak menemukan *stop bit* dari karakter yang dikirim. Kesalahan disebabkan oleh *noise* atau degradasi sinyal yang mengakibatkan kondisi *stop bit* berlawanan, juga karena perbedaan jumlah *bit* antara pengirim dan penerima. Dimana jumlah *bit* di penerima di *set 6 bit* sedangkan jumlah *bit* di pengirim di *set 7 bit* sehingga karakter yang diterima berbeda dengan karakter yang dikirimkan.
- 2). *Overrun Error* disebabkan karena kesalahan dalam menentukan *baud rate* dari kedua perangkat yang berkomunikasi, dimana pengirim di *set 9600 bps* sedangkan penerima di *set 1200 bps* sehingga karakter yang baru datang menindih karakter yang sebelumnya lebih dahulu tiba tetapi belum selesai di proses.
- 3). *Parity error* (kesalahan paritas), kesalahan ini terjadi ketika *bit parity* yang diterima tidak sesuai. Hal ini biasanya disebabkan oleh kekacauan pada saluran transmisi atau juga dapat disebabkan karena perbedaan parameter yang digunakan pada kedua peralatan yang berkomunikasi, dimana *receiver* di *set pada parity enable* sedangkan *transmitter* di *set pada parity disable*.

2.1.2 Synchronous

Pada komunikasi data *serial synchronous*, *clock* atau pewaktu dibangkitkan dan disinkronisasi oleh pengirim dan penerima. Protokol *synchronous* menghasilkan data *stream* pada *clock* yang tetap, dimana pengaturan *clock* tidak hanya pada *bit* dalam karakter, tetapi juga pewaktu antara karakter ke karakter [2]. Karakter yang dikirim pada komunikasi *serial synchronous* tidak mempunyai *start bit* dan *stop bit*, untuk itu diperlukan tambahan logika untuk mensinkronkan guna pengiriman awal karakter. Protokol *synchronous* lebih kompleks dalam pengiriman dan penerimaan data dibandingkan dengan protokol *asynchronous*. Protokol transmisi *synchronous* dalam pengiriman mempunyai lebar data dari 10 *byte* sampai ratusan *byte*, termasuk perintah untuk mendeteksi kemungkinan kesalahan dalam blok data. Penggunaan perintah pada tingkat blok dari karakter memberikan proteksi yang lebih tinggi terhadap kesalahan dengan memeriksa *bit* dalam aliran data dan lebih efisien untuk pemakaian *bandwidth* dalam komunikasi.

Protokol *synchronous*, informasi kontrol dalam setiap blok terdiri dari serangkaian jumlah blok, dimana penerima dapat meminta pengiriman ulang dengan blok yang spesifik. Format komunikasi data *serial synchronous* dapat dilihat pada Gambar 2.3. Komunikasi data *serial synchronous* tidak memerlukan *start bit* dan *stop bit* sehingga meningkatkan *bandwidth* data menjadi lebih lebar, juga memungkinkan untuk beroperasi pada *bit rate* yang tinggi dengan jarak yang sangat jauh.



Gambar 2.3. Komunikasi data *serial synchronous*.

2.1.3 Kecepatan Pengiriman Data

Komunikasi data *serial* mempunyai besaran kecepatan pengisyratan, kecepatan yang dimaksud harus dinyatakan dengan setepat-tepatnya. Secara umum ada 3 cara untuk menyatakan kecepatan sebagai berikut:

- 1). Kecepatan modulasi yang dinyatakan dengan *Baud (Bd)*.

Sebagai contoh untuk QPSK *baud* terdiri dari dua *bit* per simbol (2-*bits/symbol*) yang dikirimkan dalam setiap detik, dimana simbol tersebut memiliki lebih dari 2 keadaan yang direfresentasikan dalam *binary bits* (00, 01, 10 dan 11).

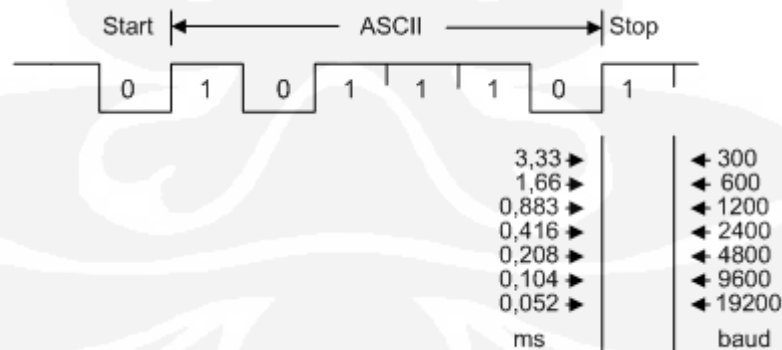
- 2). Kecepatan sinyal yang dinyatakan dengan *bit per detik (bps)*.

Bit per detik merupakan refresentasi banyaknya *bit* yang dikirimkan dalam setiap detiknya.

- 3). Kecepatan transmisi yang dinyatakan dengan karakter per detik (*cps*).

Karakter terdiri dari beberapa *bit* yang telah ditentukan, dalam hal ini penulis menggunakan 10-*bit*/karakter atau 10-*bit* setiap satu *frame* per karakter yang terdiri dari 1-*bit* *start*, 8-*bits* data dan 1-*bit* *stop*. Bila kecepatan pengiriman menggunakan 9600 *bps*, maka terdapat 960 *frame* per karakter atau 960 *cps*.

Gambar 2.4 merupakan contoh perioda waktu. Jika $T=3,33ms$, maka *Baud rate* = $1 / 3,33 ms = 300 baud$ [1].



Gambar 2.4. Karakter *ASCII asynchronous* dengan *start bit* dan *stop bit* [1].

2.2 STANDAR KOMUNIKASI

Pada tahun 1960, komite standar yang sekarang dikenal sebagai asosiasi industri elektronik mengembangkan standar *interface* untuk perangkat komunikasi data. Komunikasi data adalah pertukaran data *digital* antara komputer utama

dengan terminal komputer lainnya atau dua terminal tanpa menggunakan komputer. Kerusakan data (*error*) dapat terjadi saat data dikirimkan melalui kanal *analog* yang dirancang relatif kompleks, maka itulah dibutuhkan standar komunikasi data guna mengatasi agar tidak terjadi kerusakan data. Standar komunikasi data ini juga digunakan untuk menghubungkan berbagai jenis perangkat yang berbeda pembuatannya. Standar komunikasi data ini dikenal sebagai *RS-232C/EIA232C*.

2.2.1 RS-232C

RS-232C merupakan standard *interface* untuk komunikasi data *serial* yang menghubungkan *Data Terminal Equipment (DTE)* dengan *Data Communication Equipment (DCE)* atau dapat juga menghubungkan antara *DTE* dengan *DTE*.

Data Terminal Equipment (DTE) merupakan perangkat yang dilengkapi *Universal Asynchronous Receiver and Transmitter (UART)* atau *Universal Asynchronous Synchronous Receiver and Transmitter (USART)* yang dapat mengubah data paralel ke data *serial* atau sebaliknya. Perangkat *DTE* ini pada komputer *PC* disebut *Asynchronous Communication Card (COM)*.

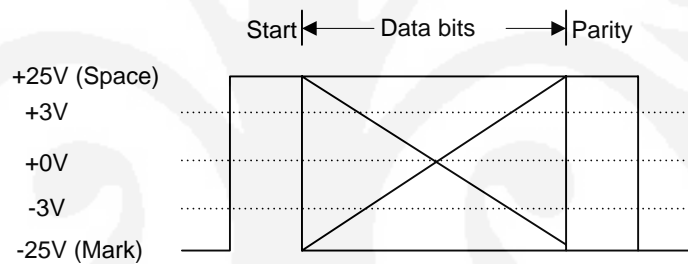
Modem sebagai *Data Communication Equipment (DCE)* adalah perangkat yang dapat mengubah data *serial* ke besaran *analog* yang dapat di transmisikan pada saluran transmisi seperti; telepon, listrik atau pemancar radio.

Untuk komunikasi antar dua komputer atau komputer dengan modem dapat menggunakan kabel *RS-232C*. Penggunaan kabel *RS-232C* jaraknya tidak lebih dari 15 meter (*50 feet*) pada kecepatan *20 Kbaud*, hal ini karena sinyal *serial digital* tersebut tidak ditumpangkan pada sinyal *carrier* (pembawa). Bila jarak yang dipergunakan lebih dari 15 meter (*50 feet*) pada kecepatan *20 Kbaud* [1], sinyal yang ditransmisikan akan mengalami degradasi oleh *noise* dan terjadi *attenuasi*. Jarak komunikasi dapat diperpanjang beberapa ratus *feet*, tetapi kecepatan data harus diturunkan, misalnya dengan kecepatan *2400 baud* dapat mencapai jarak maksimum *150 feet* [1].

Serial interface RS-232C memberi ketentuan level logika '1' disebut *mark* terletak antara *-3 Volt* hingga *-25 Volt* dan logika '0' disebut *space* terletak

antara +3 Volt hingga +25 Volt. Daerah tegangan antara -3 Volt hingga +3 Volt adalah *invalid level*, yaitu daerah yang tidak memiliki keadaan logika. Standar RS-232C tidak mendefinisikan jumlah *bit* data antara *start bit* dan *stop bit* yang dikirim, tetapi pada umumnya *bit* data yang dikirim berjumlah 5 sampai 8 *bit*.

Jika tidak ada karakter yang dikirim, *line signal* berada pada keadaan *idle*, ini ditunjukkan dengan tegangan *mark*. Proses transmisi dimulai saat *line* berubah ke *space*. Saat dimulai *start bit* lihat Gambar 2.5, selanjutnya diikuti oleh *bit* data yang ditunjukkan dengan tegangan *mark* (logika '1') dan tegangan *space* (logika '0'). Setelah *bit* data maka ada *bit parity* yang digunakan untuk menunjukkan jika ada kesalahan pada *bit-bit* data. Terakhir setelah *bit parity* dikirim, maka *line signal* kembali ke keadaan *mark*. Ini merupakan awal *stop bit*. Setelah *bit stop* selesai maka *transmitter* siap untuk mengirimkan karakter baru.



Gambar 2.5. Sinyal RS-232C yang dimulai dengan *start bit* dan diakhiri dengan *stop bit* [1].

Pada *serial port interface* terdapat dua jenis konektor RS-232C yaitu DB-9 dan DB-25, dalam penelitian ini konektor yang digunakan jenis RS-232C DB-9P female dan RS-232C DB-9P male yang mempunyai 9 *pin*. RS-232C ini dapat dilihat pada Gambar 2.6 dan Gambar 2.7.



Gambar 2.6. Konektor RS-232C DB-9P male.



Gambar 2.7. Konektor RS-232C DB-9P female.

Tabel 2.1. Standar baku RS-232C [1].

No Pin	EIA RS-232C Circuit	CCITT V-24 Circuit	RS-232C Description	Sinyal Direction
5	AB	102	Signal ground / Common ground	Ground Common
2	BB	104	Received Data (RxD)	Data from DCE
3	BA	103	Trasmitter Data (TxD)	Data to DCE
1	CF	109	Receive Line Signal Detector	Control from DCE
4	CD	108	Data Terminal Ready (DTR)	Control to DCE
6	CC	107	Data Set Ready (DSR)	Control from DCE
7	CA	105	Request To Send (RTS)	Control to DCE
8	CB	106	Clear To Send (CTS)	Control from DCE
9	CE	125	Ring Indikator	Control from DCE

Pada Tabel 2.1 diatas merupakan standar baku yang ada pada EIA (Electronic Industries Association) RS-232C dan CCITT-V24 (Comitte Consultatif International Telephonique et Telegraphique). Primary Asynchronous Communication Adapter yang disebut sebagai COM1 dengan alamat port 3F8H hingga 3FFH dan Secondary Asynchronous Communication Adapter yang disebut sebagai COM2 dengan alamat port 2F8H hingga 2FFH. Uraian lebih lanjut mengenai pin RS-232C dapat dilihat di lampiran.

2.3 MEMORI

Memori disini berfungsi sebagai tempat penyimpanan data. Memori yang digunakan pada penelitian tugas akhir ini adalah jenis *Stacic Random Access*

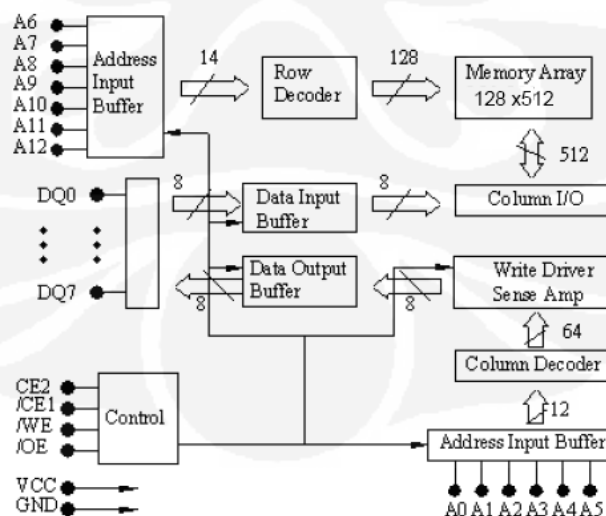
Memori (SRAM). SRAM adalah jenis memori paralel yang pengaksesan datanya dilakukan secara acak atau memori yang dapat dibaca/ditulis. Data yang disimpan dalam SRAM bersifat sementara (*Volatile*) atau datanya akan hilang bila catu dayanya dimatikan. Memori SRAM ini mempunyai kapasitas penyimpanan data sebanyak 64K (8K x 8 bit) / 8,192 words dengan teknologi CMOS yang mempunyai daya rendah. Memori ini mempunyai waktu akses dengan kecepatan tinggi maksimal sampai 70 ns dan disipasi daya saat *standby* atau tidak digunakan, arus yang dipakai lebih kecil sebesar 1 uA. Mode operasi SRAM WS6264 dapat dilihat pada Tabel 2.2.

Tabel 2.2. Mode operasi memori [3].

MODE	/CE	CE2	/WE	/OE	DQ0~7
<i>Standby</i>	H	X	X	X	<i>High Z</i>
	X	L	X	X	
<i>Output Disable</i>	L	H	H	H	<i>High Z</i>
<i>Read</i>	L	H	H	L	<i>DOUT</i>
<i>Write</i>	L	H	L	X	<i>DIN</i>

Catatan : X : Low (L) atau High (H)

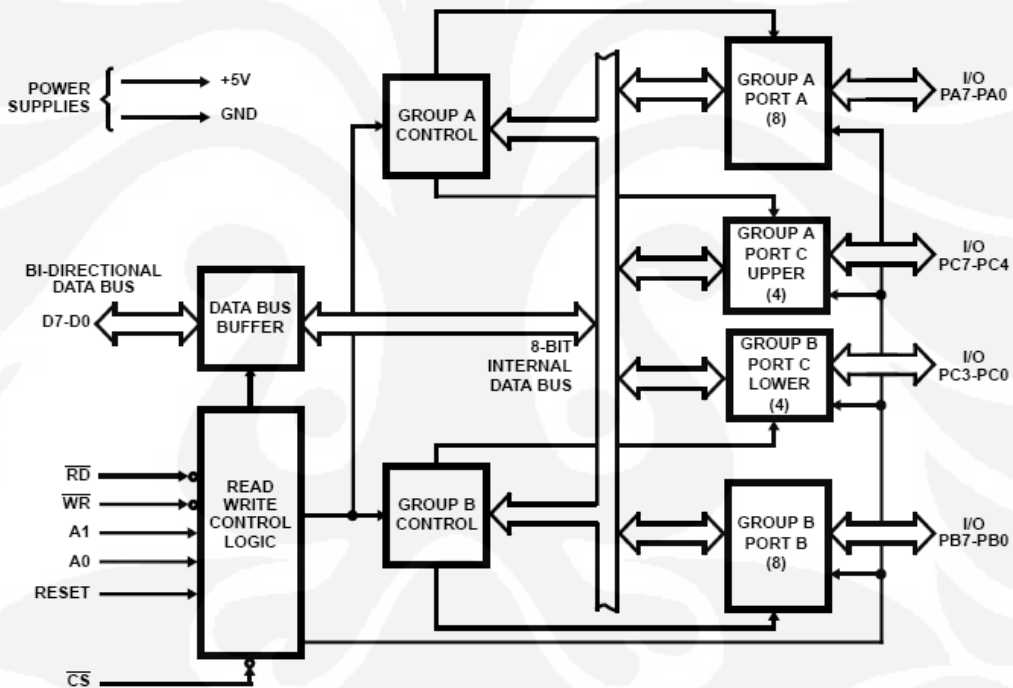
Proses pembacaan atau penulisan pada *Static RAM* tidak membutuhkan komponen eksternal. Blok diagram SRAM WS6264 terlihat pada Gambar 2.8.



Gambar 2.8. Blok diagram memori SRAM WS6264 [3].

2.4 PPI 82C55A

Programmable Peripheral Interface (PPI) merupakan *IC interface* yang digunakan untuk *input* atau *output (I/O)* yang dapat diprogram. *PPI 82C55A* ini mempunyai 24 *pin I/O*, 2 *group* pemrograman dan 3 *mode* operasi. *PPI 82C55A* mempunyai *performance* tinggi yang dibuat dengan teknologi *CMOS* yang dirancang dengan operasi daya yang rendah. *PPI 82C55A* ini juga kompatibel dengan *level TTL* dan mempunyai kecepatan operasi yang tinggi tanpa keadaan menunggu dengan *frekuensi 5MHz* dan *8MHz*. Blok diagram *PPI 82C55A* ini terlihat pada Gambar 2.9 dan untuk pengoperasian *I/O PPI 82C55A* dapat digunakan Tabel 2.3.



Gambar 2.9. Blok diagram *PPI 82C55A* [4].

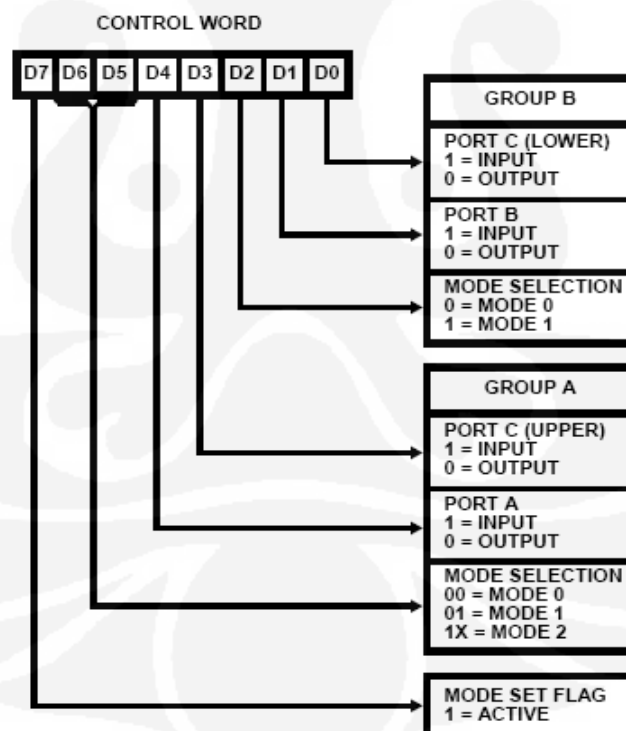
Tabel 2.3. Operasi dasar *PPI* [4].

A1	A0	RD	WR	CS	Operasi Input (Read)
0	0	0	1	0	Port A ke Data Bus
0	1	0	1	0	Port B ke Data Bus
1	0	0	1	0	Port C ke Data Bus
1	1	0	1	0	Control ke Word Data Bus

Operasi Output (Write)					
0	0	1	0	0	Data Bus ke Port A
0	1	1	0	0	Data Bus ke Port B
1	0	1	0	0	Data Bus ke Port C
1	1	1	0	0	Data Bus ke Control
Fungsi Disable					
x	x	x	x	1	Data Bus ke Three-State
x	x	1	1	0	Data Bus ke Three-State

2.4.1 Operasi Mode 0

Mode 0 merupakan dasar *Input/Output*. Konfigurasi dari fungsi ini memberikan operasi sederhana *input* dan *output* dari masing-masing ke 3 port, Format *mode I/O* dapat dilihat pada blok diagram Gambar 2.10. Operasi *mode 0* tidak menggunakan metode *handshaking*, dua port terdiri dari 8-bit (*port A* dan *port B*) dan *port C* terbagi 2 masing-masing 4-bit, dimana *port C* ini dapat berfungsi sebagai *input* dan *output*. Definisi *port mode* untuk *group A* dan *group B* dapat dilihat pada Tabel 2.4.



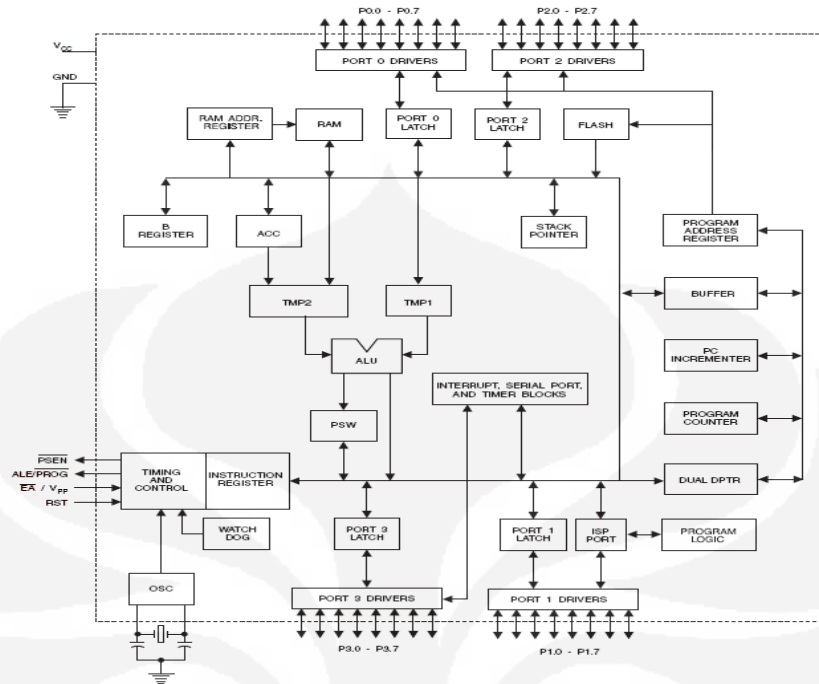
Gambar 2.10. Format *mode PPI* [4].

Tabel 2.4. Definisi *port mode 0* [4].

A		B		Group A			Group B	
D4	D3	D2	D1	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	Output	Output	1	Output	Output
0	0	0	1	Output	Output	2	Output	Input
0	0	1	0	Output	Output	3	Input	Output
0	0	1	1	Output	Output	4	Input	Input
0	1	0	0	Output	Input	5	Output	Output
0	1	0	1	Output	Input	6	Output	Input
0	1	1	0	Output	Input	7	Input	Output
0	1	1	1	Output	Input	8	Input	Input
1	0	0	0	Input	Output	9	Output	Output
1	0	0	1	Input	Output	10	Output	Input
1	0	1	0	Input	Output	11	Input	Output
1	0	1	1	Input	Output	12	Input	Input
1	1	0	0	Input	Input	13	Output	Output
1	1	0	1	Input	Input	14	Output	Input
1	1	1	0	Input	Input	15	Input	Output
1	1	1	1	Input	Input	16	Input	Input

2.5 MIKROKONTROLER AT89S51

AT89S51 adalah mikrokontroler 8-bit yang beroperasi dengan daya rendah, dan mempunyai *performance tinggi* dengan teknologi *CMOS*. Divais ini dibuat oleh *ATMEL* menggunakan teknologi memori yang data tidak hilang bila catu dayanya mati atau disebut juga dengan *NonVolatile* memori dan kompatibel dengan set instruksi *80C51* yang digunakan untuk standar industri. Pada *Flash chip* dapat diprogram ulang dengan *in-system* atau menggunakan *programmer* memori *NonVolatile* konvensional. *AT89S51* merupakan mikrokontroler dengan fleksibel yang tinggi dan solusi murah untuk membuat aplikasi kontrol. Pada mikrokontroler terdapat banyak komponen yang terintegrasi didalamnya, dimana *AT89S51* ini mempunyai *flash* memori 4K bytes, 128 RAM, 32 I/O, *Watchdog timer*, 2 data pointer, dua 16-bit timer/counter, 5 vector dengan 2 level interrupt, komunikasi *serial port* jenis *full-duplex*, *on-chip* osilator dan rangkaian *clock*. Blok diagram rangkaian terpadu mikrokontroler *AT89S51* dapat dilihat pada Gambar 2.11.



Gambar 2.11. Blok diagram AT89S51 [5].

2.5.1 Operasi Serial Port

Pengoperasian *UART* pada *AT89S51* sama dengan *UART* pada *89C51*. *Serial port* dapat dioperasikan dalam beberapa *mode* yaitu *mode 0*, *mode 1* dan *mode 3* yang masing-masing *mode* memiliki lebar frekuensi yang berbeda atau *baud rate* yang bervariasi. Fungsi utama dari *serial port* adalah untuk mengubah data dari paralel ke *serial* yang menghasilkan data *output* dan konversi data dari *serial* ke paralel yang menghasilkan data *input*. Untuk mengakses *hardware serial port* melalui pin *TXD* (P3.1) sebagai *output serial* untuk pengiriman data dan pin *RXD* (P3.0) sebagai *input serial* untuk penerimaan data. *Serial port* disini menggunakan operasi *full-duplex*, dimana data yang dikirim dan diterima secara bersamaan. Untuk komunikasi *serial* digunakan dua *register* yaitu *SCON* pada alamat 98H dan *SBUF* pada alamat 99H. *Buffer serial port (SBUF)* terbagi dua yaitu untuk menerima dan mengirim, yang kedua *register* tersebut dapat diakses melalui *special function register (SFR)*. Penulisan ke data *SBUF* untuk pengiriman dan pembacaan *SBUF* untuk menerima data. Dua *register* tersebut terpisah dan berbeda, *register* pengirim hanya untuk penulisan saja dan *register* penerima hanya untuk membaca saja.

Untuk mengoperasikan *serial port* menggunakan *register serial port control (SCON)* yang dilokasikan pada alamat 98H yang terdiri dari *register bit status dan kontrol bit*. *Bit kontrol untuk men-set mode untuk port serial dan bit status* menunjukkan akhir dari karakter yang dikirim dan diterima.

2.5.1.1 Register Kontrol Serial Port

Mode operasi dari serial port dapat dipilih dengan penulisan ke *register SCON*. Sebelum menggunakan *serial port*, *register SCON* terlebih dahulu di inisialisasi untuk memilih atau menentukan *mode operasi yang benar*. Fungsi masing-masing *bit* pada *register SCON* dapat dilihat pada Tabel 2.5.

Tabel 2.5. Register serial port control (SCON) [5].

Bit	Simbol	Alamat	Keterangan
7	FE/SM0	9FH	Framming Error bit/Serial port mode bit 0. (Tabel 2.6)
6	SM1	9EH	Serial port mode bit 1. (Tabel 2.6)
5	SM2	9DH	Serial port mode bit 2. untuk komunikasi multiprocessor dalam mode 2 dan 3.
4	REN	9CH	Receiver enable. Harus di set untuk menerima karakter.
3	TB8	9BH	Pengiriman bit 8 atau bit 9 untuk pengiriman mode 2 dan 3..
2	RB8	9AH	Penerimaan bit 8 atau bit 9
1	TI	99H	Transmit interrupt flag. Set pada akhir pengiriman karakter dan di clear oleh software.
0	RI	98H	Receive interrupt flag. Set pada akhir penerimaan karakter dan di clear oleh software.

