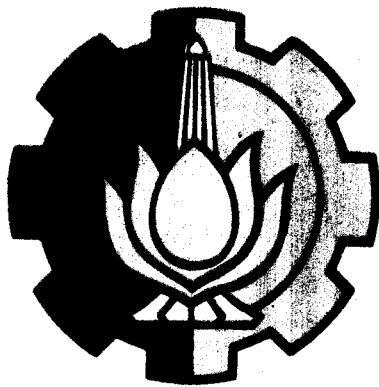


5486/ITS/11/93 ✓

PERPUSTAKAAN	
I T	
Tgl. Terima	18 DEC 1992
No. ...	TA
No. ...	665/B

PERENCANAAN DAN PEMBUATAN ALAT SISTEM PENGATURAN
LAMPU LALU LINTAS YANG TERKOORDINASI
DENGAN MENGGUNAKAN MIKROPROSESSOR 8088




RSE
625.794 028 5
SOL
P-1
1992

Oleh

Mokhammad Solehuddin

Nrp: 2852200390

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1992

	MILIK PERPUSTAKAAN
	INSTITUT TEKNOLOGI
	SEPULUH - NOPEMBER

**PERENCANAAN DAN PEMBUATAN ALAT SISTEM PENGATURAN
LAMPU LALU LINTAS YANG TERKOORDINASI
DENGAN MENGGUNAKAN MIKROPROSESSOR 8088**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar
Sarjana Teknik Elektro
Pada
Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya**

**Mengetahui / Menyetujui
Dosen Pembimbing**

DR. Ir. ABDULLAH ALKAFF

**S U R A B A Y A
NOPEMBER, 1992**

*kupersembahkan untuk
Aba dan Emak tercinta
serta saudara-saudaraku*

ABSTRAK

Adanya dua persimpangan atau lebih yang berdekatan dan berurutan dapat menimbulkan kerugian waktu bagi pemakai jalan, bila lampu lalu lintas yang ada tersebut tidak saling mempengaruhi. Untuk mengatasi hal ini, dapat digunakan beberapa sistem pengatur lalu lintas. Salah satu alternatif sistem tersebut adalah lampu lalu lintas sistem progresip dengan menggunakan mikroprosesor 8088 yang dapat mengatur urutan nyala lampu lalu lintas bagi persimpangan-persimpangan tersebut. Dengan sistem tersebut diharapkan kendaraan yang telah melewati satu persimpangan tidak akan terhalang oleh lampu merah pada persimpangan berikutnya.

Untuk dapat mengaktifkan sistem tersebut, maka dipasang suatu detektor kendaraan pada ruas jalan antar persimpangan yang dapat mendeteksi kepadatan kendaraan di ruas jalan tersebut. Yang akhirnya mikroprosesor dapat menentukan offset yang sesuai pada pengkoordinasian antar persimpangan tersebut.

Dengan membuat model lampu lalu lintas sistem progresip tersebut, dapat diketahui perilaku lampu lalu lintas terhadap setiap kondisi kepadatan kendaraan.

BAB	HALAMAN
JUDUL	i
PENGESAHAN	ii
ABSTRAK	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
KATA PENGANTAR	xi
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batas Permasalahan	3
1.4 Sistematika Pembahasan	4
II. TEORI DASAR SISTEM LALU LINTAS	6
2.1 Pendahuluan	6
2.2 Penentuan Lama Waktu Nyala Lampu	7
2.2.1 Penentuan interval kuning	7
2.2.2 Penentuan interval hijau	8
2.3 Koordinasi Antar persimpangan	10
2.3.1 Sistem sinyal progresip	12
III. MIKROPROSESSOR 8088 BESERTA KOMPONEN PENUNJANG	16
3.1 Pendahuluan	16

BAB	HALAMAN
3.2 Sistem Mikroprosesor	16
3.3 Arsitektur Sistem 3 Bus	17
3.3.1 Sistem address bus	18
3.3.2 Sistem data bus	19
3.4 Unit Memory	20
3.4.1 Read only memory	20
3.4.1.1 Sistem pembacaan data dari ROM	21
3.4.2 Random access memory	23
3.4.2.1 Sistematika pembacaan data dari RAM	23
3.4.2.1 Sistematika penulisan data ke RAM	25
3.5 Mikroprosesor 8088	26
3.5.1 