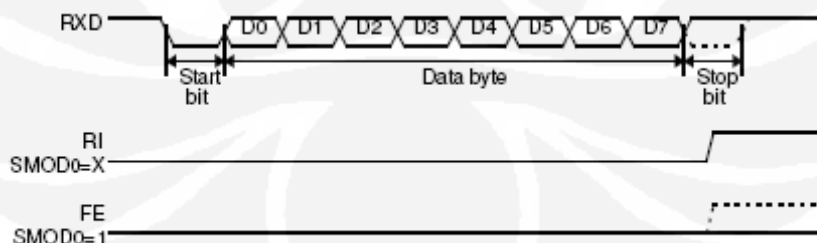
2.5.1.2 Mode Operasi Serial Port

Serial port mempunyai empat (4) *mode operasi*, pemilihan *mode operasi* dengan menuliskan 1 atau 0 pada *SM0* dan *SM1* bit dalam *register SCON*. Tiga dari *mode* mengijinkan untuk komunikasi *serial asynchronous* dengan tiap karakter yang diterima atau dikirim dalam *frame* yang terdapat *start bit* dan *stop bit*. Pemilihan *mode operasi serial port* dijelaskan pada Tabel 2.6.

Tabel 2.6. *Mode serial port* [5].

SM0	SM1	Mode	Keterangan	Baud Rate
0	0	0	<i>Shift register</i>	<i>FCPU PERIPH/6</i>
0	1	1	<i>8-bit UART</i>	<i>Variable di set oleh timer</i>
1	0	2	<i>9-bit UART</i>	<i>FCPU PERIPH /32 or /16</i>
1	1	3	<i>39-bit UART</i>	<i>Variable di set oleh timer</i>

Pada penelitian tugas akhir ini menggunakan *mode 1*, dimana terdapat *10-bit* yang dikirimkan (melalui *TxD*) atau yang diterima (melalui *RxD*). *10-bit* tersebut terdiri dari *1-bit start*, *8-bit data (LSB first)* dan *1-bit stop*. Untuk operasi penerimaan, *stop bit* masuk ke *RB8* pada *SCON* dan *baud rate* di *set* oleh *overflow timer 1*. Pengiriman dimulai dengan menulis ke *SBUF*, data *output* pada *TxD* dimulai dengan *start bit* diikuti dengan *8-bit data* kemudian *stop bit* dan *transmit interrupt (TI)* terjadi ketika *stop bit* telah terdeteksi pada *TxD*. Penerimaan data dimulai dengan terjadinya perubahan dari logika 1 ke logika 0 pada *RxD* dan kemudian peng-*clock*-an *8-bit data* dalam *register geser (shift register) port serial*. Setelah semua *8-bit data* diterima *SBUF*, *stop bit (bit ke 9)* masuk ke *RB8* pada *SCON*, dan *receive interrupt flag* di *set*. Aliran data yang diterima oleh *RxD* diawali dengan *start bit* kemudian diikuti oleh data *byte*, selanjutnya yang terakhir adalah *stop bit*. Setelah *stop bit* diterima, maka *RI register flag* bernilai 1 dan data disimpan dalam *SBUF*. Aliran data ini terlihat pada Gambar 2.12.



Gambar 2.12. Pewaktu *UART mode 1* [6].

2.5.1.3 Timer 1 Sebagai Baud Rate Clock

Baud rate adalah kecepatan untuk mengalirkan data dalam 1 detik dengan mengatur *clock/pewaktu* dengan menginisialisasi *register timer mode (TMOD)* dan menempatkan nilai yang benar dalam *register TH1*. Pembangkit *baud rate* menggunakan *timer 1* pada *mode 1* dan *mode 3* ditentukan oleh *timer 1 overflow rate* dan nilai *SMOD*. Dalam penelitian ini menggunakan *serial port mode 1* dengan *baud rate 57600* dibangkitkan oleh *timer 1 (mode 2, 8-bit auto reload)* yang *overflow rate*-nya dibagi dengan 16 dengan *SMOD = 1*. *Baud rate* dapat dipilih sesuai kebutuhan dengan memberikan nilai yang bervariasi pada *register TH1 (timer 1 high)*, variasi *baud rate* ini dapat dilihat pada Tabel 2.7. Besarnya *baud rate* dapat dihitung dengan menggunakan persamaan 2.1 untuk *SMOD = 0* dan persamaan 2.2 untuk *SMOD = 1*.

Tabel 2.7. Setting baud rate [6]

Baud Rate	Fosc (MHz)	SMOD	TH1
150	11.0592	0	40H
300		0	A0H
600		0	D0H
1200		0	E8H
2400		0	F4H
4800		0	FAH
9600		0	FDH
19200		1	FDH
57600		1	FFH

Rumus untuk mencari *Baud rate*:

SMOD 0

$$\text{Baud rate} = \text{On-chip oscillator} / 12 \times (256 - \text{TH1}) \times 32 \quad \dots \quad (2.1)$$

SMOD 1

$$\text{Baud rate} = \text{On-chip oscillator} / 12 \times (256 - \text{TH1}) \times 16 \quad \dots \quad (2.2)$$

BAB III

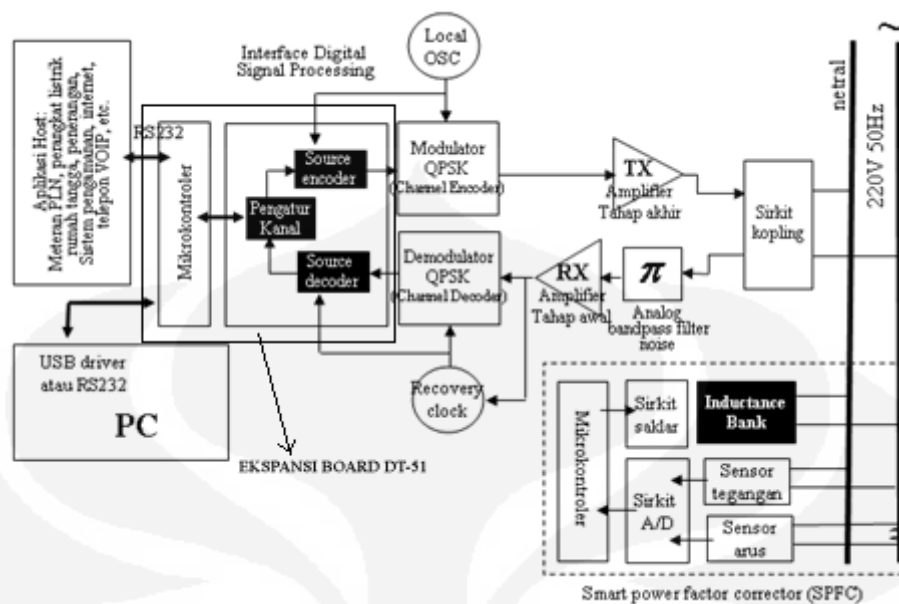
PERANCANGAN, PENGOPERASIAN DAN PEMROGRAMAN

Dalam perancangan komunikasi data *serial* diperlukan perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat keras yang digunakan terdiri dari komputer *PC*, kabel *RS-232C*, minimum sistem mikrokontroler dan *power supply*. Agar *Hardware* dapat berkomunikasi secara *serial*, maka diperlukan perangkat lunak/*software* yang berfungsi untuk mengendalikan aliran data dari komputer *PC* ke mikrokontroler *AT89S51* atau sebaliknya dari mikrokontroler *AT89S51* ke komputer *PC*.

3.1 HARDWARE

3.1.1 Rangkaian Blok Modem PLC

Untuk mem-*buffer* data informasi maupun suara yang berasal dari komputer *PC* diperlukan suatu blok sirkit pengendali. Blok sirkit pengendali ini yang diberi nama *interface* ekspansi *board* *DT-51* yang berfungsi sebagai *buffer* antara komputer *PC* dengan *modem QPSK*. Blok sirkit pengendali tersebut dirancang dan dibangun berbasis pada sistem mikrokontroler *AT89S51*. Sistem mikrokontroler ini dipilih karena harganya murah dan diperhitungkan mampu untuk memenuhi kebutuhan. Blok pengendali *interface* ekspansi *board* *DT-51* tersebut dirancang dan diprogram untuk dapat melayani komunikasi data *serial* komputer *PC* dengan menggunakan protokol *asynchronous*. Dalam komunikasi data *serial*, *interface* ekspansi *board* *DT-51* ini menggunakan dua metode untuk aliran data yaitu metode *handshaking* dan tanpa metode *handshaking*. Blok diagram lengkap dari *interface* ekspansi *board* *DT-51* yang digabungkan dengan *modem QPSK* dapat dilihat pada Gambar 3.1.



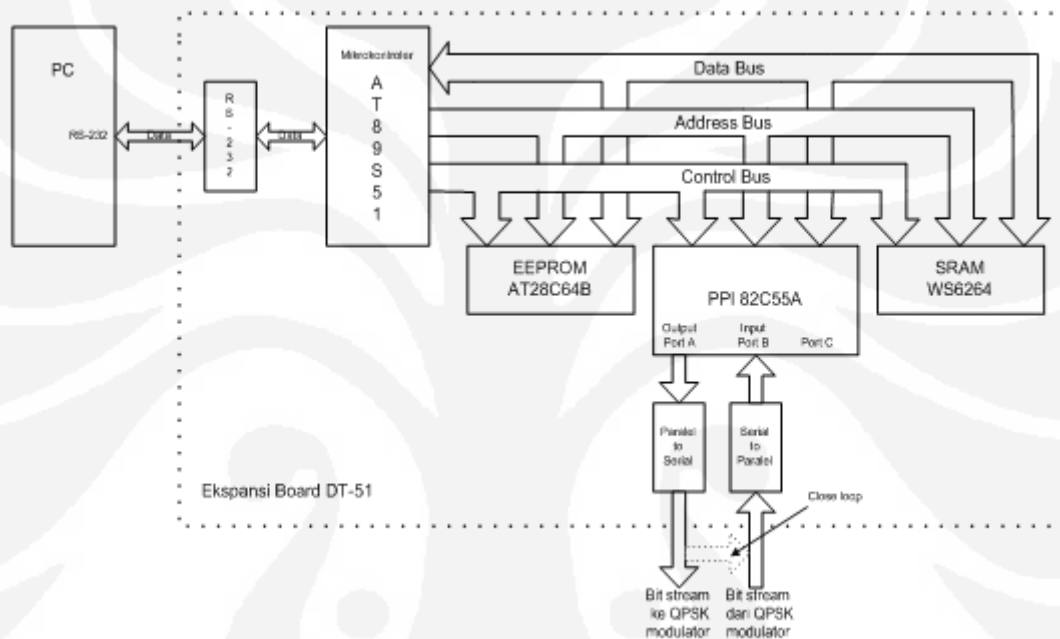
Gambar 3.1. Blok diagram *modem PLC*.

3.1.2 Rangkaian Blok Ekspansi Board DT-51

Untuk perancangan komunikasi *serial port* ini, penulis menggunakan modul DT-51 *minimum system* mikrokontroler versi 3.3 buatan *Innovative Electronics* (www.InnovativeElectronics.com). DT-51 berbasis mikrokontroler *AT89S51* yang merupakan standar industri. DT-51 memiliki *RS-232C serial port interface* yang digunakan untuk komunikasi dengan komputer *PC*, memori *nonvolatile EEPROM AT28C64B* untuk menyimpan data, konektor (alamat, kontrol, dan data) untuk ekspansi rangkaian dan 4 *port input/output (I/O)* yang tiap *port* terdiri dari 8-bit. Tata letak komponen pada *board* DT-51 dapat dilihat pada Gambar 3.2. Pada modul DT-51 tersebut, komunikasi data *serial* hanya dapat dilakukan tanpa *handshaking* (Tx, Rx dan *signal ground*) dan memori *EEPROM (AT6264B)* tidak dapat melayani penyimpanan data per blok dengan kecepatan tinggi karena proses penyimpanan lebih lambat dari data yang diterima sehingga data yang baru datang tidak tersimpan karena karakter yang lebih dulu tiba belum selesai diproses. Akibat dari lambatnya proses penyimpanan data pada memori *EEPROM*, data yang tersimpan pada memori tersebut adalah data yang rusak atau tidak sama dengan data yang dikirim oleh komputer *PC*.

Guna mengatasi masalah tersebut, penulis melakukan ekspansi dengan menambahkan pada sistem mikrokontroler tersebut *SRAM (WS6264)* sebesar 6

metode *handshaking*. Umumnya metode tanpa *handshaking* digunakan untuk komunikasi dengan kecepatan rendah, misalnya untuk aplikasi *user chatting*, pengiriman hasil pengukuran meter listrik, temperatur ruang, status pintu, absensi dlsb. Sedangkan metode *handshaking* digunakan untuk komunikasi dengan kecepatan yang tinggi misalnya untuk *transfer* suara, *video*, *file* dan sebagainya. Agar kecepatan antara medium dan komputer *PC* dapat disinkronkan maka dilakukan *buffering* data oleh blok *interface* ekspansi *board DT-51*.



Gambar 3.3. Blok diagram komunikasi data *serial* antara komputer *PC* dengan ekspansi *board DT-51* melalui *interface RS-232C*.

Buffering data tersebut menggunakan *SRAM WS6264* yang besarnya *6 kbytes*. Sedangkan hubungan antara blok *interface* ekspansi *board DT-51* dengan *modem QPSK* adalah melalui rangkaian terpadu *AT89C2051* yang berfungsi sebagai pengubah data *parallel to serial* dan data *serial to parallel*.

3.1.3 Rancangan Pengalamatan Komponen Sistem Mikrokontroler

Agar tidak terjadi kesalahan dalam pengaksesan rangkaian terpadu (*IC*) pada *board DT-51* dan *board* ekspansi, diperlukan pengalamatan pada masing-masing rangkaian terpadu. Dekoder pengalamatan rangkaian terpadu (*IC*) *PPI 82C55A* dan memori yang digunakan dapat dilihat pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1. Dekoder pengalamatan *PPI 82C55A*.

A15	A14	A13	A12	A11-A2	A1	A0	Hex	Alamat Tujuan
0	0	1	0	0	0	0	2000H	Port A
0	0	1	0	0	0	1	2001H	Port B
0	0	1	0	0	1	0	2002H	Port C
0	0	1	0	0	1	1	2003H	Control Word

Tabel 3.2. Dekoder pengalamatan memori.

A15	A14	A13	A12	A11-A0	Hex	Memori
0	1	0	0	0	4000H	AT28C64B
0	1	0	1	1	5FFFH	
0	1	1	0	0	6000H	WS6264
0	1	1	1	1	7FFFH	

Dalam mengakses dan menempatkan data program pada alamat yang benar agar tidak terjadi *crash* dan berjalan sesuai dengan kebutuhan, dapat dilihat susunan alamat peta memori pada Gambar 3.4.

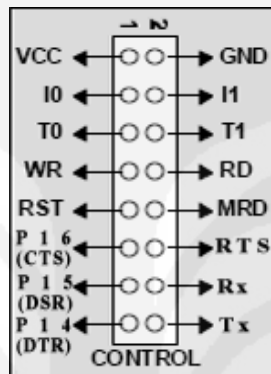
Flash Memori AT89S51	0000H 1FFFH
PPI 82C55A	2000H 3FFFH
EEPROM AT28C64B (8K)	4000H 5FFFH
SRAM WS6264 (8K)	6000H 7FFFH

Gambar 3.4. Peta memori.

3.1.4 Handshaking

Pada penelitian tugas akhir ini, penulis menggunakan metode *handshaking* untuk komunikasi data *serial asynchronous* antara ekspansi board DT-51 dengan komputer PC. Agar komunikasi data *serial asynchronous* dengan metode *handshaking* dapat berfungsi sesuai dengan kebutuhan, penulis melakukan perubahan dengan menambahkan fungsi kontrol pada konektor *CONTROL*. Dimana pin *RTS* adalah *output* dari *pin-7* rangkaian terpadu (IC) *ICL232* yang dikendalikan oleh P1.7, *pin Tx* dan *Rx* dikendalikan oleh rangkaian terpadu *ICL232*. Sedangkan *pin CTS* dihubungkan ke *pin-13* rangkaian terpadu (IC)

MAX-232CPE yang *outputnya* pada *pin-12* diumpankan ke P1.6, *pin DSR* dihubungkan ke *pin-8* rangkaian terpadu (IC) *MAX-232CPE* yang *outputnya* dihubungkan ke P1.5 dan *pin DTR* dihubungkan ke *output pin-7* pada rangkaian terpadu (IC) *MAX-232CPE* yang dikendalikan oleh P1.4. Perubahan/penambahan fungsi kontrol ini dapat dilihat pada Gambar 3.5.



Gambar 3.5. Konektor *control header pin*.

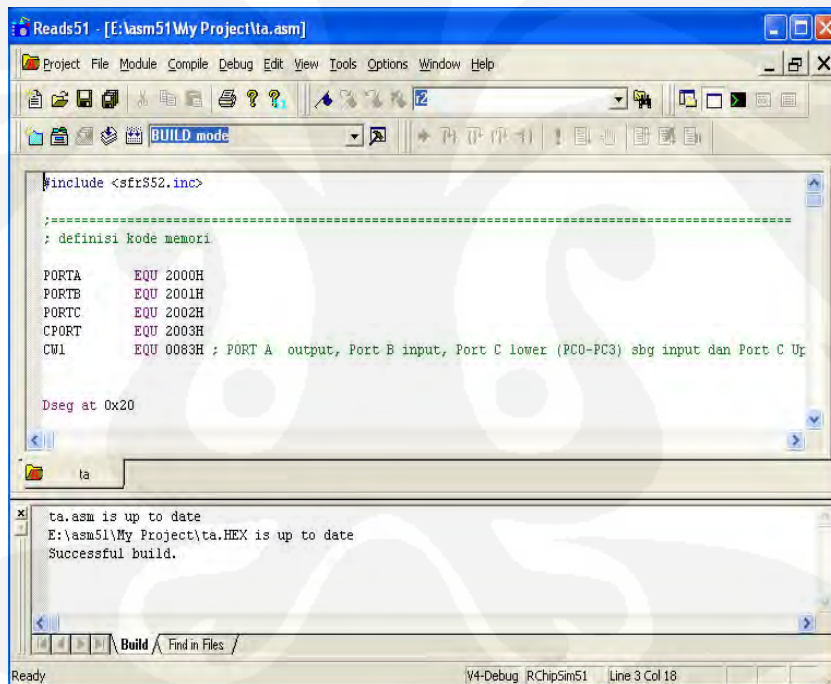
Agar komunikasi data *serial asynchronous* dengan menggunakan metode *handshaking* dapat terjalin diperlukan pengkabelan untuk menghubungkan *RS-232C* antara komputer *PC* dengan modul ekspansi *board DT-51*. Pada pengkabelan *RS-232C* ini menggunakan 7 kabel, *pin-pin* pada *RS-232C* yang dipakai yaitu: *pin-3 (TxD)*, *pin-2 (RxD)*, *pin-4 (DTR)*, *pin-5 (Signal Ground)*, *pin-6 (DSR)*, *pin-7 (RTS)* dan *pin-8 (CTS)*. Pengkabelan *RS-232C* dapat dilihat pada Tabel 3.3.

Tabel 3.3. Pengkabelan *RS-232C*.

RS-232C Komputer PC DB-9 male	RS-232C Modul Ekspansi DT-51 DB-9 female	Keterangan
3	2	<i>TxD - RxD</i>
2	3	<i>RxD - TxD</i>
4	6	<i>DTR - DSR</i>
7	8	<i>RTS - CTS</i>
6	4	<i>DSR - DTR</i>
8	7	<i>CTS - RTS</i>
5	5	<i>Signal Ground</i>

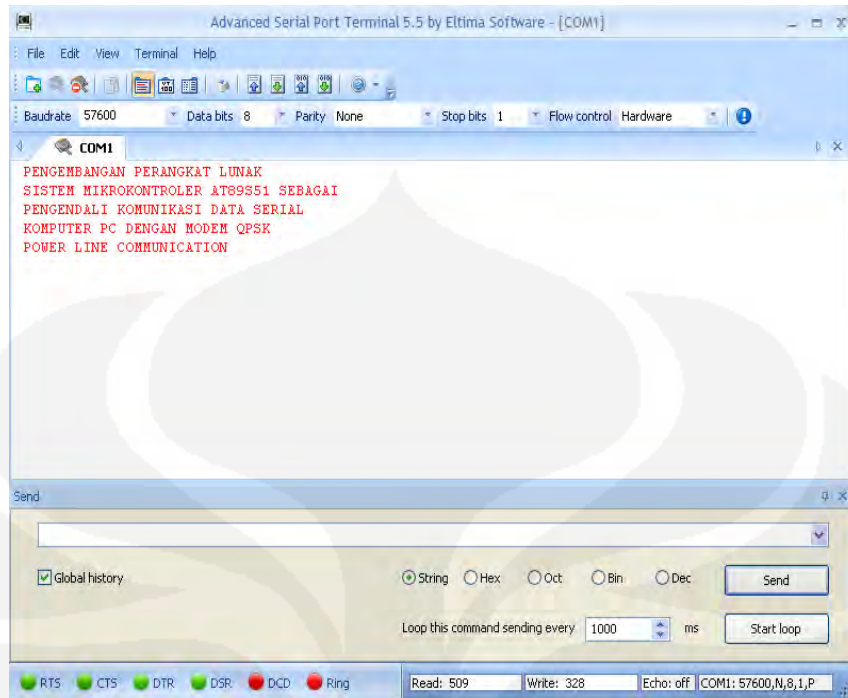
3.2 SOFTWARE

Untuk mengendalikan komunikasi data melalui *serial port* dikomputer PC diperlukan perangkat lunak atau *software*. Ada 3 (tiga) jenis perangkat lunak yang dipergunakan dalam tugas akhir ini. Perangkat lunak pertama yaitu perangkat lunak *assembler-51* yang digunakan untuk mengoperasikan mikrokontroler *AT89S51* sebagai *CPU* pada sistem *minimum* guna mengendalikan aliran data *serial asynchronous* dari komputer PC ke *modem QPSK* atau sebaliknya dari *modem QPSK* ke komputer PC. Perangkat lunak *assembler-51* ini dapat dilihat di lampiran. Perangkat lunak kedua adalah *Reads51*, yaitu *editor* yang dipergunakan untuk menulis dan kompilasi bahasa *assembler-51* menjadi *file hex*. Tampilan perangkat lunak tersebut terlihat pada Gambar 3.6 berikut ini.



Gambar 3.6. Reads51

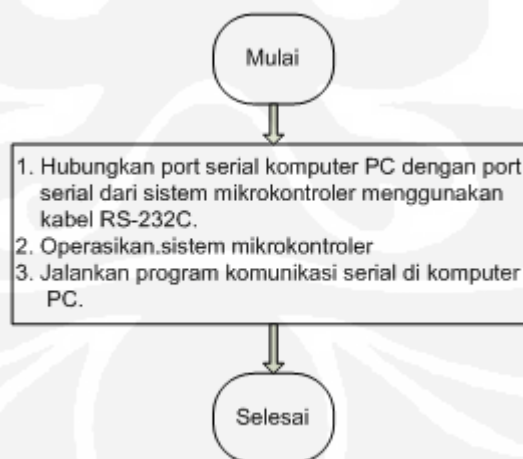
Perangkat lunak ketiga adalah *Advanced Serial Port Terminal 5.5* yang digunakan untuk mengendalikan komunikasi *serial port* komputer PC dengan mikrokontroler *AT89S51*, seperti terlihat Gambar 3.7 berikut ini.



Gambar 3.7. *Advanced serial port terminal 5.5.*

3.2.1 Flowchart Pengoperasian RS-232C

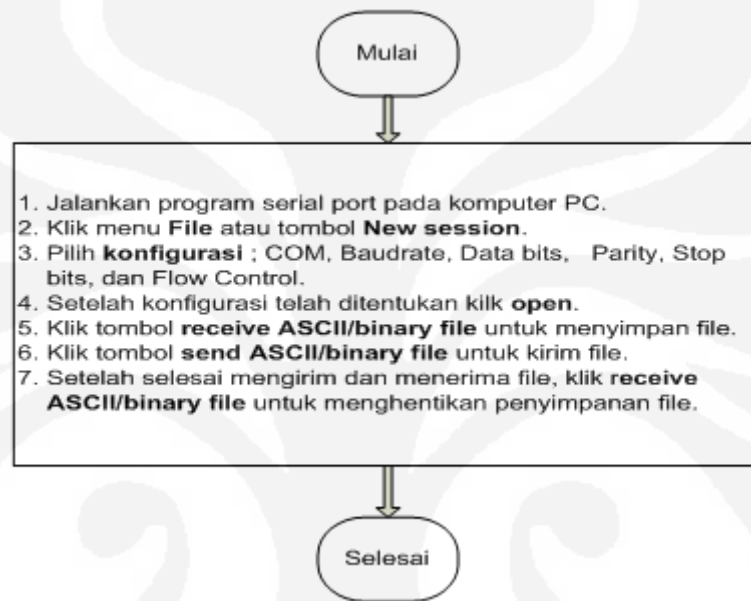
Tahapan-tahapan yang perlukan dalam pengoperasian *RS-232C* antara komputer *PC* dengan Ekspansi *board DT-51*, dapat dilihat pada Gambar 3.8.



Gambar 3.8. *Flowchart pengoperasian RS-232C.*

3.2.2 Flowchart Pengoperasian Software Serial Port Pada Komputer PC

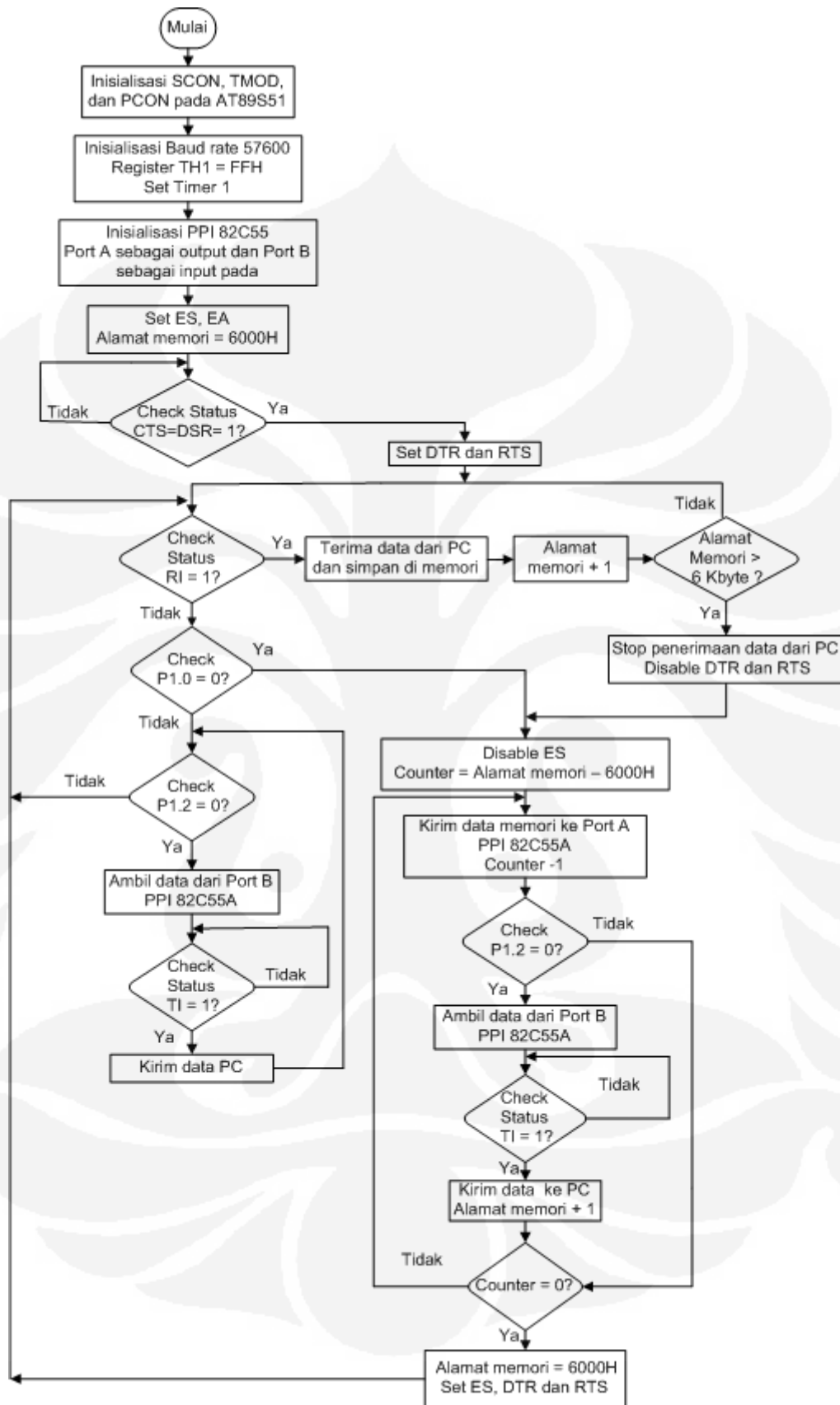
Pada komunikasi data *serial asynchronous* ini, penulis menggunakan konfigurasi sebagai berikut ; *COM 1, Baudrate 57600, 8 Data bits, Parity None, Stop bits 1*, dan *Flow control: Hardware*. Tahapan konfigurasi untuk pengoperasian *software Advanced Serial Port Terminal 5.5* dijelaskan lebih rinci pada *flowchart* Gambar 3.9.



Gambar 3.9. *Flowchart software serial port* pada komputer PC.

3.2.3 Flowchart Komunikasi Data Serial pada Mikrokontroler AT89S51

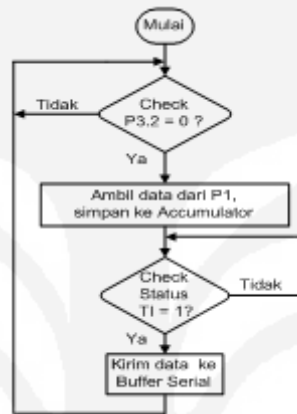
Mikrokontroler *AT89S51* dapat berkomunikasi secara *serial asynchronous* dengan komputer *PC*, terlebih dahulu harus diprogram dengan menggunakan bahasa *assembler-51*. Komunikasi data *serial asynchronous* antara mikrokontroler *AT89S51* dan komputer *PC* dilakukan dengan *full-duplex*, dimana dapat mengirim dan menerima data pada waktu bersamaan. *Full-duplex* juga dilakukan pada *I/O PPI 82C55A* yaitu *port A* dan *port B*. Untuk mempermudah dalam pembuatan program sistem mikrokontroler *AT89S51* yang berfungsi sebagai *CPU*, penulis menggunakan *flowchart*. Alur diagram dari pemrograman mikrokontroler *AT89S51* ini dapat dilihat pada Gambar 3.10.



Gambar 3.10. Flowchart komunikasi data serial pada mikrokontroler AT89S51.

3.2.4 Flowchart Komunikasi Data Paralel ke Serial Pada Mikrokontroler AT89C2051

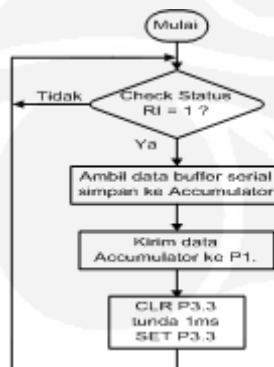
Komunikasi data ini menggunakan mikrokontroler *AT89C2051* yang berfungsi sebagai pengubah data paralel *port A PPI 82C55A* ke data *serial*. Alur program dari komunikasi data paralel ke *serial* dapat dilihat pada Gambar 3.11.



Gambar 3.11. *Flowchart* komunikasi data paralel ke *serial* pada mikrokontroler *AT89C2051*.

3.2.5 Flowchart Komunikasi Data Serial ke Paralel Pada Mikrokontroler AT89C2051

Komunikasi data ini menggunakan mikrokontroler *AT89C2051* yang berfungsi sebagai pengubah data *serial* ke data paralel yang diumpankan ke *port B PPI 82C55A*. Alur program dari komunikasi data *serial* ke paralel dapat dilihat pada Gambar 3.12.



Gambar 3.12. *Flowchart* komunikasi data *serial* ke paralel pada mikrokontroler *AT89C2051*.

3.2.6 Inisialisasi Serial Port AT89S51

Sebelum memulai komunikasi data *serial asynchronous* terlebih dahulu harus menginisialisasi *register-register* dengan benar, *Receive enable/penerimaan* karakter pada *RxD* di *set* menggunakan *register SCON*. Untuk *baud rate* yang digunakan adalah *register TMOD, PCON* dan memberikan nilai *register TH1* untuk *timer counter*.

```
MOV SCON,#52H      ; serial port mode 1
MOV TMOD,#20H     ; timer 1, mode 2 (8 bit auto reload)
MOV PCON,#80H     ; SMOD 1
MOV TH1,#FFH      ; reload count untuk 57600 baudrate.
SETB TR1          ; aktifkan timer 1
```

Untuk mengirim data ke komputer *PC* terlebih dahulu harus memeriksa apakah *TI flag register* 1 atau 0, yang memberitahukan apakah semua karakter telah terkirim atau *buffer* pengirim kosong.

```
JNB TI,$           ; periksa apakah buffer empty.
CLR TI             ; clear TI flag.
MOV SBUF,A         ; kirim karakter ke komputer PC.
```

Dan untuk menerima data dari komputer *PC* terlebih dahulu memeriksa *RI flag register* yang menunjukkan bahwa data telah diterima dengan lengkap atau *buffer* penerima penuh.

```
JNB RI,$          ; periksa apakah ada data yang diterima.
CLR RI            ; clear flag register.
MOV A,SBUF        ; baca data dari UART.
```

3.2.7 Baca dan Tulis Memori Eksternal WS6264

Untuk penyimpanan dan pembacaan data yang diterima dari komputer *PC* ke memori eksternal *SRAM* mikrokontroler *AT89S51* menggunakan instruksi:

```
MOVX @DPTR,A      ; simpan data di memori
MOVX A,@DPTR      ; baca data di memori
```

DPTR adalah *register pointer 16-bit* yang digunakan sebagai alamat memori eksternal.

3.2.8 Inisialisasi PPI 82C55A

Menginisialisasi *IC 82C55A* sangat diperlukan, agar *I/O PPI 82C55A* dapat berfungsi sesuai dengan kebutuhan. Untuk memudahkan dalam pemrograman terlebih dahulu alamat dari tiap *port* didefinisikan menggunakan *EQU*.

```
PORTA EQU 2000H      ; port A
PORTB EQU 2001H      ; port B
PORTC EQU 2002H      ; port C
CPORT EQU 2003H      ; control word
CW1 EQU 0083H        ; nilai control word
```

Mode yang digunakan *mode 0* yang nilai *control word* adalah 83H; *port A* sebagai *output*, *port B* sebagai *input*, *port C lower (PC0-PC3)* sebagai *input* dan *port C Upper (PC4-PC7)* sebagai *output*. Instruksi untuk *PPI 82C55A* ini adalah sebagai berikut:

init_8255:

```
MOV DPTR,#CPORT
MOV A,#CW1
MOVX @DPTR,A
RET
```

Untuk mengeluarkan dan menerima data pada *port* instruksinya adalah sebagai berikut:

Out_8255:

```
MOV DPTR,#PORTA      ; alamat port A
MOVX @DPTR,A         ; keluarkan data ke port A (output)
RET
```

In_8255:

```
MOV DPTR,#PORTB      ; alamat port B
MOVX A,@DPTR         ; ambil data pada port B dan simpan
RET                  ; dalam ACC.
```

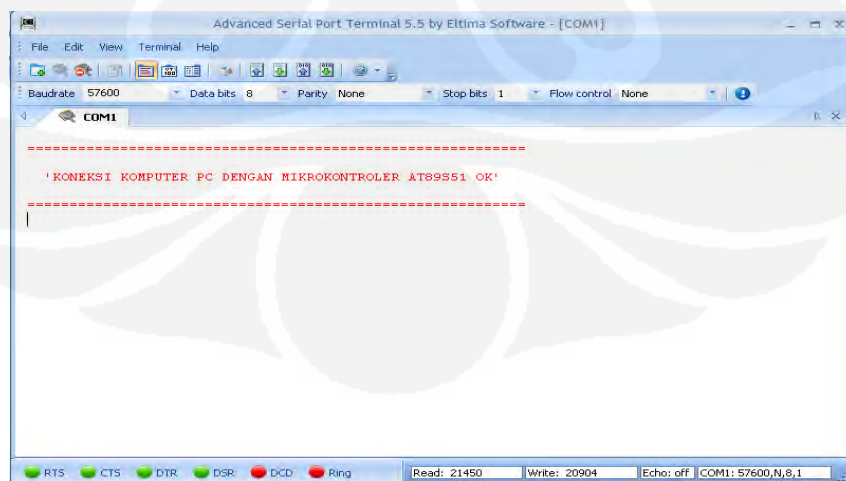
BAB IV

HASIL EKSPERIMEN DAN ANALISA

Dalam penelitian tugas akhir ini telah dilakukan tahap eksperimen *loop* tertutup (dimana data *Tx* dikembalikan ke *Rx*). Kedua tahapan eksperimen yang telah dilakukan adalah sebagai berikut :

- 1). Tahap I, penulis menggunakan komunikasi data *serial asynchronous* tanpa metode *handshaking* dimana hanya 3 kabel yang digunakan untuk komunikasi ini, yaitu : *Tx*, *Rx* dan *signal ground*. Dalam tahap ini digunakan memori *EEPROM AT28C64B* yang tersedia di minimum sistem mikrokontroler.
- 2). Tahap II, penulis menggunakan komunikasi data *serial asynchronous* dengan metode *handshaking* dan menggunakan *SRAM WS6264* sebagai *buffer* data.

Pada awal komunikasi antara komputer *PC* dengan mikrokontroler *AT89S51*, terlebih dahulu status koneksi akan diperiksa. Bila koneksi berhasil dilakukan, pada *software editor* di komputer *PC* akan tampil pesan 'KONEKSI KOMPUTER PC DENGAN AT89S51 OK'. Besar ukuran memori eksternal mikrokontroler *AT89S51* yang digunakan sebagai *buffer* data adalah 6 *kbyte*, Tampilan status koneksi ini dapat dilihat pada Gambar 4.1.

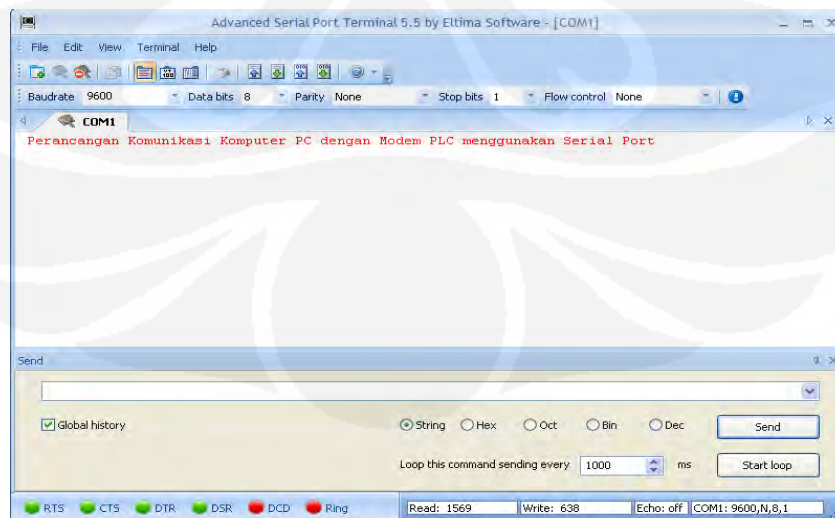


Gambar 4.1. Status koneksi komputer *PC* dengan mikrokontroler *AT89S51*.

Pada komunikasi data *serial asynchronous* ini dapat dilakukan tanpa metode *handshaking (flow control: none)* atau menggunakan metode *handshaking (flow control:Hardware)*. Metode tanpa *handshaking* dapat digunakan apabila data karakter yang dikirim dan diterima lebih kecil dari *buffer* memori eksternal pada sistem mikrokontroler *AT89S51 (6 kbyte)*. Sedangkan metode *handshaking* dapat digunakan untuk mengirim dan menerima data karakter yang lebih kecil atau lebih besar dari *buffer* memori eksternal (*6 kbyte*) pada sistem mikrokontroler *AT89S51*.

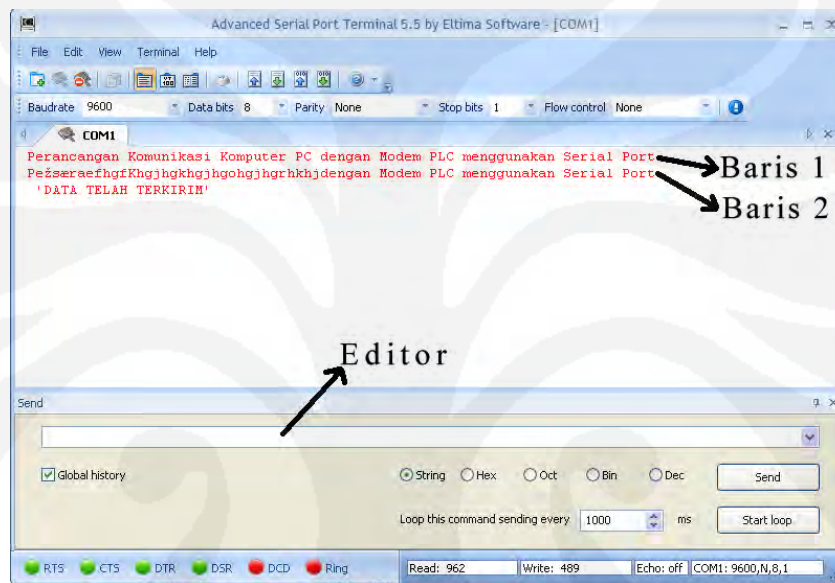
4.1 KOMUNIKASI DATA SERIAL TANPA METODE HANDSHAKING

Pada eksperimen ini, komunikasi data *serial asynchronous* dilakukan tanpa menggunakan metode *handshaking (flow control: none)*, dimana pengiriman dan penerimaan data karakter dari komputer *PC* melalui mikrokontroler *AT89S51* yang datanya dikeluarkan ke *port A PPI 82C55A* kemudian diterima oleh *port B PPI 82C55A* dan dikirimkan kembali ke komputer *PC*. Pengiriman dan penerimaan data karakter dilakukan dengan mengetikkan satu per satu karakter pada *software editor* menggunakan *keyboard*. Karakter yang diketikkan tersebut terkirim dari *keyboard* ke *RS-232C* kemudian melewati sistem mikrokontroler *AT89S51* *loop* tertutup selanjutnya ditampilkan pada *software editor*. Hasil pengiriman dan penerimaan data per karakter dari komputer *PC* ke mikrokontroler *AT89S51* dapat dilihat pada Gambar 4.2.



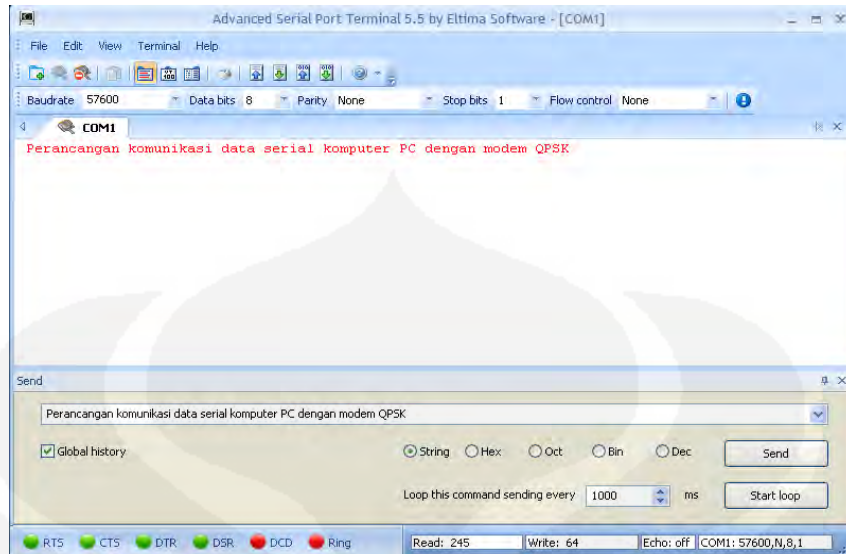
Gambar 4.2. Pengiriman dan penerimaan data pada komputer *PC*.

Pada eksperimen tahap I, data karakter ini dikirim per blok satu baris melalui *editor* dibagian bawah pada *software editor* yang besarnya *75 byte*. Hasil pengiriman dan penerimaan data mengalami kerusakan dibaris ke-2 pada Gambar 4.3. Data yang diterima dikomputer *PC* mengalami kerusakan yang disebabkan karena proses penyimpanan data memori *EEPROM AT28C64B* lebih lambat dari data yang diterima. Akibatnya data karakter yang datang berikutnya tidak tersimpan karena data yang lebih dulu tiba belum selesai diproses.



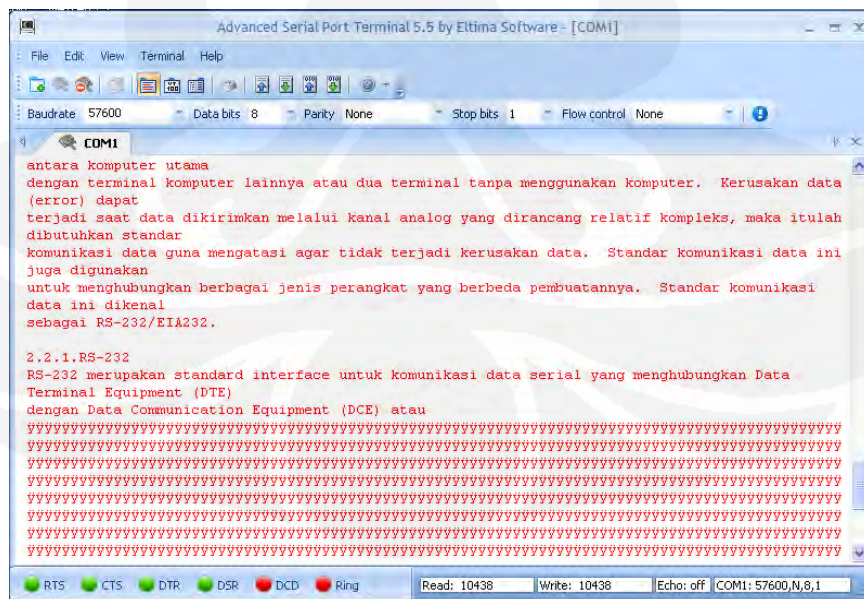
Gambar 4.3. Kerusakan pengiriman dan penerimaan data per blok *75 byte*.

Pada eksperimen tahap II ini, pengiriman dan penerimaan data pada mikrokontroler *AT89S51* dilakukan secara blok dengan ukuran data *64 byte* lebih kecil dari *buffer* memori *SRAM WS6264* pada sistem mikrokontroler *AT89S51* yang besarnya *6 kbyte*. Data karakter dikirim per blok satu baris yang dituliskan pada *editor* bagian bawah pada *software editor*. Pengiriman data karakter secara blok ini tidak mengalami kerusakan data karena *buffer* memori menyimpan data lebih cepat dari data yang datang berikutnya sehingga tidak ada data hilang atau semua data karakter tersimpan pada *buffer* memori eksternal *SRAM WS6264*. Hasil eksperimen ini dapat dilihat pada Gambar 4.4.



Gambar 4.4. Pengiriman dan penerimaan data per blok 64 byte tanpa metode *handshaking*.

Hasil eksperimen tahap II dengan pengiriman dan penerimaan data lebih besar dari *buffer* memori pada mikrokontroler AT89S51 yang besarnya 6 *kbyte*, dapat dilihat pada Gambar 4.5. Komunikasi data *serial asynchronous* dari komputer PC ke mikrokontroler AT89S51 dilakukan secara blok (*file*) dengan besar ukuran data 10.1 *kbyte* dengan *format text* (*.txt). Hasil dari pengiriman dan penerimaan *file* data ini mengalami kerusakan.



Gambar 4.5. Pengiriman dan penerimaan data secara blok dengan ukuran 10.1 *kbyte* tanpa metode *handshaking*.

Dari hasil eksperimen komunikasi data *serial asynchronous* tanpa menggunakan metode *handshaking (flow control:none)*, kerusakan data karakter yang dikirim dan diterima dapat disebabkan oleh beberapa hal yaitu sebagai berikut :

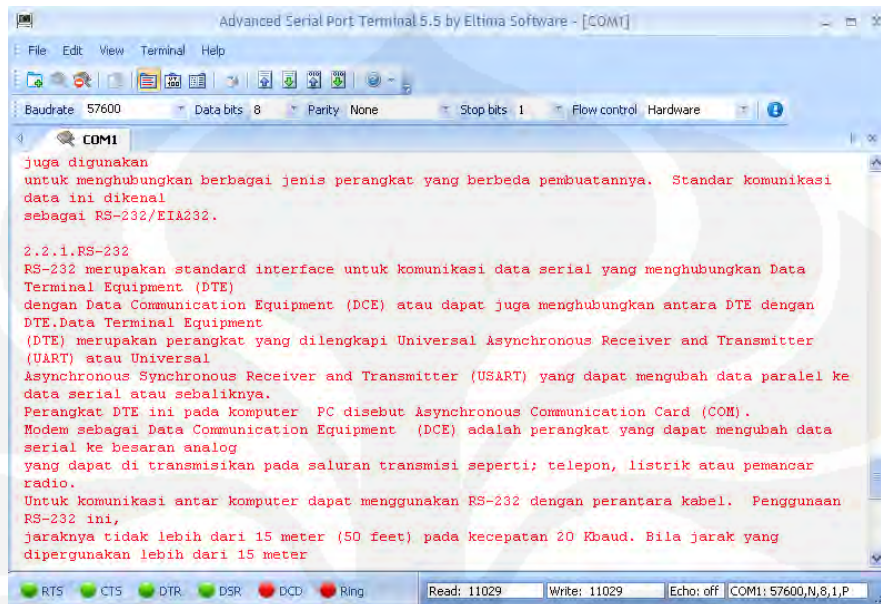
- 1). Proses penyimpanan pada memori, dimana penyimpanan yang dilakukan oleh memori belum selesai sepenuhnya sehingga karakter yang baru datang tidak tersimpan/hilang.
- 2). Kapasitas *buffer* memori eksternal penyimpanan pada mikrokontroler *AT89S51* lebih kecil dari jumlah data yang diterima dari komputer *PC* sehingga data yang berikutnya tidak tersimpan atau hilang karena komputer *PC* terus mengirim data hingga selesai.
- 3). Tidak terjadi sinkronisasi antara pengirim dan penerima ketika pengiriman data dilakukan secara blok, yang dikarenakan tidak tersedianya jalur instruksi/perintah (*handshaking*) untuk memberitahukan komputer *PC* agar menghentikan sementara proses pengiriman ke mikrokontroler *AT89S51* ketika memori data telah penuh.

4.2 KOMUNIKASI DATA SERIAL DENGAN METODE HANDSHAKING

Untuk mengirim dan menerima data yang sangat besar atau lebih besar dari *buffer* memori mikrokontroler *AT89S51* (6 *kbyte*) tanpa adanya kerusakan data, maka pada eksperimen tahap II penulis menggunakan metode *handshaking (flow control:Hardware)* untuk menghentikan sementara proses pengiriman data dikomputer *PC* ke *buffer* memori sistem mikrokontroler *AT89S51* ketika telah penuh dengan men-*disable CTS* dan *DSR* pada komputer *PC*. Setelah semua data *buffer* memori pada sistem mikrokontroler *AT89S51* dikirimkan keluar ke *port A PPI 82C55A*, mikrokontroler akan mengambil/menerima data berikutnya di komputer *PC* dengan mengirimkan sinyal *DTR* dan *RTS* yang mendahului pengiriman dan penerimaan data.

Pengiriman dan penerimaan data dilakukan secara blok dengan metode *handshaking* menggunakan *loop* tertutup dari komputer *PC* ke mikrokontroler

AT89S51 yang datanya dikeluarkan ke *port A* kemudian diterima oleh *port B PPI 82C55A* dan dikirimkan kembali ke komputer *PC*.



Gambar 4.6. Pengiriman dan penerimaan *file* data 10.1 *kbyte* dengan metode *handshaking*.

Pada komunikasi data *serial asynchronous* ini, data yang dikirim dan diterima ukurannya lebih besar dari *buffer* memori eksternal *SRAM WS6264* pada sistem mikrokontroler *AT89S51* (6 *kbyte*). Besar ukuran *file* data yang dikirim dan diterima adalah 10.1 *kbyte* dengan *format text* (*.txt). Hasil dari proses komunikasi data *serial asynchronous* ini tidak mengalami kerusakan data yang dikirim dan diterima oleh komputer *PC* terlihat pada Gambar 4.6.

Dari hasil komunikasi data *serial asynchronous* dengan menggunakan metode *handshaking* (*flow control:Hardware*), tidak terjadi kerusakan data yang dikirim dan diterima karena komunikasi antara komputer *PC* dan sistem mikrokontroler *AT89S51* telah sinkron. Dengan menggunakan metode *handshaking*, penulis dapat mengirim dan menerima beragam format data seperti *.txt, *.jpg, *.gif, *.pdf, *.doc, *.mp3, dan sebagainya dengan besar ukuran *file* bervariasi.

BAB V

KESIMPULAN

5.1 KESIMPULAN

Dari data yang didapat selama penelitian dan penulisan laporan ini, penulis dapat menyimpulkan sebagai berikut:

1. Komunikasi data *serial asynchronous* tanpa metode *handshaking* tidak dapat melayani pengiriman dan penerimaan data yang melampui kapasitas *buffer* memori yang tersedia.
2. Komunikasi *serial asynchronous* dapat terjalin dengan baik, dengan syarat bahwa *baud rate* kedua peralatan yang berkomunikasi harus sama dan bila *baud rate* tidak sama, maka data yang dikirim atau diterima tidak sinkron.
3. Jika *baud rate* antara pengirim/penerima lebih tinggi harus ada suatu protokol yang mengatur agar pengiriman dapat ditunda sampai penerima siap menerima data selanjutnya. Karena hampir tidak mungkin kedua perangkat memiliki *baud rate* yang sama, maka *transfer* blok data harus dilakukan secara *synchronous* antara kedua perangkat (*PC & modem*) tersebut dengan memanfaatkan fasilitas *hardware synchronous* pada *RS-232C*.
4. Untuk pengoperasian komunikasi *serial asynchronous* pada mikrokontroler *AT89S51*, terlebih dahulu harus menginisialisasi *register SCON, TMOD, PCON* dan *register TH1* untuk menentukan *baud rate*. Penulisan dan pembacaan data pada memori *EEPROM*, harus diberikan waktu tunda yang cukup agar data yang dibaca tidak salah setelah proses penulisan. Kecepatan proses penulisan dan pembacaan *SRAM* lebih cepat dari *EEPROM*.
5. Agar komunikasi data *serial* komputer *PC* dengan mikrokontroler *AT89S51* sinkron dalam mengirim dan menerima data yang sangat besar dari *buffer* memori eksternal sistem mikrokontroler *AT89S51*, harus menggunakan metode *handshaking* (*DTR, RTS, DSR dan CTS*) untuk menghentikan sementara proses pengiriman/penerimaan data saat *buffer* memori penuh. Selanjut mengaktifkan kembali *handshaking* untuk mengambil/mengirim data berikutnya setelah semua *buffer* memori digunakan atau telah kosong.

DAFTAR ACUAN

- [1] Douglas V. Hall, *The Mikroprocessor and Interfacing Programming and Hardware Second Edition* (Mc-Graw Hill International., 1992).
- [2] Harold S.Stone, *Microcomputer Interfacing* (Addison-Wesley., 1983).
- [3] -----, *WS6264 High Speed Super Low Power SRAM*.
http://www.datasheetcatalog.org/datasheets2/32/322965_1.pdf
- [4] -----, *82C55A CMOS Programmable Peripheral Interface* (Harris Corporation., 1996).
<http://www.qsl.net/yo5ofh/doc/8255a.pdf>
- [5] I.Scott Mackenzie, *THE 8051 Microcontroller Third Edition* (New Jersey: Prentice Hall., 1999).
- [6] -----, *Atmel 8051 Microcontrollers Hardware Manual* (Atmel Corporation., 1982).
http://www.atmel.com/dyn/resources/prod_documents/doc4316.pdf

DAFTAR PUSTAKA

- [1] Douglas V. Hall, *The Mikroprocessor and Interfacing Programming and Hardware Second Edition* (Mc-Graw Hill International., 1992).
- [2] -----, *64K (8K x 8) Parallel EEPROM with Page Write and Software Data Protection AT28C64B* (Atmel Corporation., 1998).
http://www.atmel.com/dyn/resources/prod_documents/doc0270.pdf
- [3] -----, *82C55A CMOS Programmable Peripheral Interface* (Harris Corporation., 1996).
<http://www.qsl.net/yo5ofh/doc/8255a.pdf>
- [4] I.Scott Mackenzie, *THE 8051 Microcontroller Third Edition* (New Jersey: Prentice Hall., 1999) .
- [5] -----, *Atmel 8051 Microcontrollers Hardware Manual* (Atmel Corporation., 1982).
http://www.atmel.com/dyn/resources/prod_documents/doc4316.pdf
- [6] -----, *8-bit Microcontroller with 4K Bytes In-System Programmable Flash AT89S51* (Atmel Corporation., 2005).
http://www.atmel.com/dyn/resources/prod_documents/doc2487.pdf
- [7] M.Morris Mano, *Computer system Architecture Second Edition* (New Jersey: Prentice Hall., 1982).
- [8] Vijay Ahuja, *Design and Analysis of Computer Communication Networks* (Mc-Graw Hill., 1982).
- [9] John Uffenbeck, *Microcomputer And Microprocessor Second Edition* (New Jersey: Prentice Hall., 1991).
- [10] -----, *ICL232 +5V Powered Dual RS-232 Transmitter/Receiver* (Harris Corporation., 1993).
<http://www.intersil.com/data/fn/fn3020.pdf>
- [11] -----, *MAX232 +5V-Powered Multichannel RS-232 Drivers/Receivers* (Maxim Integrated Products., 2006).
http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1798/t/al
- [12] -----, *WS6264 High Speed Super Low Power SRAM*.
http://www.datasheetcatalog.org/datasheets2/32/322965_1.pdf

LAMPIRAN 1

2.2.1 RS-232C

Pin 1 : *Receive line signal detector (RLSD)* atau data *carrier detect (DCD)*
CD sinyal yang menentukan bahwa penerima siap berhubungan dengan *DTE*.

Pin 2 : *Receive Data (RxD)*
Merupakan *output* dari *DCE* ke *DTE*, yaitu jalur pengiriman data *DCE* ke *DTE* yang akan beroperasi dalam arah *half-duplex*.

Pin 3 : *Transmitted Data (TxD)*
Sebagai jalur pengiriman data dari *DTE* ke *DCE* dimana akan berlogika 0 (*mark*), apabila tidak ada data yang ditransmisikan.

Pin 4 : *Data Terminal Ready (DTR)*
Untuk mengaktifkan sinyal *DTE*, yang menginformasikan pada *DCE* bahwa *DTE* telah aktif dan siap berhubungan.

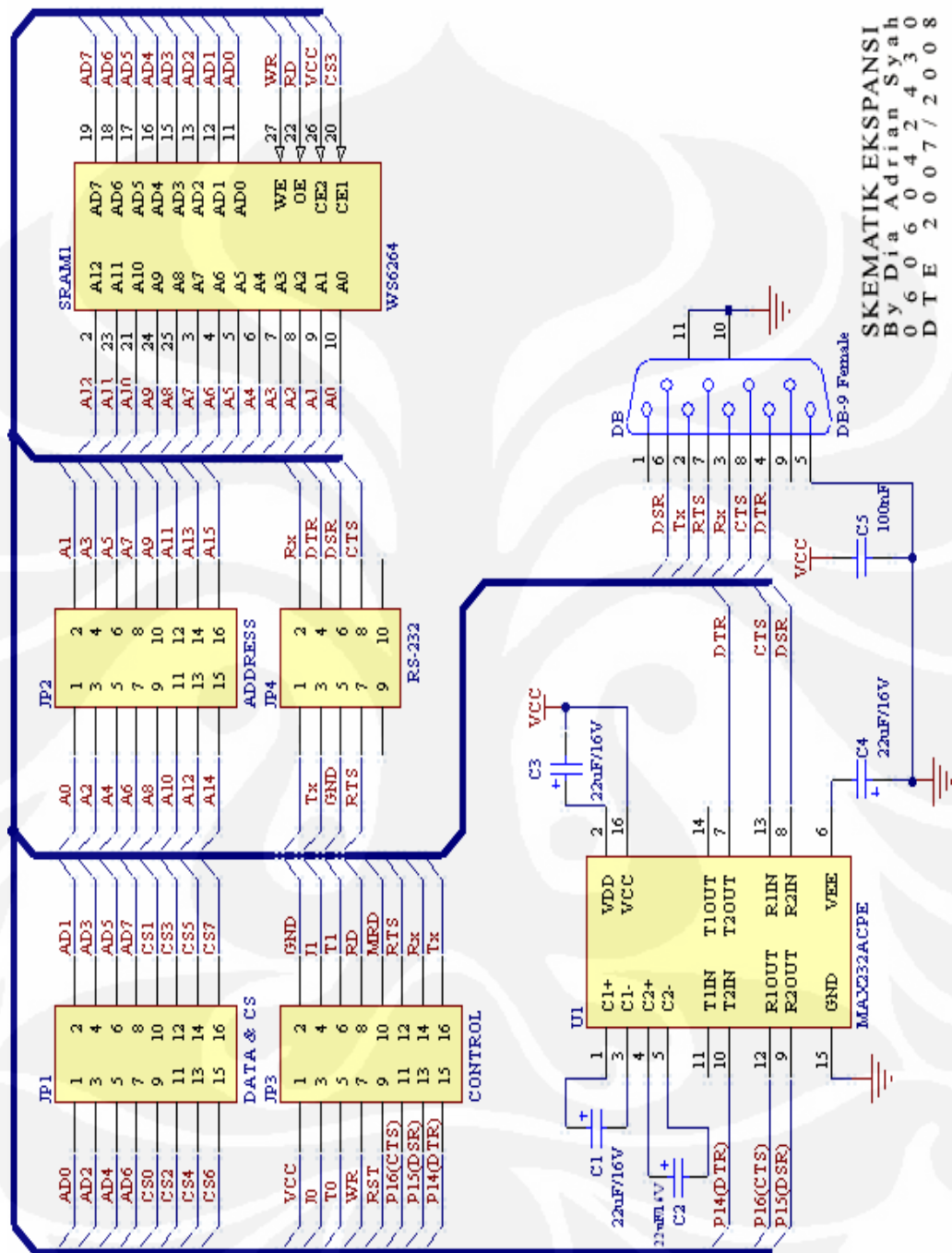
Pin 5 : *Signal Ground / Common Ground*
Jalur *signal ground* yang merupakan tegangan 0 volt yang digunakan sebagai referensi mengukur tegangan sinyal yang dikirim atau diterima. *Signal ground* umumnya dihubungkan dengan *protective ground* untuk mengurangi *noise* yang sering timbul akibat *noise signal*.

Pin 6 : *Data Set Ready (DSR)*
Untuk mengaktifkan sinyal *DCE* yang menginformasikan pada *DCE* bahwa *DCE* telah aktif dan siap untuk berhubungan / bekerja.

Pin 7 : *Request To Send (RTS)*
Berguna untuk memberitahukan *DCE* bahwa *DTE* akan mengirimkan data, *Request To Send* ini berperan sebagai *protocol* perangkat keras yang mendahului pengiriman data dari *DTE* ke *DCE*.

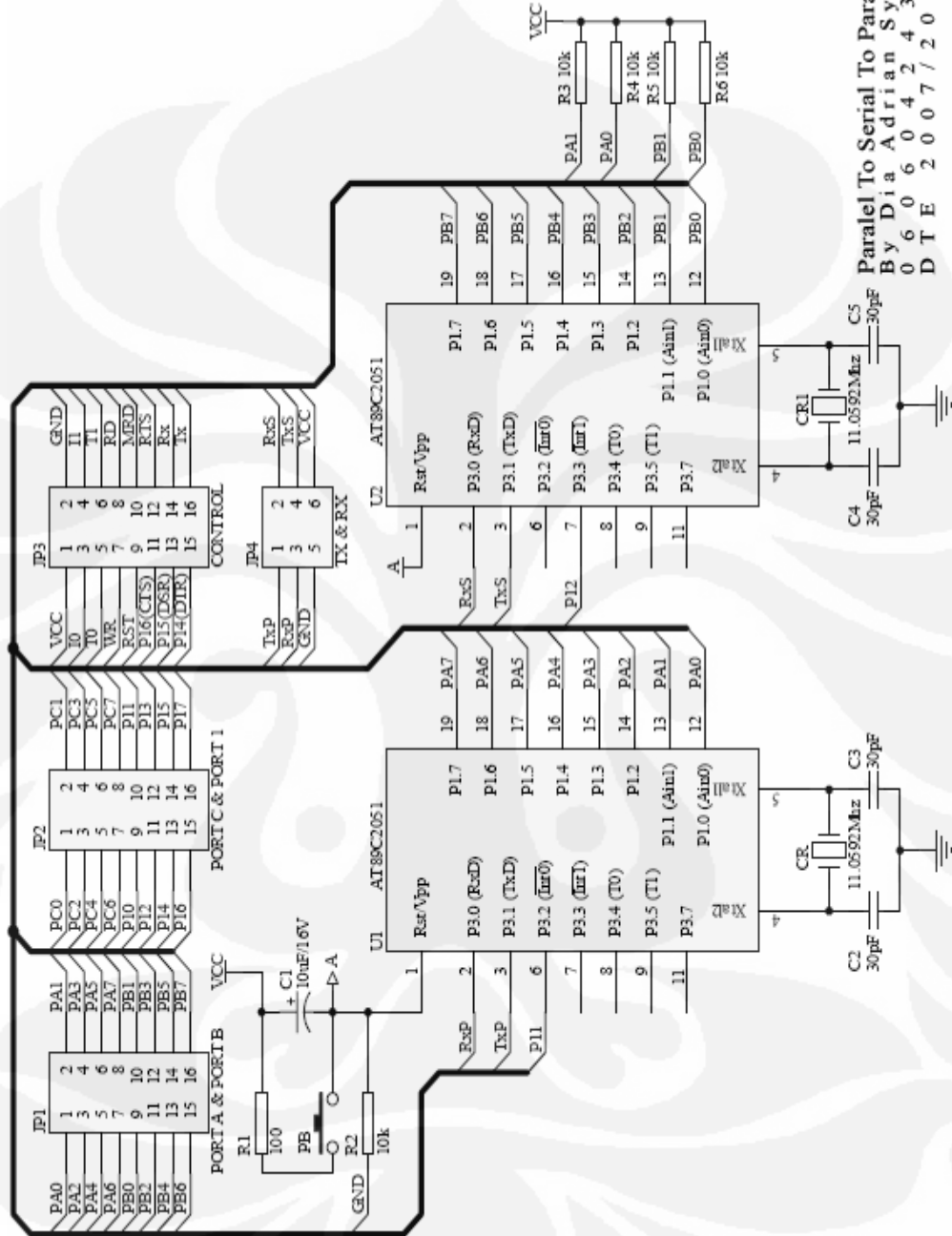
Pin 8 : *Clear To Send (CTS)*
Berfungsi untuk memberitahukan *DTE* bahwa *DCE* siap untuk menerima data, dan *CTS* disini berperan sebagai *protocol* yang mendahului pengiriman data tersebut.

2. Skematik Modul Ekspansi



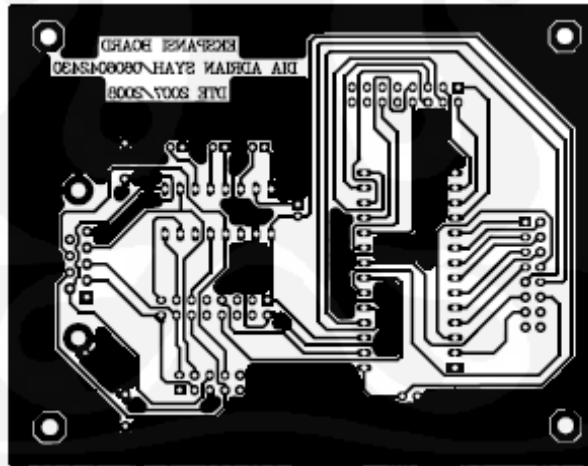
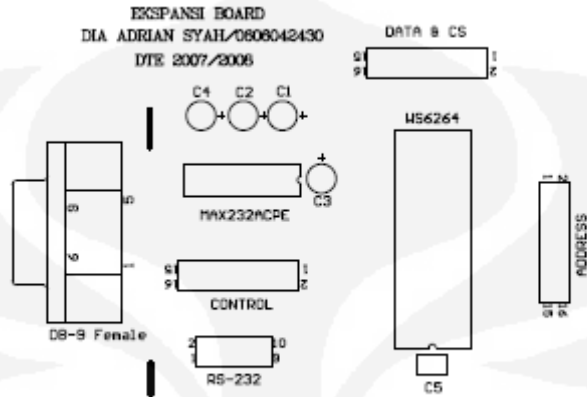
SKEMATIK EKSPANSI
 By Dia Adrian Syah
 0 6 0 6 0 4 2 4 3 0
 D T E 2 0 0 7 / 2 0 0 8

3. Skematik Paralel ke Serial ke Paralel

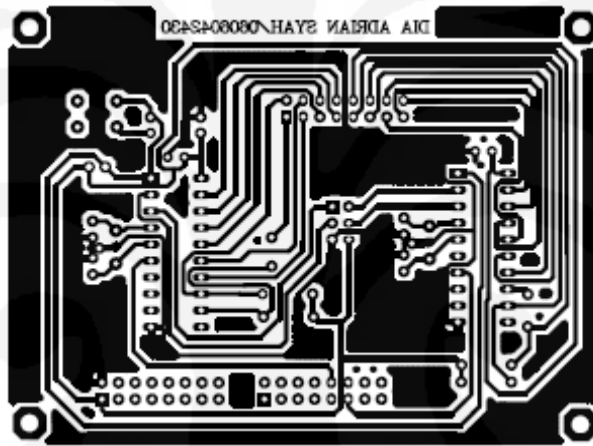
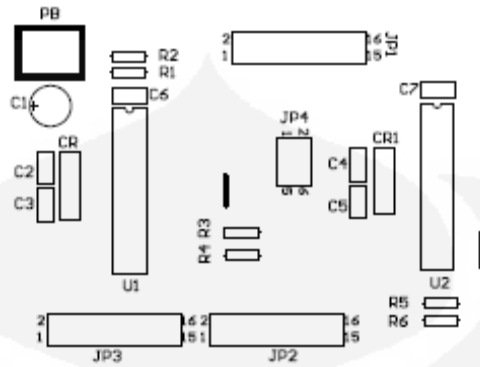


Paralel To Serial To Paralel
 By Dia Adrian Syah
 0606042430
 D T E 2007/2008

4. Layout Modul Ekspansi



5. Layout Paralel ke Serial ke Paralel



6. Gambar Perangkat



LAMPIRAN 3

1. Listing Program AT89S51

```
; File name : komserial.asm
; Author : Dia Adrian Syah
; NPM : 0606042430
; Program Studi : Teknik Elektro FTUI
; Project : Tugas Akhir 2007/2008
; Title : PENGEMBANGAN PERANGKAT LUNAK SISTEM MIKROKONTROLER AT89S51
        SEBAGAI PENGENDALI KOMUNIKASI DATA SERIAL KOMPUTER PC DENGAN
        MODEM QPSK POWER LINE COMMUNICATION
; Comment : Komunikasi serial menggunakan Handshaking (P1.4=DTR dan P1.7=RTS)
; Crystal : 11.0592 Mhz
;=====
; Definisi kode memori
PORTA EQU 2000H
PORTB EQU 2001H
PORTC EQU 2002H
CPORT EQU 2003H
CW1 EQU 0083H      ; PORT A output, Port B input, Port C lower (PC0-PC3) sbg input dan
                  ; Port C Upper (PC4-PC7) sbg output

Dseg at 0x20
        X : DS 1
        Y : DS 1

end
;=====
; Program start dan interupt serial
CSEG
        ORG 0000H
        LJMP begin
        ORG 23H
        LJMP serial_IT
;=====
;Kode string
kirim: DB 0DH," KONEKSI KOMPUTER PC DENGAN MIKROKONTROLER AT89S51 OK'
        ",0DH,0
garis: DB 0DH,"=====",0DH,0
;=====
; Ambil data dari PC menggunakan serial buffer
serial_IT:
        INC R3
        CJNE R3,#50,NEXTT
        MOV R3,#00H
        CLR P1.0      ; Kirim data memori ke PPI 82C55
        RETI
NEXTT:  JNB RI,serial_IT ; Periksa apakah ada data yang diterima
        CLR RI      ; Clear flag register
        MOV A,SBUF  ; Baca data dari UART
        MOVX @DPTR,A ; Simpan data dimemori
        INC DPTR    ; Alamat memori berikutnya
        MOV R1,DPH
        MOV R2,DPL
        MOV R3,#00
```

```

CJNE R1,#78H,serial_IT ; 6 KByte/*Hentikan pengiriman data dari PC*/
SETB P1.4 ; DTR tidak aktif (output)
SETB P1.7 ; RTS tidak aktif (Output)
CLR P1.0 ; Kirim data memori ke PPI 82C55
RETI

;=====
; Code program awal
ORG 0100h
begin:
MOV SCON,#52H ; Serial port mode 1
MOV TMOD,#20H ; Timer 1, mode 2 (8 bit auto reload)
MOV PCON,#80H ; SMOD=1
MOV TH1,#FFH ; Reload count untuk 57600 baudrate
SETB TR1 ; Aktifkan timer 1
ORL AUXR,#01H
ACALL init_8255 ; Inialisasi IC 82C55A
SETB ES ; Enable serial interrupt
SETB EA ; Enable global interrupt
ACALL Status ; Periksa status koneksi dengan PC
CLR P1.4 ; DTR aktif (output)
CLR P1.7 ; RTS aktif (Output)
ACALL String_status ; kirim string status
MOV DPTR,#6000H ; Alamat memori eksternal(SRAM 6264)
MOV R1,DPH
MOV R2,DPL
MOV SP,#30H
MOV R3,#00H
LOOP: ACALL SEND_PPI
ACALL Receive_PPI
SJMP LOOP

;=====
; Pengiriman data melalui I/O 82C55A
SEND_PPI:
prikas: MOV R7,#00
JB P1.0,exit
INC R7
ACALL delay
CJNE R7,#03,prikas
CJNE R1,#60H,periksa1
MOV R6,#01H
CJNE R2,#00,next1
SETB P1.0
ACALL Receive_PPI
LJMP exit
periksa1: MOV A,R1
SUBB A,#5FH
MOV R6,A
next1: CLR ES
MOV DPTR,#6000h
nexbuf: MOVX A,@DPTR
PUSH DPH
PUSH DPL
ACALL OutPA_8255
CLR P1.1 ;clock sisi rendah
ACALL delay_1ms
SETB P1.1 ;clock sisi tinggi
cek: MOV R7,#00
JB P1.2,$

```

```

JNB P1.2,$
ACALL InPB_8255
ACALL Txbyte
POP DPL
POP DPH
INC DPTR
DJNZ R2,nexbuf
DJNZ R6,nexbuf
MOV DPTR,#6000h
MOV R1,DPH
MOV R2,DPL
CLR P1.4 ; DTR aktif (output)
CLR P1.7 ; RTS aktif (Output)
SETB P1.0
SETB ES
exit:
RET
Receive_PPI: JB P1.2,skip
JNB P1.2,$
ACALL InPB_8255
ACALL Txbyte
MOV DPTR,#6000h
skip: RET
;=====
; Insialisasi 82C55A
init_8255:
MOV DPTR,#CPORT
MOV A,#CW1 ;PORT A output, Port B input, Port C (PC0-PC3) sbg
MOVX @DPTR,A ;input dan Port C Upper (PC4-PC7) sbg output
RET
;=====
; Keluarkan data ke PORT A IC 82C55A
OutPA_8255:
MOV DPTR,#PORTA ; Output
MOVX @DPTR,A
RET
;=====
; Ambil data pada PORT B IC 82C55A
InPB_8255:
MOV DPTR,#PORTB ; Input
MOVX A,@DPTR
RET
;=====
; Kirim data ke PC menggunakan serial buffer
Txbyte:
JNB TI,$ ; Periksa apakah ada data yang akan dikirim
CLR TI
MOV SBUF,A
RET
;=====
; Tampilkan semua string koneksi
String_status :
MOV DPTR,#garis
ACALL batas
MOV DPTR,#kirim
ACALL Kirim_Status
MOV DPTR,#garis
ACALL batas
RET

```



```

=====
; Periksa Koneksi Komputer PC
Status :
DSR:      MOV R7,#00
          JB P1.5,Status      ; Periksa DTR
          INC R7
          ACALL delay_1ms
          CJNE R7,#05,DSR
Status1:  MOV R7,#00
CTS:     JB P1.6,Status1     ; Periksa CTS
          INC R7
          ACALL delay_1ms
          CJNE R7,#05,CTS
          RET
=====
; String koneksi
Kirim_Status:
          CLR A
          MOVC A,@A+DPTR
          JZ selesai
          ACALL Txbyte
          INC DPTR
          SJMP Kirim_Status
selesai: RET
batas:   CLR A
          MOVC A,@A+DPTR
          JZ skip1
          ACALL Txbyte
          INC DPTR
          SJMP batas
skip1:   RET
=====
; Waktu tunda
delay:
          MOV X,#2
jio:     MOV Y,#100
lop:     ACALL Receive_PPI
          DJNZ Y,lop          ; 2 cycle (1 cycle=1 us)
          DJNZ X,jio         ; 2 cycle
          RET
delay_1ms:
          MOV X,#2
ji:      MOV Y,#130
          DJNZ Y,$           ; 2 cycle (1 cycle=1 us)
          DJNZ X,ji         ; 2 cycle
          RET
=====
END

```

2. Listing Program Paralel ke Serial

```

; File name : par2serial.asm
; Author : Dia Adrian Syah
; NPM : 0606042430
; Program Studi : Teknik Elektro FTUI
; Project : Tugas Akhir 2007/2008
; Title : PENGEMBANGAN PERANGKAT LUNAK SISTEM MIKROKONTROLER AT89S51

```

SEBAGAI PENGENDALI KOMUNIKASI DATA SERIAL KOMPUTER PC DENGAN
MODEM QPSK POWER LINE COMMUNICATION

; Comment : Pengubah data paralel P1 ke serial menggunakan IC AT89C2051
; crystal : 11.0592 Mhz

=====

; Definisi kode memori

Dseg at 0x20

X : DS 1

Y : DS 1

end

=====

; Code program awal

CSEG

ORG 0000H

Start:

MOV SCON,#52H

; Serial port mode 1

MOV TMOD,#20H

; Timer 1, mode 2 (8 bit auto reload)

MOV PCON,#80H

; SMOD=1

MOV TH1,#FFH

; Reload count untuk 57600 baudrate

SETB TR1

; Aktifkan timer 1

SETB ES

SETB EA

LOOP:

JB P3.2,\$

; periksa apakah ada data

MOV A,P1

; ambil data P1

ACALL Txbyte

; kirim ke port serial

JNB P3.2,\$

AJMP LOOP

=====

; Kirim data ke serial buffer

Txbyte:

JNB TI,\$

; Periksa apakah ada data yang akan dikirim

CLR TI

MOV SBUF,A

RET

=====

; Waktu tunda

delay_1ms:

MOV X,#2

jio:

MOV Y,#130

DJNZ Y,\$

; 2 cycle (1 cycle=1 us)

DJNZ X,jio

; 2 cycle

RET

=====

END

3. Listing Program Serial ke Paralel

; File name : serial2par.asm

; Author : Dia Adrian Syah

; NPM : 0606042430

; Program Studi : Teknik Elektro FTUI

; Project : Tugas Akhir 2007/2008

; Title : PENGEMBANGAN PERANGKAT LUNAK SISTEM MIKROKONTROLER AT89S51
SEBAGAI PENGENDALI KOMUNIKASI DATA SERIAL KOMPUTER PC DENGAN
MODEM QPSK POWER LINE COMMUNICATION

; Comment : Pengubah data dari serial port ke paralel P1 menggunakan IC AT89C2051

```

; crystal : 11.0592 Mhz
;=====
; Definisi kode memori
Dseg at 0x20
        X : DS 1
        Y : DS 1

end
;=====
; Code program awal
CSEG
        ORG 0000H

Start:
        MOV SCON,#52H      ; Serial port mode 1
        MOV TMOD,#20H     ; Timer 1, mode 2 (8 bit auto reload)
        MOV PCON,#80H     ; SMOD=1
        MOV TH1,#FFH     ; Reload count untuk 57600 baudrate
        SETB TR1         ; Aktifkan timer 1
        SETB ES
        SETB EA
LOOP:   ACALL Rxbyte
        MOV P1,A          ; keluarkan data ke P1
        CLR P3.3
        ACALL delay_1ms
        SETB P3.3
        AJMP LOOP

;=====
; Terima data dari serial buffer
RxByte: JNB RI,$
        CLR RI
        MOV A,SBUF
        RET

;=====
; Waktu tunda
delay_1ms:
jio:   MOV X,#2
        MOV Y,#130
        DJNZ Y,$         ; 2 cycle (1 cycle=1 us)
        DJNZ X,jio       ; 2 cycle
        RET

;=====
END

```

LAMPIRAN 4



DATA SHEET IC

+5V Powered Dual RS-232 Transmitter/Receiver

December 1993

Features

- Meets All RS-232C Specifications
- Requires Only Single +5V Power Supply
- Onboard Voltage Doubler/Inverter
- Low Power Consumption
- 2 Drivers
 - $\pm 9V$ Output Swing for +5V Input
 - 300Ω Power-off Source Impedance
 - Output Current Limiting
 - TTL/CMOS Compatible
 - $30V/\mu s$ Maximum Slew Rate
- 2 Receivers
 - $\pm 30V$ Input Voltage Range
 - $3k\Omega$ to $7k\Omega$ Input Impedance
 - 0.5V Hysteresis to Improve Noise Rejection
- All Critical Parameters are Guaranteed Over the Entire Commercial, Industrial and Military Temperature Ranges

Applications

- Any System Requiring RS-232 Communications Port
 - Computer - Portable and Mainframe
 - Peripheral - Printers and Terminals
 - Portable Instrumentation
 - Modems
 - Dataloggers

Description

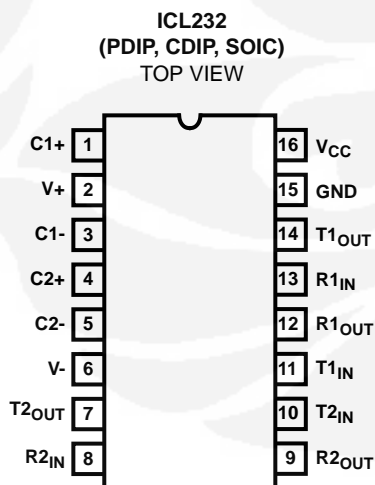
The ICL232 is a dual RS-232 transmitter/receiver interface circuit that meets all EIA RS-232C specifications. It requires a single +5V power supply, and features two onboard charge pump voltage converters which generate +10V and -10V supplies from the 5V supply.

The drivers feature true TTL/CMOS input compatibility, slew-rate-limited output, and 300Ω power-off source impedance. The receivers can handle up to +30V, and have a $3k\Omega$ to $7k\Omega$ input impedance. The receivers also have hysteresis to improve noise rejection.

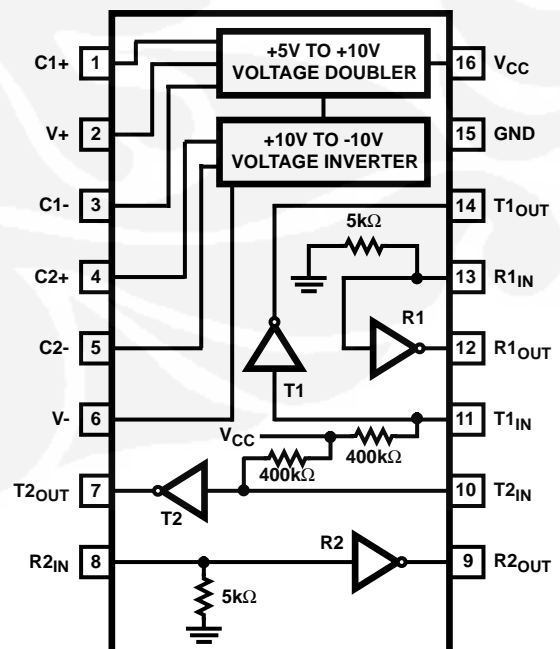
Ordering Information

PART NUMBER	TEMPERATURE RANGE	PACKAGE
ICL232CPE	0°C to +70°C	16 Lead Plastic DIP
ICL232CJE	0°C to +70°C	16 Lead Ceramic DIP
ICL232CBE	0°C to +70°C	16 Lead SOIC (W)
ICL232IPE	-40°C to +85°C	16 Lead Plastic DIP
ICL232IJE	-40°C to +85°C	16 Lead Ceramic DIP
ICL232IBE	-40°C to +85°C	16 Lead SOIC (W)
ICL232MJE	-55°C to +125°C	16 Lead Ceramic DIP

Pinouts



Functional Diagram



Specifications ICL232

Absolute Maximum Ratings

V_{CC} to Ground	(GND -0.3V) < V_{CC} < 6V
V+ to Ground	(V_{CC} -0.3V) < V+ < 12V
V- to Ground	-12V < V- < (GND +0.3V)
Input Voltages	
T1 _{IN} , T2 _{IN}	(V- -0.3V) < V _{IN} < (V+ +0.3V)
R1 _{IN} , R2 _{IN}	±30V
Output Voltages	
T1 _{OUT} , T2 _{OUT}	(V- -0.3V) < V _{TXOUT} < (V+ +0.3V)
R1 _{OUT} , R2 _{OUT}	(GND -0.3V) < V _{RXOUT} < (V _{CC} +0.3V)
Short Circuit Duration	
T1 _{OUT} , T2 _{OUT}	Continuous
R1 _{OUT} , R2 _{OUT}	Continuous
Storage Temperature Range	-65°C to +150°C
Lead Temperature (Soldering 10s)	+300°C

Thermal Information

Thermal Resistance	θ_{JA}	θ_{JC}
Ceramic DIP Package	80°C/W	24°C/W
Plastic DIP Package	100°C/W	-
SOIC Package	100°C/W	-
Maximum Power Dissipation	250mW	
Operating Temperature Range		
ICL232C	0°C to +70°C	
ICL232I	-40°C to +85°C	
ICL232M	-55°C to +125°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.

Electrical Specifications

Test Conditions: $V_{CC} = +5V \pm 10\%$, $T_A =$ Operating Temperature Range. Test Circuit as in Figure 8 Unless Otherwise Specified

PARAMETER	TEST CONDITIONS	LIMITS			UNITS
		MIN	TYP	MAX	
Transmitter Output Voltage Swing, T _{OUT}	T1 _{OUT} and T2 _{OUT} loaded with 3kΩ to Ground	±5	±9	±10	V
Power Supply Current, I _{CC}	Outputs Unloaded, T _A = +25°C	-	5	10	mA
T _{IN} , Input Logic Low, V _{IL}		-	-	0.8	V
T _{IN} , Input Logic High, V _{IH}		2.0	-	-	V
Logic Pullup Current, I _P	T1 _{IN} , T2 _{IN} = 0V	-	15	200	μA
RS-232 Input Voltage Range, V _{IN}		-30	-	+30	V
Receiver Input Impedance, R _{IN}	V _{IN} = ±3V	3.0	5.0	7.0	kΩ
Receiver Input Low Threshold, V _{IN} (H-L)	V _{CC} = 5.0V, T _A = +25°C	0.8	1.2	-	V
Receiver Input High Threshold, V _{IN} (L-H)	V _{CC} = 5.0V, T _A = +25°C	-	1.7	2.4	V
Receiver Input Hysteresis, V _{HYST}		0.2	0.5	1.0	V
TTL/CMOS Receiver Output Voltage Low, V _{OL}	I _{OUT} = 3.2mA	-	0.1	0.4	V
TTL/CMOS Receiver Output Voltage High, V _{OH}	I _{OUT} = -1.0mA	3.5	4.6	-	V
Propagation Delay, t _{PD}	RS-232 to TTL	-	0.5	-	μs
Instantaneous Slew Rate, SR	C _L = 10pF, R _L = 3kΩ, T _A = +25°C (Notes 1, 2)	-	-	30	V/μs
Transition Region Slew Rate, SR _T	R _L = 3kΩ, C _L = 2500pF Measured from +3V to -3V or -3V to +3V	-	3	-	V/μs
Output Resistance, R _{OUT}	V _{CC} = V+ = V- = 0V, V _{OUT} = ±2V	300	-	-	Ω
RS-232 Output Short Circuit Current, I _{SC}	T1 _{OUT} or T2 _{OUT} shorted to GND	-	±10	-	mA

NOTES:

1. Guaranteed by design.
2. See Figure 4 for definition.

Typical Performance Curves

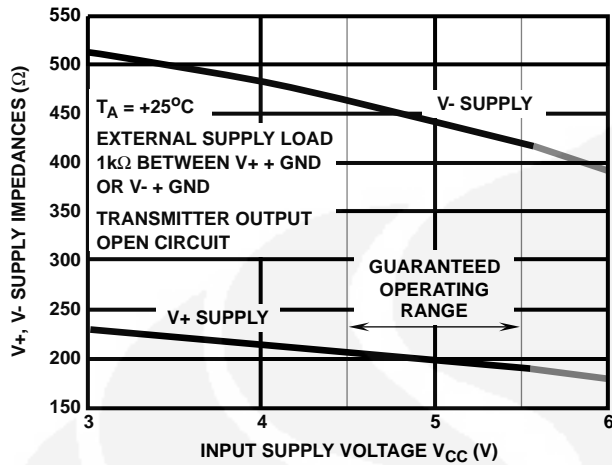


FIGURE 1. V+, V- OUTPUT IMPEDANCES vs V_{CC}

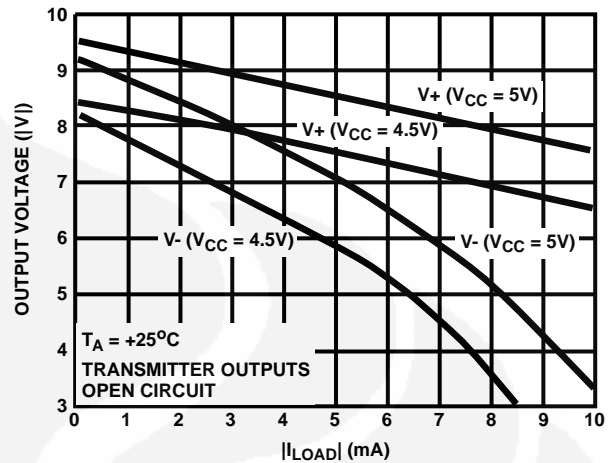


FIGURE 2. V+, V- OUTPUT VOLTAGES vs LOAD CURRENT

Pin Descriptions

PLASTIC DIP, CERAMIC DIP	SOIC	PIN NAME	DESCRIPTION
1	1	C1+	External capacitor "+" for internal voltage doubler.
2	2	V+	Internally generated +10V (typical) supply.
3	3	C1-	External capacitor "-" for internal voltage doubler.
4	4	C2+	External capacitor "+" internal voltage inverter.
5	5	C2-	External capacitor "-" internal voltage inverter.
6	6	V-	Internally generated -10V (typical) supply.
7	7	T2 _{OUT}	RS-232 Transmitter 2 output ±10V (typical).
8	8	R2 _{IN}	RS-232 Receiver 2 input, with internal 5K pulldown resistor to GND.
9	9	R2 _{out}	Receiver 2 TTL/CMOS output.
10	10	T2 _{IN}	Transmitter 2 TTL/CMOS input, with internal 400K pullup resistor to V _{CC} .
11	11	T1 _{IN}	Transmitter 1 TTL/CMOS input, with internal 400K pullup resistor to V _{CC} .
12	12	R1 _{OUT}	Receiver 1 TTL/CMOS output.
13	13	R1 _{IN}	RS-232 Receiver 1 input, with internal 5K pulldown resistor to GND.
14	14	T1 _{OUT}	RS-232 Transmitter 1 output ±10V (typical).
15	15	GND	Supply Ground.
16	16	VCC	Positive Power Supply +5V ±10%

Detailed Description

The ICL232 is a dual RS-232 transmitter/receiver powered by a single +5V power supply which meets all EIA RS232C specifications and features low power consumption. The functional diagram illustrates the major elements of the ICL232. The circuit is divided into three sections: a voltage doubler/inverter, dual transmitters, and dual receivers.

Voltage Converter

An equivalent circuit of the dual charge pump is illustrated in Figure 3.

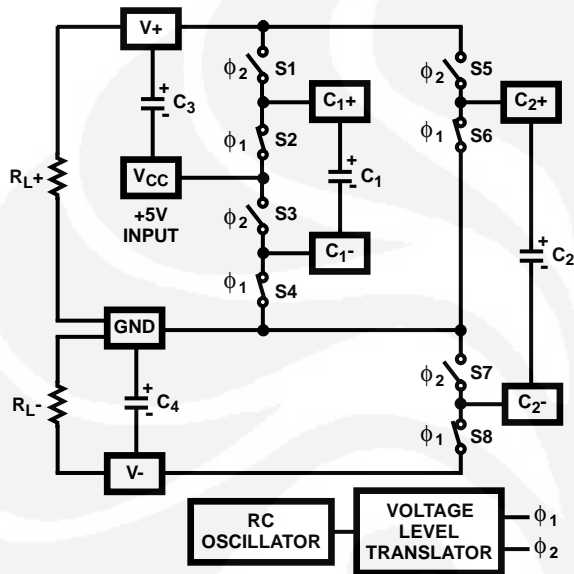


FIGURE 3. DUAL CHARGE PUMP

The voltage quadrupler contains two charge pumps which use two phases of an internally generated clock to generate +10V and -10V. The nominal clock frequency is 16kHz. During phase one of the clock, capacitor C1 is charged to V_{CC} . During phase two, the voltage on C1 is added to V_{CC} , producing a signal across C2 equal to twice V_{CC} . At the same time, C3 is also charged to $2V_{CC}$, and then during phase one, it is inverted with respect to ground to produce a signal across C4 equal to $-2V_{CC}$. The voltage converter accepts input voltages up to 5.5V. The output impedance of the doubler (V+) is approximately 200Ω , and the output impedance of the inverter (V-) is approximately 450Ω . Typical graphs are presented which show the voltage converters output vs input voltage and output voltages vs load characteristics. The test circuit (Figure 8) uses $1\mu F$ capacitors for C1-C4, however, the value is not critical. Increasing the values of C1 and C2 will lower the output impedance of the voltage doubler and inverter, and increasing the values of the reservoir capacitors, C3 and C4, lowers the ripple on the V+ and V- supplies.

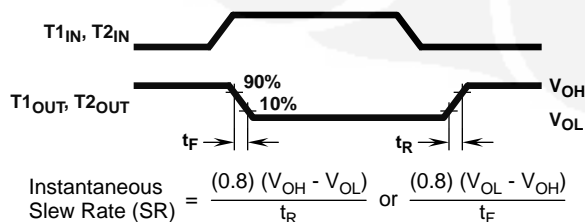


FIGURE 4. SLEW RATE DEFINITION

Transmitters

The transmitters are TTL/CMOS compatible inverters which translate the inputs to RS-232 outputs. The input logic threshold is about 26% of V_{CC} , or 1.3V for $V_{CC} = 5V$. A logic 1 at the input results in a voltage of between -5V and V- at the output, and a logic 0 results in a voltage between +5V and $(V+ - 0.6V)$. Each transmitter input has an internal $400k\Omega$ pullup resistor so any unused input can be left unconnected and its output remains in its low state. The output voltage swing meets the RS-232C specification of $\pm 5V$ minimum with the worst case conditions of: both transmitters driving $3k\Omega$ minimum load impedance, $V_{CC} = 4.5V$, and maximum allowable operating temperature. The transmitters have an internally limited output slew rate which is less than $30V/\mu s$. The outputs are short circuit protected and can be shorted to ground indefinitely. The powered down output impedance is a minimum of 300Ω with $\pm 2V$ applied to the outputs and $V_{CC} = 0V$.

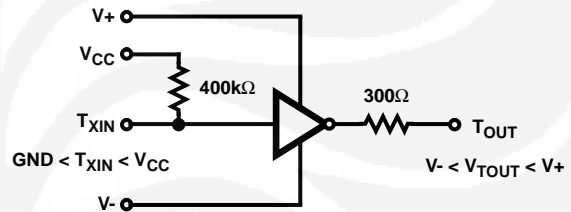


FIGURE 5. TRANSMITTER

Receivers

The receiver inputs accept up to $\pm 30V$ while presenting the required $3k\Omega$ to $7k\Omega$ input impedance even if the power is off ($V_{CC} = 0V$). The receivers have a typical input threshold of 1.3V which is within the $\pm 3V$ limits, known as the transition region, of the RS-232 specification. The receiver output is 0V to V_{CC} . The output will be low whenever the input is greater than 2.4V and high whenever the input is floating or driven between +0.8V and -30V. The receivers feature 0.5V hysteresis to improve noise rejection.

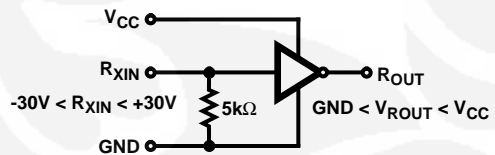


FIGURE 6. RECEIVER

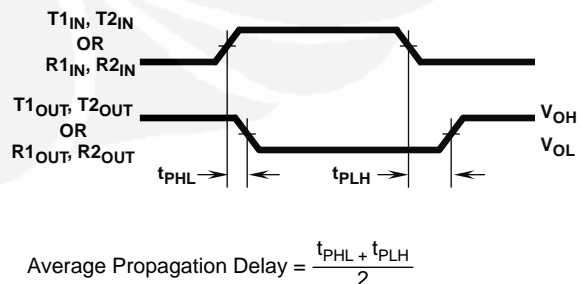


FIGURE 7. PROPAGATION DELAY DEFINITION

Test Circuits

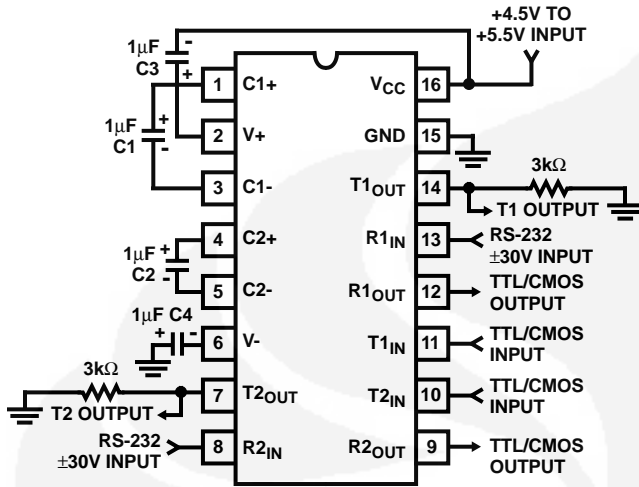


FIGURE 8. GENERAL TEST CIRCUIT

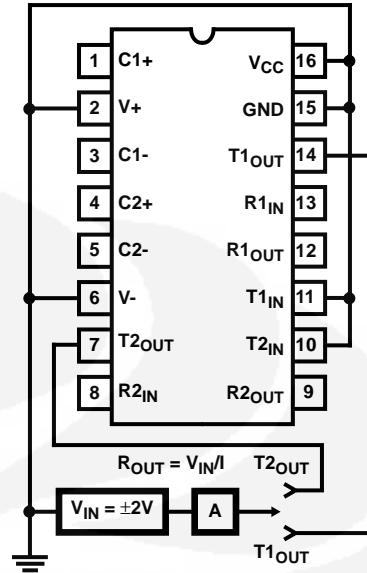


FIGURE 9. POWER-OFF SOURCE RESISTANCE CONFIGURATION

Applications

The ICL232 may be used for all RS-232 data terminal and communication links. It is particularly useful in applications where ±12V power supplies are not available for conventional RS-232 interface circuits. The applications presented represent typical interface configurations.

A simple duplex RS-232 port with CTS/RTS handshaking is illustrated in Figure 10. Fixed output signals such as DTR (data terminal ready) and DSRS (data signaling rate select) is generated by driving them through a 5kΩ resistor connected to V+.

capacitors (C3 and C4). The benefit of sharing common reservoir capacitors is the elimination of two capacitors and the reduction of the charge pump source impedance which effectively increases the output swing of the transmitters.

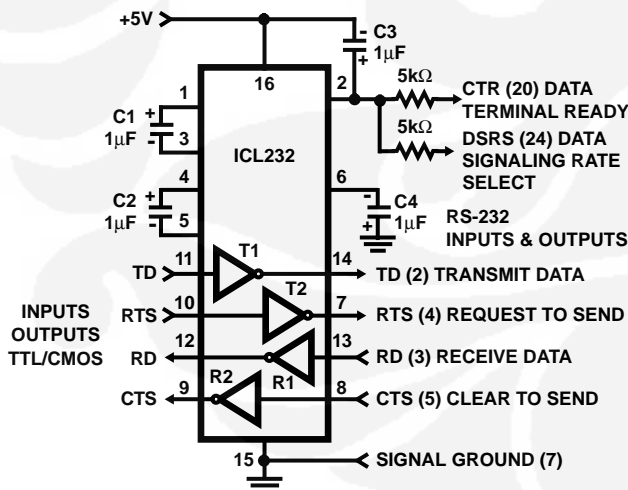


FIGURE 10. SIMPLE DUPLEX RS-232 PORT WITH CTS/RTS HANDSHAKING

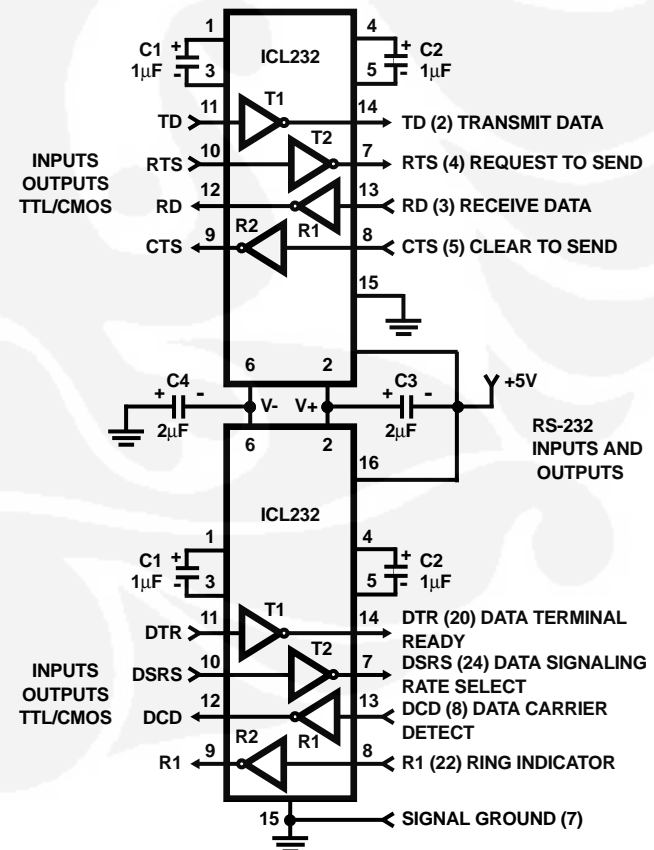


FIGURE 11. COMBINING TWO ICL232s FOR 4 PAIRS OF RS-232 INPUTS AND OUTPUTS

In applications requiring four RS-232 inputs and outputs (Figure 11), note that each circuit requires two charge pump capacitors (C1 and C2) but can share common reservoir

Features

- Fast Read Access Time - 150 ns
- Automatic Page Write Operation
 - Internal Address and Data Latches for 64 Bytes
- Fast Write Cycle Times
 - Page Write Cycle Time: 10 ms Maximum
 - 1 to 64-byte Page Write Operation
- Low Power Dissipation
 - 40 mA Active Current
 - 100 μ A CMOS Standby Current
- Hardware and Software Data Protection
- DATA Polling and Toggle Bit for End of Write Detection
- High Reliability CMOS Technology
 - Endurance: 100,000 Cycles
 - Data Retention: 10 Years
- Single 5V \pm 10% Supply
- CMOS and TTL Compatible Inputs and Outputs
- JEDEC Approved Byte-wide Pinout
- Commercial and Industrial Temperature Ranges

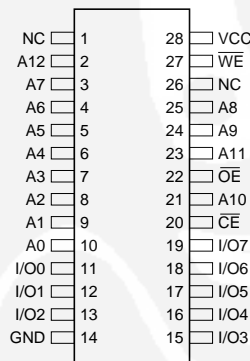
Description

The AT28C64B is a high-performance electrically-erasable and programmable read only memory (EEPROM). Its 64K of memory is organized as 8,192 words by 8 bits. Manufactured with Atmel's advanced nonvolatile CMOS technology, the device offers access times to 150 ns with power dissipation of just 220 mW. When the device is deselected, the CMOS standby current is less than 100 μ A. *(continued)*

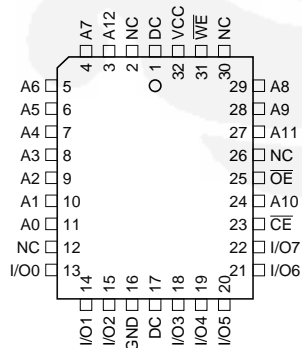
Pin Configurations

Pin Name	Function
A0 - A12	Addresses
$\overline{\text{CE}}$	Chip Enable
$\overline{\text{OE}}$	Output Enable
$\overline{\text{WE}}$	Write Enable
I/O0 - I/O7	Data Inputs/Outputs
NC	No Connect
DC	Don't Connect

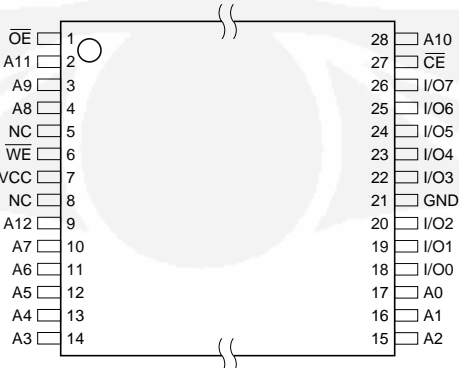
PDIP, SOIC
Top View



PLCC
Top View



TSOP
Top View



Note: PLCC package pins 1 and 17 are DON'T CONNECT.



**64K (8K x 8)
Parallel
EEPROM with
Page Write and
Software Data
Protection**

AT28C64B

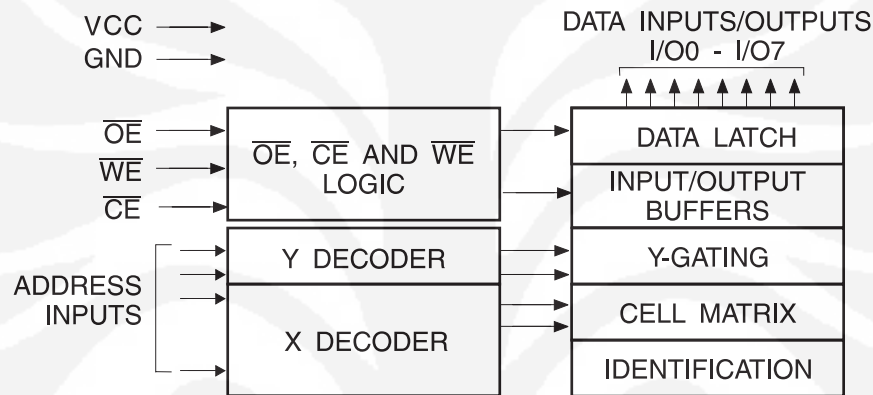
Rev. 0270G-06/99



The AT28C64B is accessed like a Static RAM for the read or write cycle without the need for external components. The device contains a 64-byte page register to allow writing of up to 64 bytes simultaneously. During a write cycle, the addresses and 1 to 64 bytes of data are internally latched, freeing the address and data bus for other operations. Following the initiation of a write cycle, the device will automatically write the latched data using an internal control timer. The end of a write cycle can be detected by DATA POLLING of I/O₇. Once the end of a write cycle has been detected, a new access for a read or write can begin.

Atmel's AT28C64B has additional features to ensure high quality and manufacturability. The device utilizes internal error correction for extended endurance and improved data retention characteristics. An optional software data protection mechanism is available to guard against inadvertent writes. The device also includes an extra 64 bytes of EEPROM for device identification or tracking.

Block Diagram



Absolute Maximum Ratings*

Temperature Under Bias	-55°C to +125°C
Storage Temperature	-65°C to +150°C
All Input Voltages (including NC Pins) with Respect to Ground	-0.6V to +6.25V
All Output Voltages with Respect to Ground	-0.6V to $V_{CC} + 0.6V$
Voltage on \overline{OE} and A9 with Respect to Ground	-0.6V to +13.5V

***NOTICE:** Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability

Device Operation

READ: The AT28C64B is accessed like a Static RAM. When \overline{CE} and \overline{OE} are low and \overline{WE} is high, the data stored at the memory location determined by the address pins is asserted on the outputs. The outputs are put in the high-impedance state when either \overline{CE} or \overline{OE} is high. This dual line control gives designers flexibility in preventing bus contention in their systems.

BYTE WRITE: A low pulse on the \overline{WE} or \overline{CE} input with \overline{CE} or \overline{WE} low (respectively) and \overline{OE} high initiates a write cycle. The address is latched on the falling edge of \overline{CE} or \overline{WE} , whichever occurs last. The data is latched by the first rising edge of \overline{CE} or \overline{WE} . Once a byte write has been started, it will automatically time itself to completion. Once a programming operation has been initiated and for the duration of t_{WC} , a read operation will effectively be a polling operation.

PAGE WRITE: The page write operation of the AT28C64B allows 1 to 64 bytes of data to be written into the device during a single internal programming period. A page write operation is initiated in the same manner as a byte write; after the first byte is written, it can then be followed by 1 to 63 additional bytes. Each successive byte must be loaded within $150\ \mu\text{s}$ (t_{BLC}) of the previous byte. If the t_{BLC} limit is exceeded, the AT28C64B will cease accepting data and commence the internal programming operation. All bytes during a page write operation must reside on the same page as defined by the state of the A6 to A12 inputs. For each \overline{WE} high to low transition during the page write operation, A6 to A12 must be the same.

The A0 to A5 inputs specify which bytes within the page are to be written. The bytes may be loaded in any order and may be altered within the same load period. Only bytes which are specified for writing will be written; unnecessary cycling of other bytes within the page does not occur.

DATA POLLING: The AT28C64B features \overline{DATA} Polling to indicate the end of a write cycle. During a byte or page write cycle an attempted read of the last byte written will result in the complement of the written data to be presented on I/O₇. Once the write cycle has been completed, true data is valid on all outputs, and the next write cycle may begin. \overline{DATA} Polling may begin at any time during the write cycle.

TOGGLE BIT: In addition to \overline{DATA} Polling, the AT28C64B provides another method for determining the end of a write cycle. During the write operation, successive attempts to read data from the device will result in I/O₆ toggling between one and zero. Once the write has completed, I/O₆ will stop toggling, and valid data will be read. Toggle bit reading may begin at any time during the write cycle.

DATA PROTECTION: If precautions are not taken, inadvertent writes may occur during transitions of the host system power supply. Atmel has incorporated both hardware and software features that will protect the memory against inadvertent writes.

HARDWARE DATA PROTECTION: Hardware features protect against inadvertent writes to the AT28C64B in the following ways: (a) V_{CC} sense—if V_{CC} is below 3.8V (typical), the write function is inhibited; (b) V_{CC} power-on delay—once V_{CC} has reached 3.8V, the device will automatically time out 5 ms (typical) before allowing a write; (c) write inhibit—holding any one of \overline{OE} low, \overline{CE} high, or \overline{WE} high inhibits write cycles; and (d) noise filter—pulses of less than 15 ns (typical) on the \overline{WE} or \overline{CE} inputs will not initiate a write cycle.

SOFTWARE DATA PROTECTION: A software controlled data protection feature has been implemented on the AT28C64B. When enabled, the software data protection (SDP), will prevent inadvertent writes. The SDP feature may be enabled or disabled by the user; the AT28C64B is shipped from Atmel with SDP disabled.

SDP is enabled by the user issuing a series of three write commands in which three specific bytes of data are written to three specific addresses (refer to the *Software Data Protection Algorithm* diagram in this data sheet). After writing the 3-byte command sequence and waiting t_{WC} , the entire AT28C64B will be protected against inadvertent writes. It should be noted that even after SDP is enabled, the user may still perform a byte or page write to the AT28C64B by preceding the data to be written by the same 3-byte command sequence used to enable SDP.

Once set, SDP remains active unless the disable command sequence is issued. Power transitions do not disable SDP, and SDP protects the AT28C64B during power-up and power-down conditions. All command sequences must conform to the page write timing specifications. The data in the enable and disable command sequences is not actually written into the device; their addresses may still be written with user data in either a byte or page write operation.

After setting SDP, any attempt to write to the device without the 3-byte command sequence will start the internal write timers. No data will be written to the device. However, for the duration of t_{WC} , read operations will effectively be polling operations.

DEVICE IDENTIFICATION: An extra 64 bytes of EEPROM memory are available to the user for device identification. By raising A9 to $12V \pm 0.5V$ and using address locations 1FC0H to 1FFFH, the additional bytes may be written to or read from in the same manner as the regular memory array.

DC and AC Operating Range

		AT28C64B-15	AT28C64B-20	AT28C64B-25
Operating Temperature (Case)	Com.	0°C - 70°C	0°C - 70°C	0°C - 70°C
	Ind.	-40°C - 85°C	-40°C - 85°C	-40°C - 85°C
V _{CC} Power Supply		5V ± 10%	5V ± 10%	5V ± 10%

Operating Modes

Mode	$\overline{\text{CE}}$	$\overline{\text{OE}}$	$\overline{\text{WE}}$	I/O
Read	V _{IL}	V _{IL}	V _{IH}	D _{OUT}
Write ⁽²⁾	V _{IL}	V _{IH}	V _{IL}	D _{IN}
Standby/Write Inhibit	V _{IH}	X ⁽¹⁾	X	High Z
Write Inhibit	X	X	V _{IH}	
Write Inhibit	X	V _{IL}	X	
Output Disable	X	V _{IH}	X	High Z
Chip Erase	V _{IL}	V _H ⁽³⁾	V _{IL}	High Z

- Notes: 1. X can be V_{IL} or V_{IH}.
 2. Refer to the *AC Write Waveforms* diagrams in this data sheet.
 3. V_H = 12.0V ± 0.5V.

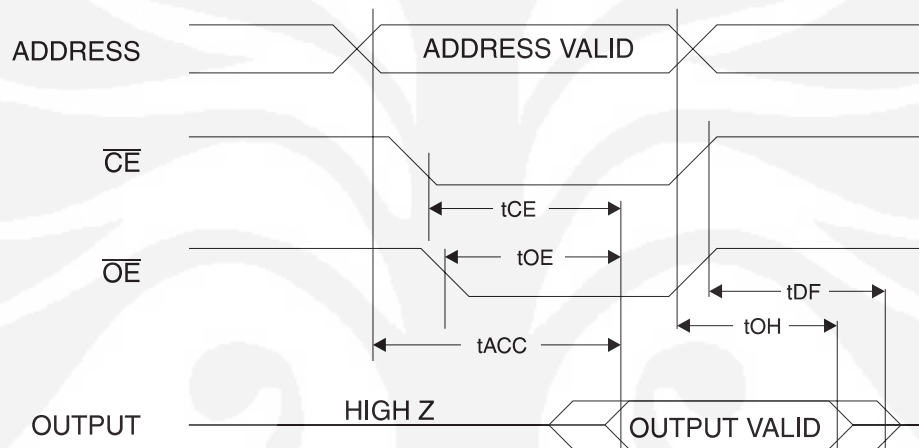
DC Characteristics

Symbol	Parameter	Condition	Min	Max	Units
I _{LI}	Input Load Current	V _{IN} = 0V to V _{CC} + 1V		10	μA
I _{LO}	Output Leakage Current	V _{I/O} = 0V to V _{CC}		10	μA
I _{SB1}	V _{CC} Standby Current CMOS	$\overline{\text{CE}} = V_{\text{CC}} - 0.3\text{V}$ to V _{CC} + 1V Com., Ind.		100	μA
I _{SB2}	V _{CC} Standby Current TTL	$\overline{\text{CE}} = 2.0\text{V}$ to V _{CC} + 1V		2	mA
I _{CC}	V _{CC} Active Current	f = 5 MHz; I _{OUT} = 0 mA		40	mA
V _{IL}	Input Low Voltage			0.8	V
V _{IH}	Input High Voltage		2.0		V
V _{OL}	Output Low Voltage	I _{OL} = 2.1 mA		0.40	V
V _{OH}	Output High Voltage	I _{OH} = -400 μA	2.4		V

AC Read Characteristics

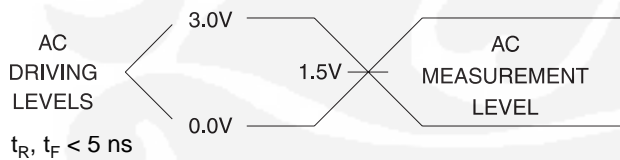
Symbol	Parameter	AT28C64B-15		AT28C64B-20		AT28C64B-25		Units
		Min	Max	Min	Max	Min	Max	
t_{ACC}	Address to Output Delay		150		200		250	ns
$t_{CE}^{(1)}$	\overline{CE} to Output Delay		150		200		250	ns
$t_{OE}^{(2)}$	\overline{OE} to Output Delay	0	70	0	80	0	100	ns
$t_{DF}^{(3)(4)}$	\overline{CE} or \overline{OE} to Output Float	0	50	0	55	0	60	ns
t_{OH}	Output Hold from \overline{OE} , \overline{CE} or Address, whichever occurred first	0		0		0		ns

AC Read Waveforms⁽¹⁾⁽²⁾⁽³⁾⁽⁴⁾

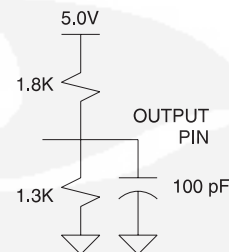


- Notes:
- \overline{CE} may be delayed up to $t_{ACC} - t_{CE}$ after the address transition without impact on t_{ACC} .
 - \overline{OE} may be delayed up to $t_{CE} - t_{OE}$ after the falling edge of \overline{CE} without impact on t_{CE} or by $t_{ACC} - t_{OE}$ after an address change without impact on t_{ACC} .
 - t_{DF} is specified from \overline{OE} or \overline{CE} whichever occurs first ($C_L = 5 \text{ pF}$).
 - This parameter is characterized and is not 100% tested.

Input Test Waveforms and Measurement Level



Output Test Load



Pin Capacitance

$f = 1 \text{ MHz}$, $T = 25^\circ\text{C}^{(1)}$

Symbol	Typ	Max	Units	Conditions
C_{IN}	4	6	pF	$V_{IN} = 0V$
C_{OUT}	8	12	pF	$V_{OUT} = 0V$

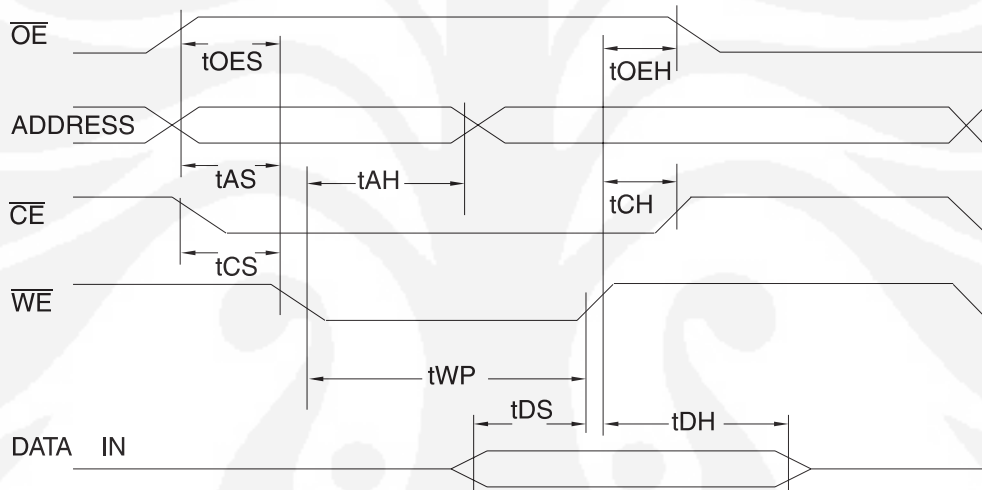
- Note: 1. This parameter is characterized and is not 100% tested.

AC Write Characteristics

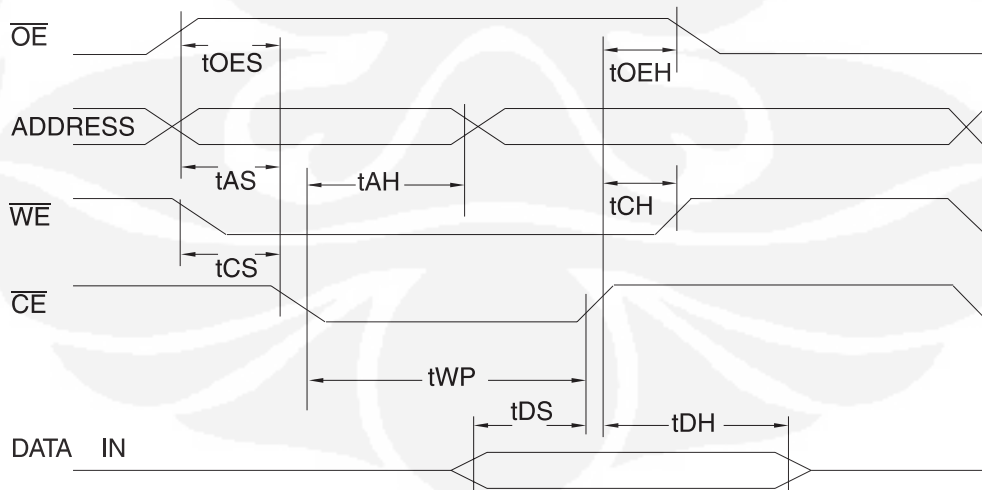
Symbol	Parameter	Min	Max	Units
t_{AS}, t_{OES}	Address, \overline{OE} Set-up Time	0		ns
t_{AH}	Address Hold Time	50		ns
t_{CS}	Chip Select Set-up Time	0		ns
t_{CH}	Chip Select Hold Time	0		ns
t_{WP}	Write Pulse Width (\overline{WE} or \overline{CE})	100		ns
t_{DS}	Data Set-up Time	50		ns
t_{DH}, t_{OEH}	Data, \overline{OE} Hold Time	0		ns

AC Write Waveforms

\overline{WE} Controlled



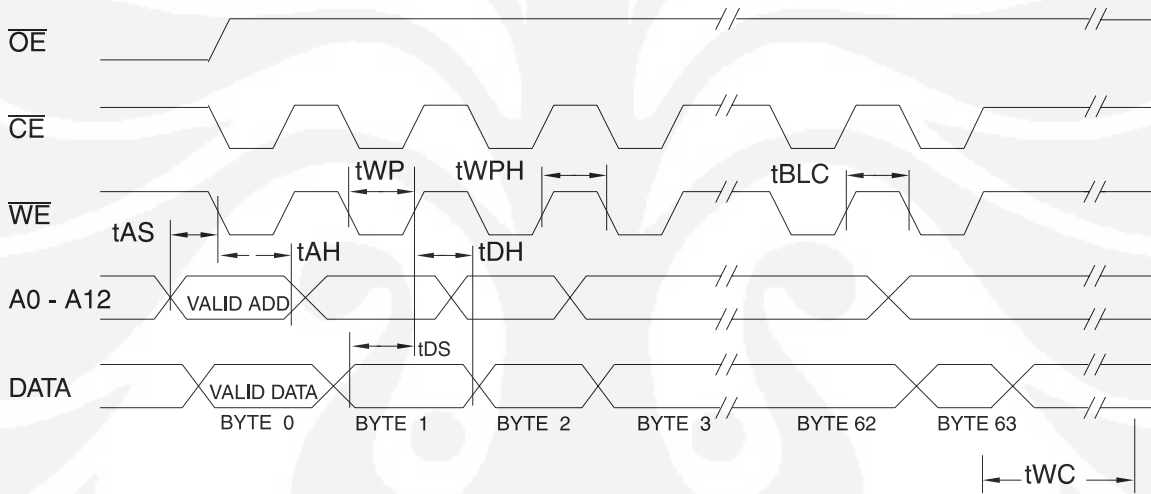
\overline{CE} Controlled



Page Mode Characteristics

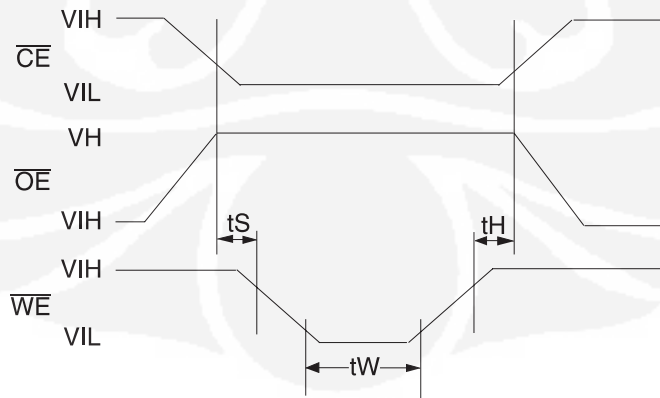
Symbol	Parameter	Min	Max	Units
t_{WC}	Write Cycle Time		10	ms
t_{WC}	Write Cycle Time (28C64B SL184)	0	2	ms
t_{AS}	Address Set-up Time	0		ns
t_{AH}	Address Hold Time	50		ns
t_{DS}	Data Set-up Time	50		ns
t_{DH}	Data Hold Time	0		ns
t_{WP}	Write Pulse Width	100		ns
t_{BLC}	Byte Load Cycle Time		150	μ s
t_{WPH}	Write Pulse Width High	50		ns

Page Mode Write Waveforms⁽¹⁾⁽²⁾



- Notes: 1. A6 through A12 must specify the same page address during each high to low transition of \overline{WE} (or \overline{CE}).
 2. \overline{OE} must be high only when \overline{WE} and \overline{CE} are both low.

Chip Erase Waveforms



$t_S = t_H = 1 \mu$ sec (min.)
 $t_W = 10$ msec (min.)
 $V_H = 12.0 \pm 0.5V$



CMOS Programmable Peripheral Interface

August 1996

Features

- Pin Compatible with NMOS 8255A
- 24 Programmable I/O Pins
- Fully TTL Compatible
- High Speed, No "Wait State" Operation with 5MHz and 8MHz 80C86 and 80C88
- Direct Bit Set/Reset Capability
- Enhanced Control Word Read Capability
- Scaled SAJI IV CMOS Process
- 2.5mA Drive Capability on All I/O Ports
- Low Standby Power (ICCSB)10µA

Ordering Information

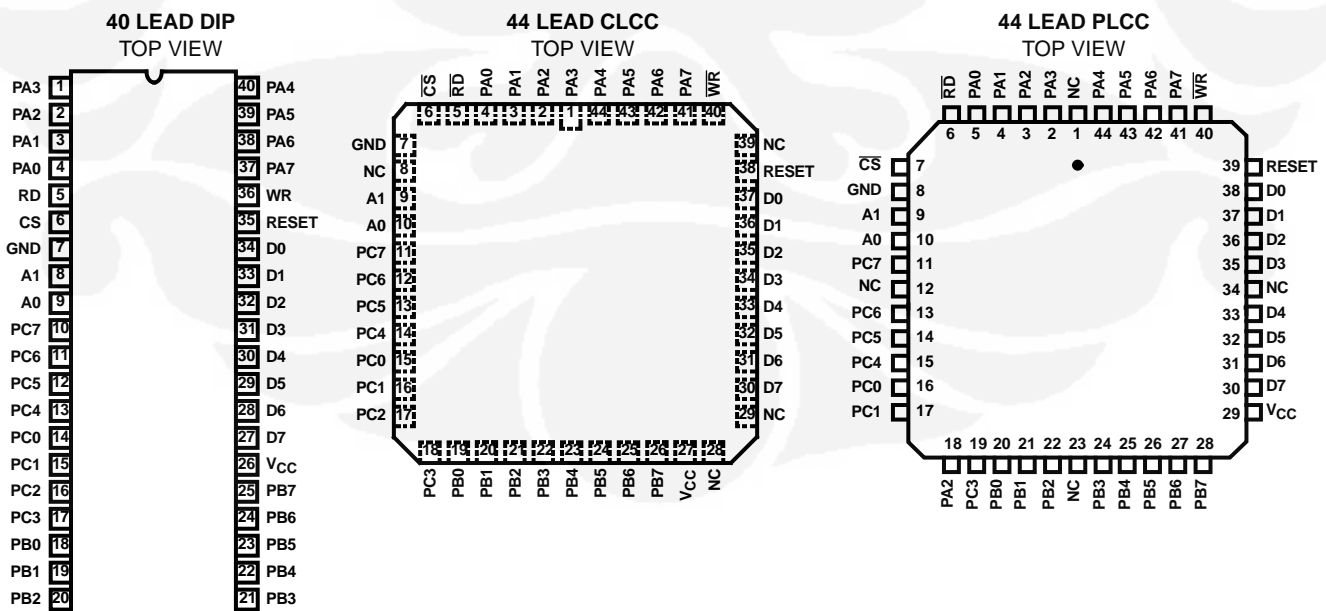
PACKAGE	TEMPERATURE RANGE	5MHz	8MHz	PKG. NO.
Plastic DIP	0°C to +70°C	CP82C55A-5	CP82C55A	E40.6
	-40°C to +85°C	IP82C55A-5	IP82C55A	E40.6
PLCC	0°C to +70°C	CS82C55A-5	CS82C55A	N44.65
	-40°C to +85°C	IS82C55A-5	IS82C55A	N44.65
CERDIP	0°C to +70°C	CD82C55A-5	CD82C55A	F40.6
	-40°C to +85°C	ID82C55A-5	ID82C55A	F40.6
	-55°C to +125°C	MD82C55A-5/B	MD82C55A/B	F40.6
SMD#		8406601QA	8406602QA	F40.6
CLCC	-55°C to +125°C	MR82C55A-5/B	MR82C55A/B	J44.A
		8406601XA	8406602XA	J44.A

Description

The Harris 82C55A is a high performance CMOS version of the industry standard 8255A and is manufactured using a self-aligned silicon gate CMOS process (Scaled SAJI IV). It is a general purpose programmable I/O device which may be used with many different microprocessors. There are 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The high performance and industry standard configuration of the 82C55A make it compatible with the 80C86, 80C88 and other microprocessors.

Static CMOS circuit design insures low operating power. TTL compatibility over the full military temperature range and bus hold circuitry eliminate the need for pull-up resistors. The Harris advanced SAJI process results in performance equal to or greater than existing functionally equivalent products at a fraction of the power

Pinouts

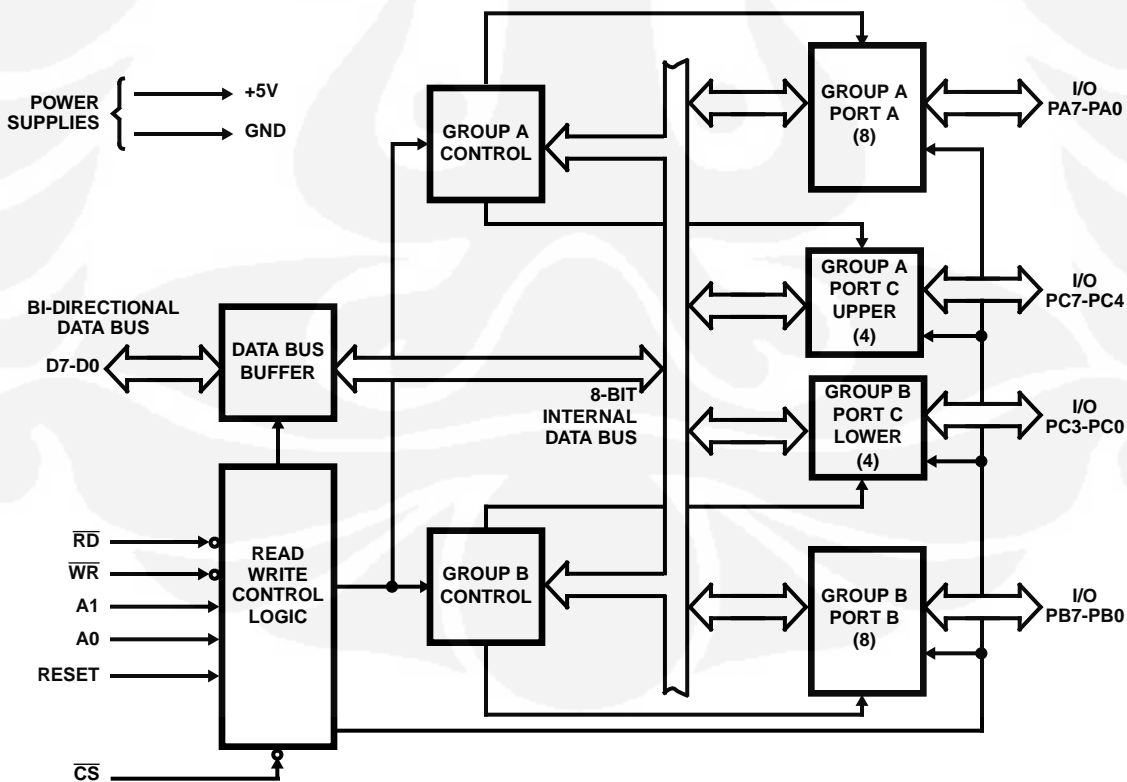


82C55A

Pin Description

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
V _{CC}	26		V _{CC} : The +5V power supply pin. A 0.1μF capacitor between pins 26 and 7 is recommended for decoupling.
GND	7		GROUND
D0-D7	27-34	I/O	DATA BUS: The Data Bus lines are bidirectional three-state pins connected to the system data bus.
RESET	35	I	RESET: A high on this input clears the control register and all ports (A, B, C) are set to the input mode with the "Bus Hold" circuitry turned on.
\overline{CS}	6	I	CHIP SELECT: Chip select is an active low input used to enable the 82C55A onto the Data Bus for CPU communications.
\overline{RD}	5	I	READ: Read is an active low input control signal used by the CPU to read status information or data via the data bus.
\overline{WR}	36	I	WRITE: Write is an active low input control signal used by the CPU to load control words and data into the 82C55A.
A0-A1	8, 9	I	ADDRESS: These input signals, in conjunction with the \overline{RD} and \overline{WR} inputs, control the selection of one of the three ports or the control word register. A0 and A1 are normally connected to the least significant bits of the Address Bus A0, A1.
PA0-PA7	1-4, 37-40	I/O	PORT A: 8-bit input and output port. Both bus hold high and bus hold low circuitry are present on this port.
PB0-PB7	18-25	I/O	PORT B: 8-bit input and output port. Bus hold high circuitry is present on this port.
PC0-PC7	10-17	I/O	PORT C: 8-bit input and output port. Bus hold circuitry is present on this port.

Functional Description



Functional Description

Data Bus Buffer

This three-state bi-directional 8-bit buffer is used to interface the 82C55A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

(CS) Chip Select. A "low" on this input pin enables the communication between the 82C55A and the CPU.

(RD) Read. A "low" on this input pin enables 82C55A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 82C55A.

(WR) Write. A "low" on this input pin enables the CPU to write data or control words into the 82C55A.

(A0 and A1) Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word register. They are normally connected to the least significant bits of the address bus (A0 and A1).

82C55A BASIC OPERATION

A1	A0	RD	WR	CS	INPUT OPERATION (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
OUTPUT OPERATION (WRITE)					
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
DISABLE FUNCTION					
X	X	X	X	1	Data Bus → Three-State
X	X	1	1	0	Data Bus → Three-State

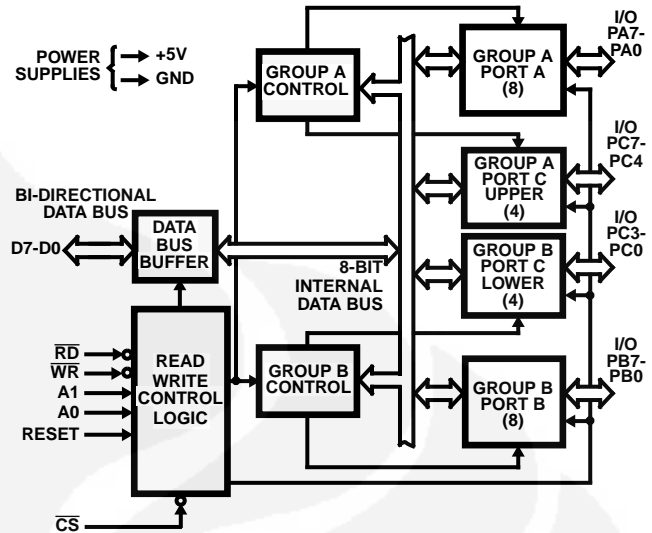


FIGURE 1. 82C55A BLOCK DIAGRAM. DATA BUS BUFFER, READ/WRITE, GROUP A & B CONTROL LOGIC FUNCTIONS

(RESET) Reset. A "high" on this input initializes the control register to 9Bh and all ports (A, B, C) are set to the input mode. "Bus hold" devices internal to the 82C55A will hold the I/O port inputs to a logic "1" state with a maximum hold current of 400µA.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 82C55A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 82C55A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A - Port A and Port C upper (C7 - C4)

Control Group B - Port B and Port C lower (C3 - C0)

The control word register can be both written and read as shown in the "Basic Operation" table. Figure 4 shows the control word format for both Read and Write operations. When the control word is read, bit D7 will always be a logic "1", as this implies control word mode information.

Ports A, B, and C

The 82C55A contains three 8-bit ports (A, B, and C). All can be configured to a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 82C55A.

Port A One 8-bit data output latch/buffer and one 8-bit data input latch. Both "pull-up" and "pull-down" bus-hold devices are present on Port A. See Figure 2A.

Port B One 8-bit data input/output latch/buffer and one 8-bit data input buffer. See Figure 2B.

Port C One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal output and status signal inputs in conjunction with ports A and B. See Figure 2B.

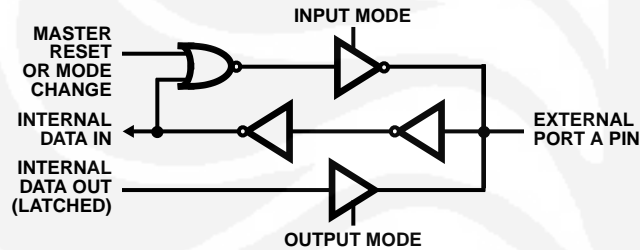


FIGURE 2A. PORT A BUS-HOLD CONFIGURATION

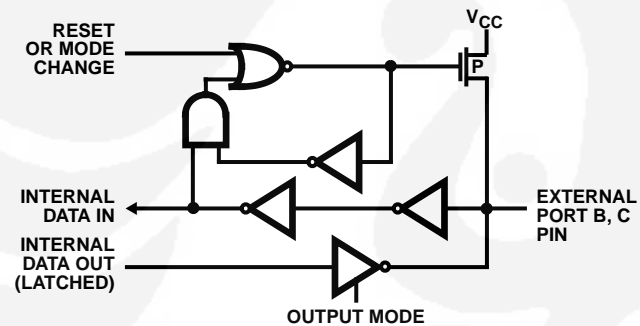


FIGURE 2B. PORT B AND C BUS-HOLD CONFIGURATION

FIGURE 2. BUS-HOLD CONFIGURATION

Operational Description

Mode Selection

There are three basic modes of operation that can be selected by the system software:

- Mode 0 - Basic Input/Output
- Mode 1 - Strobed Input/Output
- Mode 2 - Bi-directional Bus

When the reset input goes "high", all ports will be set to the input mode with all 24 port lines held at a logic "one" level by internal bus hold devices. After the reset is removed, the 82C55A can remain in the input mode with no additional initialization required. This eliminates the need to pullup or pull-down resistors in all-CMOS designs. The control word register will contain 9Bh. During the execution of the system

program, any of the other modes may be selected using a single output instruction. This allows a single 82C55A to service a variety of peripheral devices with a simple software maintenance routine. Any port programmed as an output port is initialized to all zeros when the control word is written.

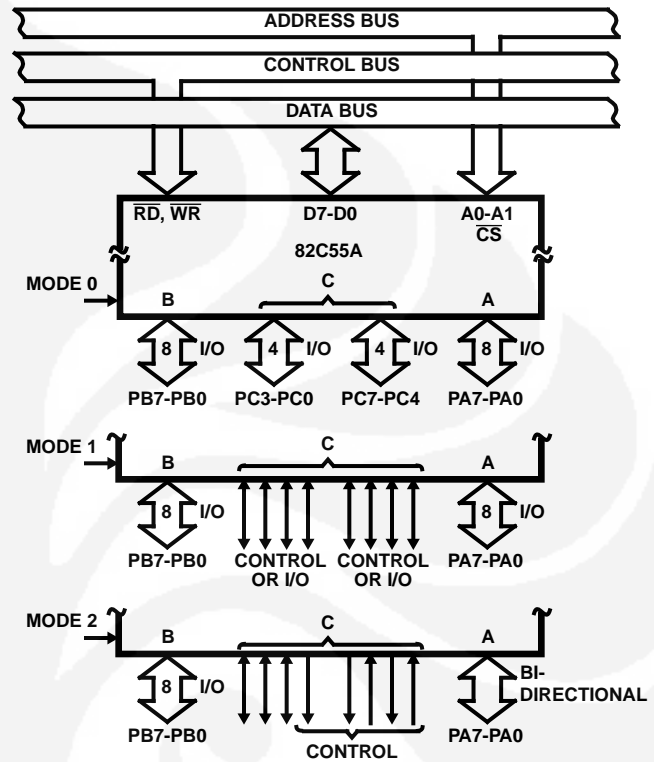


FIGURE 3. BASIC MODE DEFINITIONS AND BUS INTERFACE

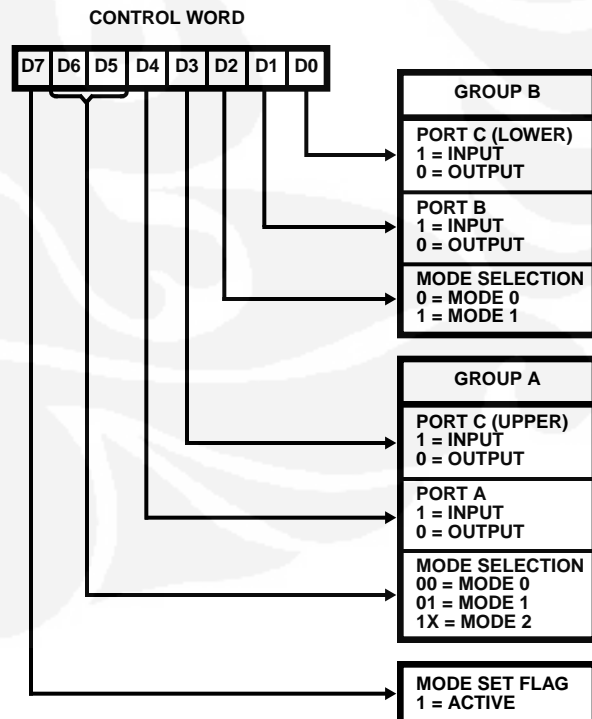


FIGURE 4. MODE DEFINITION FORMAT

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance: Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

The mode definitions and possible mode combinations may seem confusing at first, but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 82C55A has taken into account things such as efficient PC board layout, control signal definition vs. PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature (Figure 5)

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were output ports.

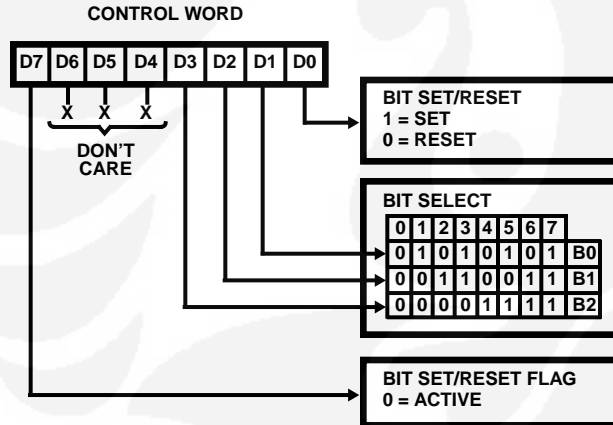


FIGURE 5. BIT SET/RESET FORMAT

Interrupt Control Functions

When the 82C55A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the programmer to enable or disable a CPU interrupt by a specific I/O device without affecting any other device in the interrupt structure.

INTE Flip-Flop Definition

(BIT-SET)-INTE is SET - Interrupt Enable

(BIT-RESET)-INTE is Reset - Interrupt Disable

NOTE: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

Mode 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No handshaking is required, data is simply written to or read from a specific port.

Mode 0 Basic Functional Definitions:

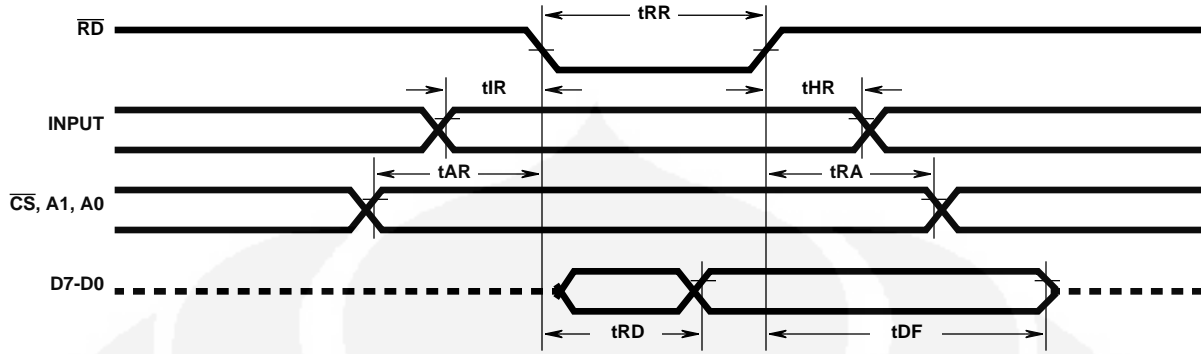
- Two 8-bit ports and two 4-bit ports
- Any Port can be input or output
- Outputs are latched
- Input are not latched
- 16 different Input/Output configurations possible

MODE 0 PORT DEFINITION

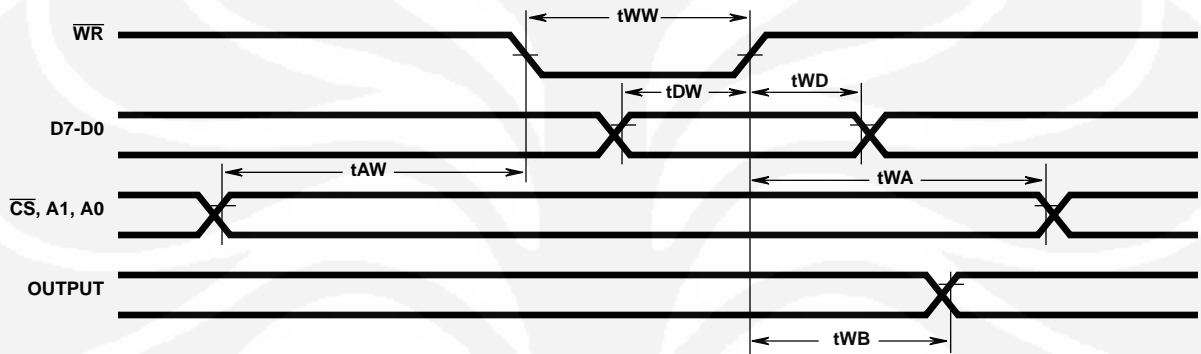
A		B		GROUP A		#	GROUP B	
D4	D3	D1	D0	PORT A	PORTC (Upper)		PORT B	PORTC (Lower)
0	0	0	0	Output	Output	0	Output	Output
0	0	0	1	Output	Output	1	Output	Input
0	0	1	0	Output	Output	2	Input	Output
0	0	1	1	Output	Output	3	Input	Input
0	1	0	0	Output	Input	4	Output	Output
0	1	0	1	Output	Input	5	Output	Input
0	1	1	0	Output	Input	6	Input	Output
0	1	1	1	Output	Input	7	Input	Input
1	0	0	0	Input	Output	8	Output	Output
1	0	0	1	Input	Output	9	Output	Input
1	0	1	0	Input	Output	10	Input	Output
1	0	1	1	Input	Output	11	Input	Input
1	1	0	0	Input	Input	12	Output	Output
1	1	0	1	Input	Input	13	Output	Input
1	1	1	0	Input	Input	14	Input	Output
1	1	1	1	Input	Input	15	Input	Input

82C55A

Mode 0 (Basic Input)

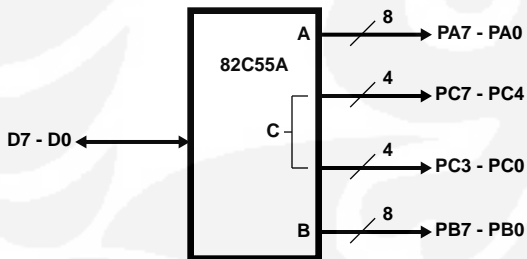
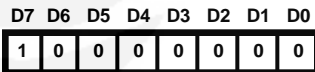


Mode 0 (Basic Output)

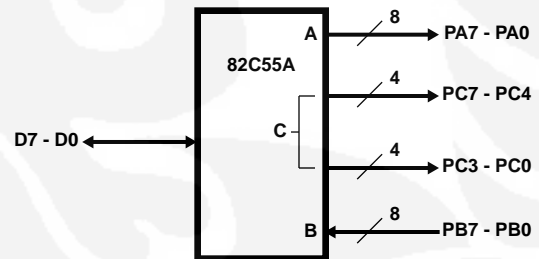
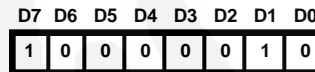


Mode 0 Configurations

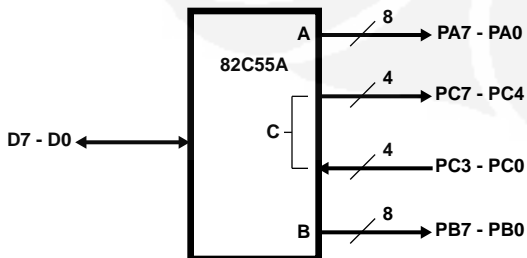
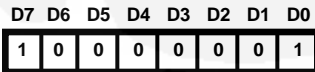
CONTROL WORD #0



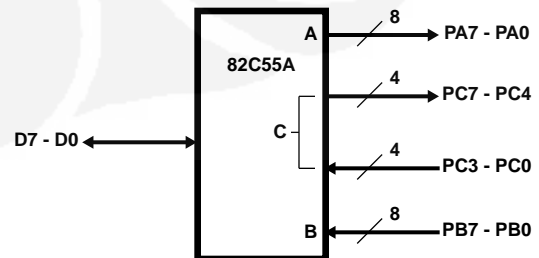
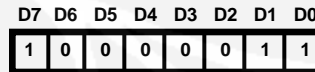
CONTROL WORD #2



CONTROL WORD #1



CONTROL WORD #3

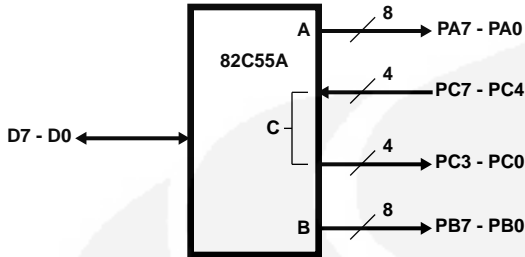


82C55A

Mode 0 Configurations (Continued)

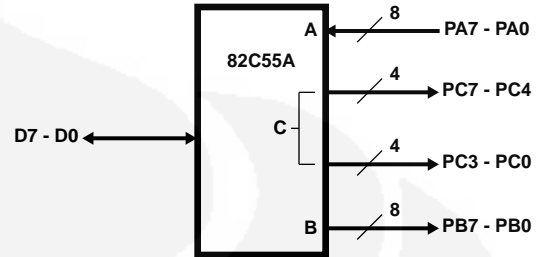
CONTROL WORD #4

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	0	0



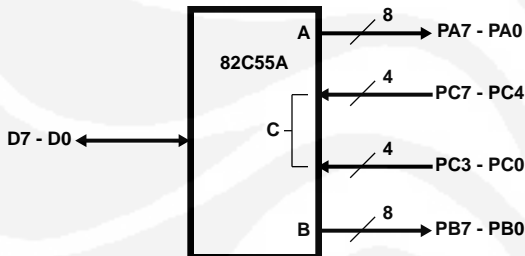
CONTROL WORD #8

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	0



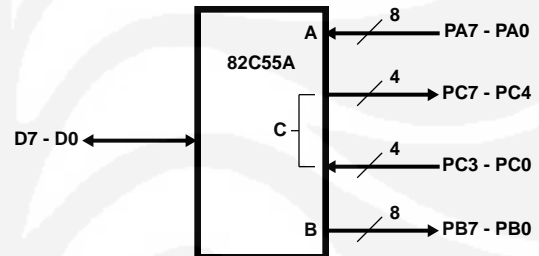
CONTROL WORD #5

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	0	1



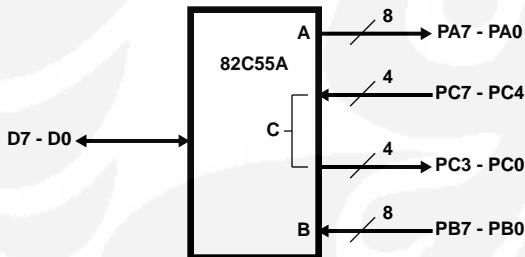
CONTROL WORD #9

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	1



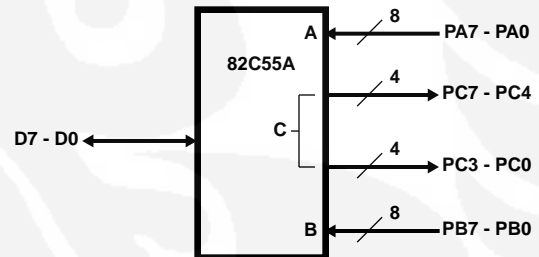
CONTROL WORD #6

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	1	0



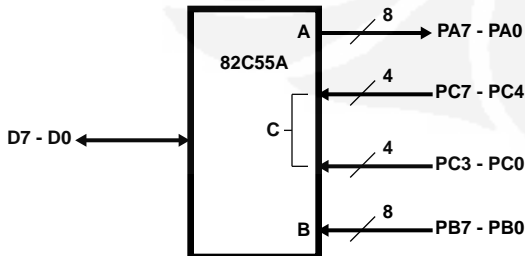
CONTROL WORD #10

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	0



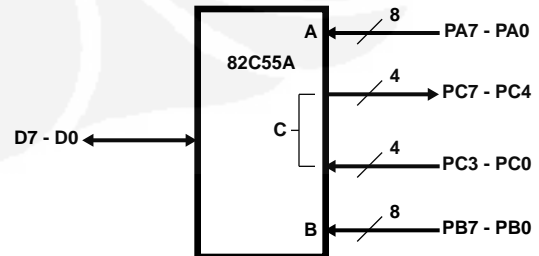
CONTROL WORD #7

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	1	1



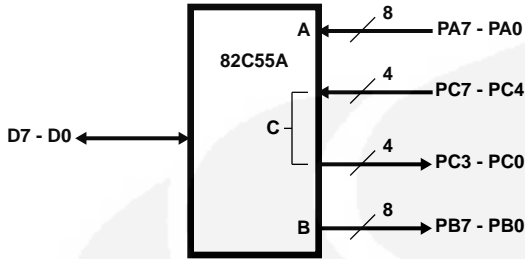
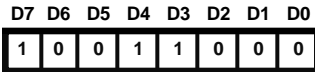
CONTROL WORD #11

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	1

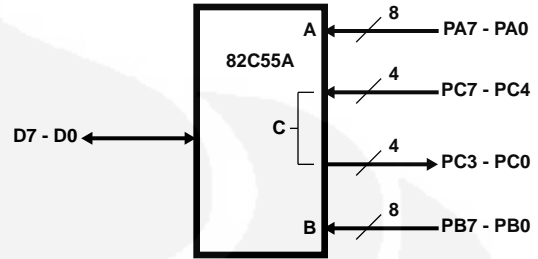
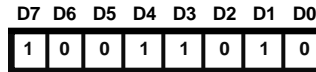


Mode 0 Configurations (Continued)

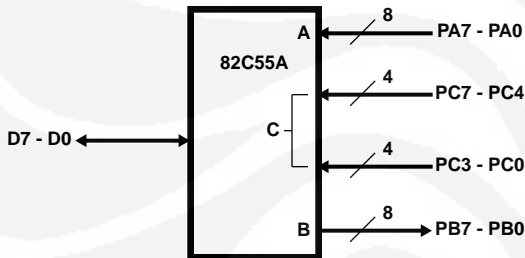
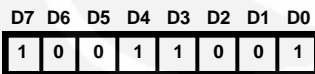
CONTROL WORD #12



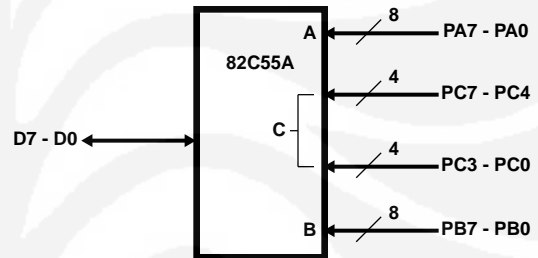
CONTROL WORD #14



CONTROL WORD #13



CONTROL WORD #15



Operating Modes

Mode 1 - (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "hand shaking" signals. In mode 1, port A and port B use the lines on port C to generate or accept these "hand shaking" signals.

Mode 1 Basic Function Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit port and one 4-bit control/data port
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit port.

Input Control Signal Definition

(Figures 6 and 7)

STB (Strobe Input)

A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch: in essence, and acknowledgment. IBF is set by \overline{STB} input being low and is reset by the rising edge of the \overline{RD} input.

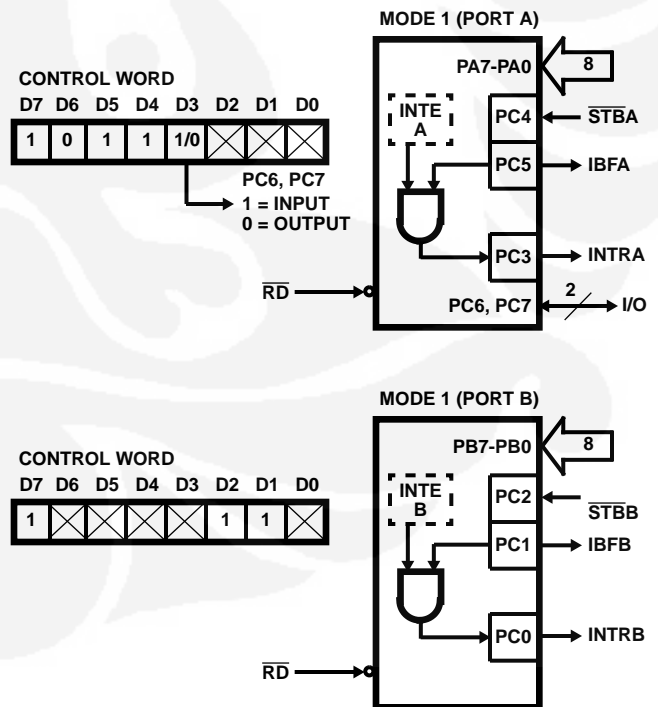


FIGURE 6. MODE 1 INPUT



High Speed Super Low Power SRAM

8K-Word By 8 Bit

WS6264

■ GENERAL DESCRIPTION

The WS6264 is a high performance, high speed and super low power CMOS Static Random Access Memory organized as 8,192 words by 8bits and operates from a single 4.5V to 5.5V supply voltage. Advanced CMOS technology and circuit techniques provide both high speed, super low power features and maximum access time of 70ns in 5.0V operation. Easy memory expansion is provided by using two chip enable inputs (/CE1, CE2) and active LOW output enable (/OE).

The WS6264 has an automatic power down feature, reducing the power consumption significantly when chip is deselected. The WS6264 is available in JEDEC standard 28-pin SOP(300 mil) and PDIP (600 mil) packages.

■ FEATURES

- Operation voltage : 4.5 ~ 5.5V
- Ultra low power consumption:
Operating current 1mA@1MHz & CMOS standby current 1.0uA (Typ.) in Vcc=5.0V
- High speed access time: 70ns.
- Automatic power down when chip is deselected.
- Three state outputs and TTL compatible.
- Data retention supply voltage as low as 2.0V.
- Easy expansion with /CE1, CE2 and /OE options.

■ PRODUCT FAMILY

Product Family	Operating Temp.	Vcc Range	Speed (ns)	Standby Current (Typ.) I _{CCSB1}	Package Type
WS6264LLFP	0~70°C	4.5~5.5V	70	1.0uA	28 SOP
WS6264LLP					28 PDIP
WS6264LLFPI	-40~85°C		70	1.0uA	28 SOP
WS6264LLPI					28 PDIP

Rev. 1.0

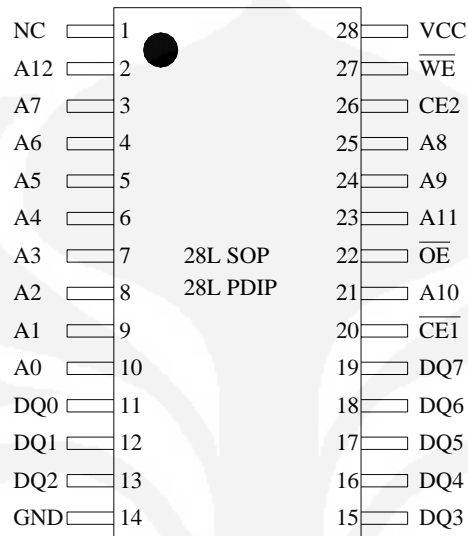


High Speed Super Low Power SRAM

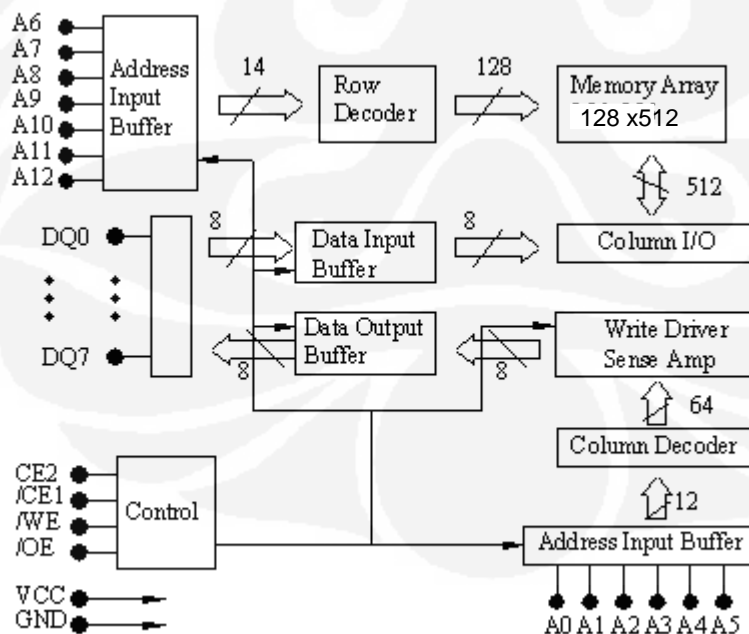
8K-Word By 8 Bit

WS6264

■ PIN CONFIGURATIONS



■ FUNCTIONAL BLOCK DIAGRAM



Rev. 1.0



High Speed Super Low Power SRAM

8K-Word By 8 Bit

WS6264

■ PIN DESCRIPTIONS

Name	Type	Function
A0 – A12	Input	Address inputs for selecting one of the 8,192 x 8 bit words in the RAM
/CE1,CE2	Input	/CE1 is active LOW and CE2 is active HIGH. Both chip enables must be active when data read from or write to the device. If either chip enable is not active, the device is deselected and in a standby power down mode. The DQ pins will be in high impedance state when the device is deselected.
/WE	Input	The Write enable input is active LOW. It controls read and write operations. With the chip selected, when /WE is HIGH and /OE is LOW, output data will be present on the DQ pins, when /WE is LOW, the data present on the DQ pins will be written into the selected memory location.
/OE	Input	The output enable input is active LOW. If the output enable is active while the chip is selected and the write enable is inactive, data will be present on the DQ pins and they will be enabled. The DQ pins will be in the high impedance state when /OE is inactive.
DQ0~DQ7	I/O	These 8 bi-directional ports are used to read data from or write data into the RAM.
Vcc	Power	Power Supply
Gnd	Power	Ground
NC		No connection

■ TRUTH TABLE

MODE	/CE1	CE2	/WE	/OE	DQ0~7	Vcc Current
Standby	H	X	X	X	High Z	I _{CCSB} , I _{CCSB1}
	X	L	X	X		
Output Disable	L	H	H	H	High Z	I _{CC}
Read	L	H	H	L	D _{OUT}	I _{CC}
Write	L	H	L	X	D _{IN}	I _{CC}

Rev. 1.0



High Speed Super Low Power SRAM

8K-Word By 8 Bit

WS6264

■ ABSOLUTE MAXIMUM RATINGS⁽¹⁾

Symbol	Parameter	Rating	Unit
V _{TERM}	Terminal Voltage with Respect to GND	-0.5 to V _{CC} +0.5	V
T _{BIAS}	Temperature Under Bias	-40 to +125	°C
T _{STG}	Storage Temperature	-65 to +150	°C
P _T	Power Dissipation	1.0	W
I _{OUT}	DC Output Current	50	mA

1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

■ OPERATING RANGE

Range	Ambient Temperature	V _{CC}
Commercial	0~70°C	4.5 ~ 5.5V
Industrial	-40~85°C	4.5 ~ 5.5V

■ CAPACITANCE⁽¹⁾(T_A=25°C, f=1.0MHz)

Symbol	Parameter	Conduction	MAX.	Unit
C _{IN}	Input Capacitance	V _{IN} =0V	8	pF
C _{DQ}	Input/Output Capacitance	V _{DI/O} =0V	10	pF

1. This parameter is guaranteed, and not 100% tested.

Rev. 1.0



High Speed Super Low Power SRAM

8K-Word By 8 Bit

WS6264

■ DC ELECTRICAL CHARACTERISTICS (TA = 0° ~70°C, VCC = 5.0V)

Name	Parameter	Test Condition	MIN	TYP ⁽¹⁾	MAX	Unit
V _{IL}	Guaranteed Input Low Voltage ⁽²⁾	V _{CC} =5.0V	-0.5		0.8	V
V _{IH}	Guaranteed Input High Voltage ⁽²⁾	V _{CC} =5.0V	2.2		V _{CC} +0.5	V
I _{IL}	Input Leakage Current	V _{CC} =MAX, V _{IN} =0 to V _{CC}	-1		1	uA
I _{OL}	Output Leakage Current	V _{CC} =MAX, /CE1=V _{Ih} , or CE2= V _{IL} , or /OE=V _{Ih} , or /WE= V _{IL} V _{IO} =0V to V _{CC}	-1		1	uA
V _{OL}	Output Low Voltage	V _{CC} =MAX, I _{OL} = 1mA			0.4	V
V _{OH}	Output High Voltage	V _{CC} =MIN, I _{OH} = -1mA	2.4			V
I _{CC}	Operating Power Supply Current	/CE1=V _{IL} , I _{DQ} =0mA, F=F _{MAX} =1/ t _{RC}			30	mA
I _{CCSB}	TTL Standby Supply	/CE1=V _{IH} , I _{DQ} =0mA,			10	mA
I _{CCSB1}	CMOS Standby Current	/CE1 ≥ V _{CC} -0.2V, CE2= 0.2V, V _{IN} ≥ V _{CC} -0.2V or V _{IN} ≤ 0.2V,		1	10	uA

1. Typical characteristics are at TA = 25°C.

2. These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.



+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where $\pm 12V$ is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than $5\mu W$. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers
Low-Power Modems
Interface Translation
Battery-Powered RS-232 Systems
Multidrop RS-232 Networks

Next-Generation Device Features

- ◆ For Low-Voltage, Integrated ESD Applications
MAX3222E/MAX3232E/MAX3237E/MAX3241E/
MAX3246E: +3.0V to +5.5V, Low-Power, Up to
1Mbps, True RS-232 Transceivers Using Four
0.1 μF External Capacitors (MAX3246E Available
in a UCSP™ Package)
- ◆ For Low-Cost Applications
MAX221E: $\pm 15kV$ ESD-Protected, +5V, 1 μA ,
Single RS-232 Transceiver with AutoShutdown™

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

AutoShutdown and UCSP are trademarks of Maxim Integrated Products, Inc.

Ordering Information continued at end of data sheet.

*Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.047/0.33	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package



For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-integrated.com.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V _{CC})	-0.3V to +6V	18-Pin Plastic DIP (derate 11.1mW/°C above +70°C)	..889mW
V+ (Note 1)	(V _{CC} - 0.3V) to +14V	20-Pin Plastic DIP (derate 8.00mW/°C above +70°C)	..440mW
V- (Note 1)	+0.3V to +14V	16-Pin Narrow SO (derate 8.70mW/°C above +70°C)	..696mW
Input Voltages		16-Pin Wide SO (derate 9.52mW/°C above +70°C)762mW
T _{IN}	-0.3V to (V _{CC} - 0.3V)	18-Pin Wide SO (derate 9.52mW/°C above +70°C)762mW
R _{IN} (Except MAX220)	±30V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)800mW
R _{IN} (MAX220)	±25V	20-Pin SSOP (derate 8.00mW/°C above +70°C)640mW
T _{OUT} (Except MAX220) (Note 2)	±15V	16-Pin CERDIP (derate 10.00mW/°C above +70°C)800mW
T _{OUT} (MAX220)	±13.2V	18-Pin CERDIP (derate 10.53mW/°C above +70°C)842mW
Output Voltages		Operating Temperature Ranges	
T _{OUT}	±15V	MAX2_AC_, MAX2_C_0°C to +70°C
R _{OUT}	-0.3V to (V _{CC} + 0.3V)	MAX2_AE_, MAX2_E_-40°C to +85°C
Driver/Receiver Output Short Circuited to GND	Continuous	MAX2_AM_, MAX2_M_-55°C to +125°C
Continuous Power Dissipation (T _A = +70°C)		Storage Temperature Range-65°C to +160°C
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)	..842mW	Lead Temperature (soldering, 10s) (Note 3)+300°C

Note 1: For the MAX220, V+ and V- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Note 2: Input voltage measured with T_{OUT} in high-impedance state, SHDN or V_{CC} = 0V.

Note 3: Maximum reflow temperature for the MAX233A is +225°C.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High	All devices except MAX220		2	1.4		V
	MAX220: V _{CC} = 5.0V		2.4			
Logic Pullup/Input Current	All except MAX220, normal operation			5	40	µA
	SHDN = 0V, MAX222/MAX242, shutdown, MAX220			±0.01	±1	
Output Leakage Current	V _{CC} = 5.5V, SHDN = 0V, V _{OUT} = ±15V, MAX222/MAX242			±0.01	±10	µA
	V _{CC} = SHDN = 0V	V _{OUT} = ±15V		±0.01	±10	
		MAX220, V _{OUT} = ±12V			±25	
Data Rate				200	116	kbps
Transmitter Output Resistance	V _{CC} = V+ = V- = 0V, V _{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V _{OUT} = 0V	V _{OUT} = 0V	±7	±22		mA
		MAX220			±60	
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
	MAX220				±25	
RS-232 Input Threshold Low	V _{CC} = 5V	All except MAX243 R _{2IN}	0.8	1.3		V
		MAX243 R _{2IN} (Note 4)	-3			
RS-232 Input Threshold High	V _{CC} = 5V	All except MAX243 R _{2IN}		1.8	2.4	V
		MAX243 R _{2IN} (Note 4)		-0.5	-0.1	

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C1–C4 = 0.1μF, MAX220, C1 = 0.047μF, C2–C4 = 0.33μF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 Input Hysteresis	All except MAX220/MAX243, V _{CC} = 5V, no hysteresis in SHDN		0.2	0.5	1.0	V
	MAX220			0.3		
	MAX243			1		
RS-232 Input Resistance	T _A = +25°C (MAX220)		3	5	7	kΩ
			3	5	7	
TTL/CMOS Output Voltage Low	I _{OUT} = 3.2mA			0.2	0.4	V
	I _{OUT} = 1.6mA (MAX220)				0.4	
TTL/CMOS Output Voltage High	I _{OUT} = -1.0mA		3.5	V _{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V _{OUT} = GND		-2	-10		mA
	Sinking V _{OUT} = V _{CC}		10	30		
TTL/CMOS Output Leakage Current	SHDN = V _{CC} or EN = V _{CC} (SHDN = 0V for MAX222), 0V ≤ V _{OUT} ≤ V _{CC}			±0.05	±10	μA
EN Input Threshold Low	MAX242			1.4	0.8	V
EN Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5		5.5	V
V _{CC} Supply Current (SHDN = V _{CC}), Figures 5, 6, 11, 19	No load	MAX220		0.5	2	μA
		MAX222/MAX232A/MAX233A/ MAX242/MAX243		4	10	
	3kΩ load both inputs	MAX220		12		
		MAX222/MAX232A/MAX233A/ MAX242/MAX243		15		
Shutdown Supply Current	MAX222/ MAX242	T _A = +25°C		0.1	10	μA
		T _A = 0°C to +70°C		2	50	
		T _A = -40°C to +85°C		2	50	
		T _A = -55°C to +125°C		35	100	
SHDN Input Leakage Current	MAX222/MAX242				±1	μA
SHDN Threshold Low	MAX222/MAX242			1.4	0.8	V
SHDN Threshold High	MAX222/MAX242		2.0	1.4		V
Transition Slew Rate	C _L = 50pF to 2500pF, R _L = 3kΩ to 7kΩ, V _{CC} = 5V, T _A = +25°C, measured from +3V to -3V or -3V	MAX222/MAX232A/MAX233/ MAX242/MAX243	6	12	30	V/μs
		MAX220	1.5	3	30.0	
Transmitter Propagation Delay TLL to RS-232 (Normal Operation), Figure 1	t _{PHLT}	MAX222/MAX232A/MAX233/ MAX242/MAX243		1.3	3.5	μs
		MAX220		4	10	
	t _{PLHT}	MAX222/MAX232A/MAX233/ MAX242/MAX243		1.5	3.5	
		MAX220		5	10	

Note 4: MAX243 R_{2OUT} is guaranteed to be low when R_{2IN} is ≥ 0V or is floating.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Detailed Description

The MAX220–MAX249 contain four sections: dual charge-pump DC-DC voltage converters, RS-232 drivers, RS-232 receivers, and receiver and transmitter enable control inputs.

Dual Charge-Pump Voltage Converter

The MAX220–MAX249 have two internal charge-pumps that convert +5V to $\pm 10V$ (unloaded) for RS-232 driver operation. The first converter uses capacitor C1 to double the +5V input to +10V on C3 at the V+ output. The second converter uses capacitor C2 to invert +10V to -10V on C4 at the V- output.

A small amount of power may be drawn from the +10V (V+) and -10V (V-) outputs to power external circuitry (see the *Typical Operating Characteristics* section), except on the MAX225 and MAX245–MAX247, where these pins are not available. V+ and V- are not regulated, so the output voltage drops with increasing load current. Do not load V+ and V- to a point that violates the minimum $\pm 5V$ EIA/TIA-232E driver output voltage when sourcing current from V+ and V- to external circuitry.

When using the shutdown feature in the MAX222, MAX225, MAX230, MAX235, MAX236, MAX240, MAX241, and MAX245–MAX249, avoid using V+ and V- to power external circuitry. When these parts are shut down, V- falls to 0V, and V+ falls to +5V. For applications where a +10V external supply is applied to the V+ pin (instead of using the internal charge pump to generate +10V), the C1 capacitor must not be installed and the SHDN pin must be tied to VCC. This is because V+ is internally connected to VCC in shutdown mode.

RS-232 Drivers

The typical driver output voltage swing is $\pm 8V$ when loaded with a nominal 5k Ω RS-232 receiver and VCC = +5V. Output swing is guaranteed to meet the EIA/TIA-232E and V.28 specification, which calls for $\pm 5V$ minimum driver output levels under worst-case conditions. These include a minimum 3k Ω load, VCC = +4.5V, and maximum operating temperature. Unloaded driver output voltage ranges from (V+ -1.3V) to (V- +0.5V).

Input thresholds are both TTL and CMOS compatible. The inputs of unused drivers can be left unconnected since 400k Ω input pullup resistors to VCC are built in (except for the MAX220). The pullup resistors force the outputs of unused drivers low because all drivers invert. The internal input pullup resistors typically source 12 μA , except in shutdown mode where the pullups are disabled. Driver outputs turn off and enter a high-impedance state—where leakage current is typically microamperes (maximum 25 μA)—when in shutdown

mode, in three-state mode, or when device power is removed. Outputs can be driven to $\pm 15V$. The power-supply current typically drops to 8 μA in shutdown mode. The MAX220 does not have pullup resistors to force the outputs of the unused drivers low. Connect unused inputs to GND or VCC.

The MAX239 has a receiver three-state control line, and the MAX223, MAX225, MAX235, MAX236, MAX240, and MAX241 have both a receiver three-state control line and a low-power shutdown control. Table 2 shows the effects of the shutdown control and receiver three-state control on the receiver outputs.

The receiver TTL/CMOS outputs are in a high-impedance, three-state mode whenever the three-state enable line is high (for the MAX225/MAX235/MAX236/MAX239–MAX241), and are also high-impedance whenever the shutdown control line is high.

When in low-power shutdown mode, the driver outputs are turned off and their leakage current is less than 1 μA with the driver output pulled to ground. The driver output leakage remains less than 1 μA , even if the transmitter output is backdriven between 0V and (VCC + 6V). Below -0.5V, the transmitter is diode clamped to ground with 1k Ω series impedance. The transmitter is also zener clamped to approximately VCC + 6V, with a series impedance of 1k Ω .

The driver output slew rate is limited to less than 30V/ μs as required by the EIA/TIA-232E and V.28 specifications. Typical slew rates are 24V/ μs unloaded and 10V/ μs loaded with 3 Ω and 2500pF.

RS-232 Receivers

EIA/TIA-232E and V.28 specifications define a voltage level greater than 3V as a logic 0, so all receivers invert. Input thresholds are set at 0.8V and 2.4V, so receivers respond to TTL level inputs as well as EIA/TIA-232E and V.28 levels.

The receiver inputs withstand an input overvoltage up to $\pm 25V$ and provide input terminating resistors with

Table 2. Three-State Control of Receivers

PART	SHDN	SHDN	EN	EN(R)	RECEIVERS
MAX223	—	Low High High	X Low High	—	High Impedance Active High Impedance
MAX225	—	—	—	Low High	High Impedance Active
MAX235 MAX236 MAX240	Low Low High	—	—	Low High X	High Impedance Active High Impedance

+5V-Powered, Multichannel RS-232 Drivers/Receivers

nominal 5k Ω values. The receivers implement Type 1 interpretation of the fault conditions of V.28 and EIA/TIA-232E.

The receiver input hysteresis is typically 0.5V with a guaranteed minimum of 0.2V. This produces clear output transitions with slow-moving input signals, even with moderate amounts of noise and ringing. The receiver propagation delay is typically 600ns and is independent of input swing direction.

Low-Power Receive Mode

The low-power receive mode feature of the MAX223, MAX242, and MAX245-MAX249 puts the IC into shutdown mode but still allows it to receive information. This is important for applications where systems are periodically awakened to look for activity. Using low-power receive mode, the system can still receive a signal that will activate it on command and prepare it for communication at faster data rates. This operation conserves system power.

Negative Threshold—MAX243

The MAX243 is pin compatible with the MAX232A, differing only in that RS-232 cable fault protection is removed on one of the two receiver inputs. This means that control lines such as CTS and RTS can either be driven or left floating without interrupting communication. Different cables are not needed to interface with different pieces of equipment.

The input threshold of the receiver without cable fault protection is -0.8V rather than +1.4V. Its output goes positive only if the input is connected to a control line that is actively driven negative. If not driven, it defaults to the 0 or "OK to send" state. Normally, the MAX243's other receiver (+1.4V threshold) is used for the data line (TD or RD), while the negative threshold receiver is connected to the control line (DTR, DTS, CTS, RTS, etc.).

Other members of the RS-232 family implement the optional cable fault protection as specified by EIA/TIA-232E specifications. This means a receiver output goes high whenever its input is driven negative, left floating, or shorted to ground. The high output tells the serial communications IC to stop sending data. To avoid this, the control lines must either be driven or connected with jumpers to an appropriate positive voltage level.

Shutdown—MAX222-MAX242

On the MAX222, MAX235, MAX236, MAX240, and MAX241, all receivers are disabled during shutdown. On the MAX223 and MAX242, two receivers continue to operate in a reduced power mode when the chip is in shutdown. Under these conditions, the propagation delay increases to about 2.5 μ s for a high-to-low input transition. When in shutdown, the receiver acts as a CMOS inverter with no hysteresis. The MAX223 and MAX242 also have a receiver output enable input ($\overline{\text{EN}}$ for the MAX242 and EN for the MAX223) that allows receiver output control independent of $\overline{\text{SHDN}}$ (SHDN for MAX241). With all other devices, $\overline{\text{SHDN}}$ (SHDN for MAX241) also disables the receiver outputs.

The MAX225 provides five transmitters and five receivers, while the MAX245 provides ten receivers and eight transmitters. Both devices have separate receiver and transmitter-enable controls. The charge pumps turn off and the devices shut down when a logic high is applied to the ENT input. In this state, the supply current drops to less than 25 μ A and the receivers continue to operate in a low-power receive mode. Driver outputs enter a high-impedance state (three-state mode). On the MAX225, all five receivers are controlled by the $\overline{\text{ENR}}$ input. On the MAX245, eight of the receiver outputs are controlled by the $\overline{\text{ENR}}$ input, while the remaining two receivers (RA5 and RB5) are always active. RA1-RA4 and RB1-RB4 are put in a three-state mode when $\overline{\text{ENR}}$ is a logic high.

Receiver and Transmitter Enable Control Inputs

The MAX225 and MAX245-MAX249 feature transmitter and receiver enable controls.

The receivers have three modes of operation: full-speed receive (normal active), three-state (disabled), and low-power receive (enabled receivers continue to function at lower data rates). The receiver enable inputs control the full-speed receive and three-state modes. The transmitters have two modes of operation: full-speed transmit (normal active) and three-state (disabled). The transmitter enable inputs also control the shutdown mode. The device enters shutdown mode when all transmitters are disabled. Enabled receivers function in the low-power receive mode when in shutdown.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Tables 1a–1d define the control states. The MAX244 has no control pins and is not included in these tables.

The MAX246 has ten receivers and eight drivers with two control pins, each controlling one side of the device. A logic high at the A-side control input ($\overline{\text{ENA}}$) causes the four A-side receivers and drivers to go into a three-state mode. Similarly, the B-side control input ($\overline{\text{ENB}}$) causes the four B-side drivers and receivers to go into a three-state mode. As in the MAX245, one A-side and one B-side receiver (RA5 and RB5) remain active at all times. The entire device is put into shutdown mode when both the A and B sides are disabled ($\overline{\text{ENA}} = \overline{\text{ENB}} = +5\text{V}$).

The MAX247 provides nine receivers and eight drivers with four control pins. The $\overline{\text{ENRA}}$ and $\overline{\text{ENRB}}$ receiver enable inputs each control four receiver outputs. The $\overline{\text{ENTA}}$ and $\overline{\text{ENTB}}$ transmitter enable inputs each control four drivers. The ninth receiver (RB5) is always active. The device enters shutdown mode with a logic high on both $\overline{\text{ENTA}}$ and $\overline{\text{ENTB}}$.

The MAX248 provides eight receivers and eight drivers with four control pins. The $\overline{\text{ENRA}}$ and $\overline{\text{ENRB}}$ receiver enable inputs each control four receiver outputs. The $\overline{\text{ENTA}}$ and $\overline{\text{ENTB}}$ transmitter enable inputs control four drivers each. This part does not have an always-active receiver. The device enters shutdown mode and transmitters go into a three-state mode with a logic high on both $\overline{\text{ENTA}}$ and $\overline{\text{ENTB}}$.

The MAX249 provides ten receivers and six drivers with four control pins. The $\overline{\text{ENRA}}$ and $\overline{\text{ENRB}}$ receiver enable inputs each control five receiver outputs. The $\overline{\text{ENTA}}$ and $\overline{\text{ENTB}}$ transmitter enable inputs control three drivers each. There is no always-active receiver. The device enters shutdown mode and transmitters go into a three-state mode with a logic high on both $\overline{\text{ENTA}}$ and $\overline{\text{ENTB}}$. In shutdown mode, active receivers operate in a low-power receive mode at data rates up to 20kb/s.

Applications Information

Figures 5 through 25 show pin configurations and typical operating circuits. In applications that are sensitive to power-supply noise, VCC should be decoupled to ground with a capacitor of the same value as C1 and C2 connected as close as possible to the device.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

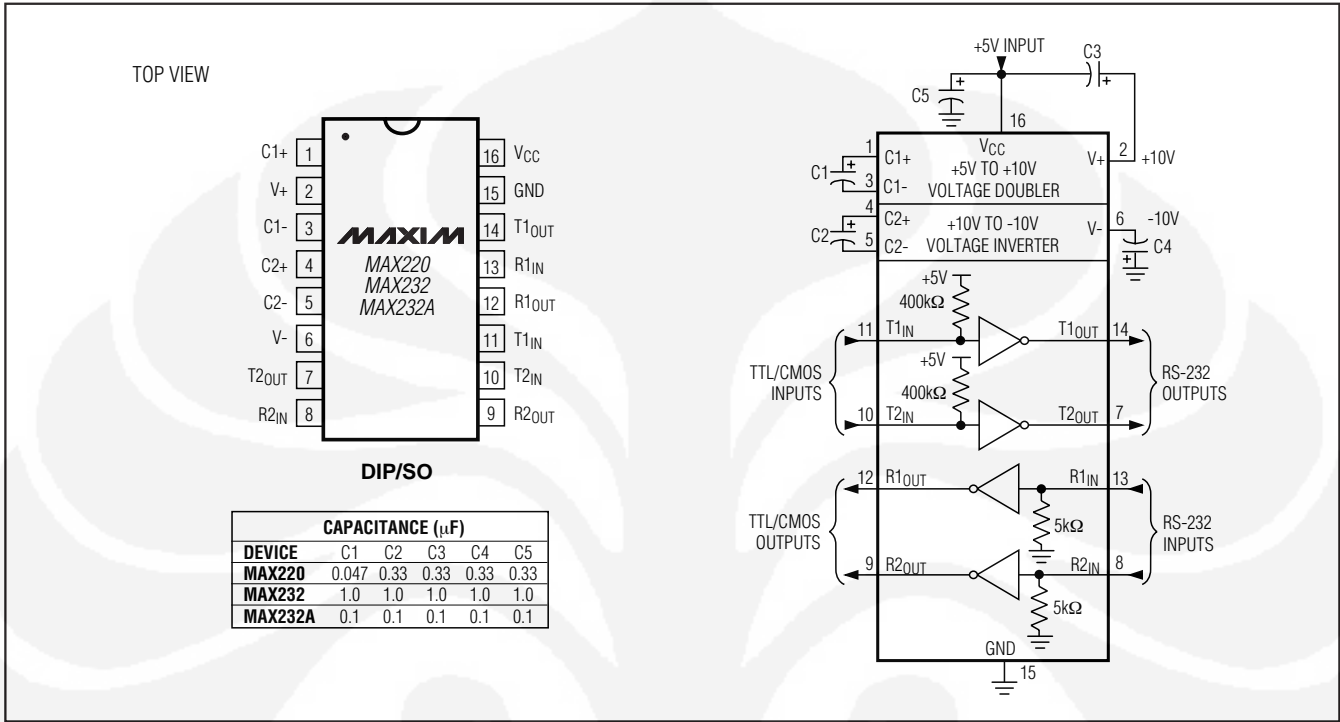


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

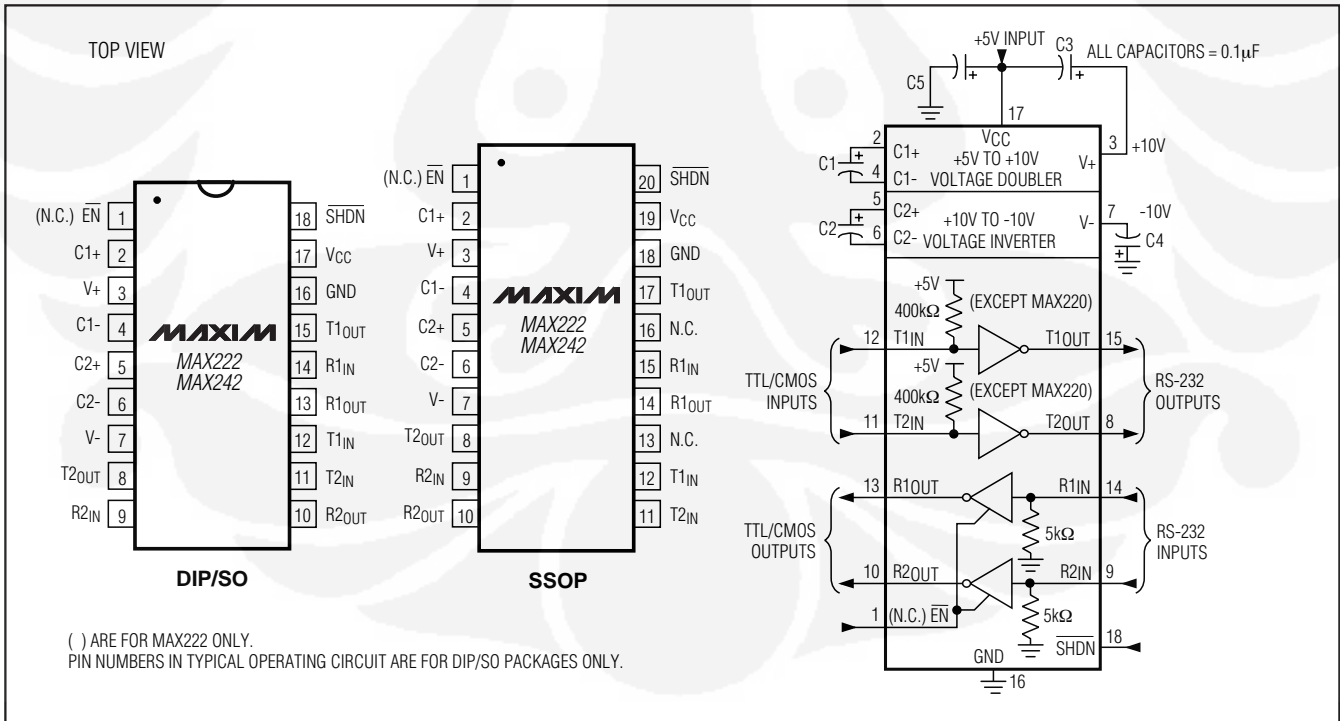


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit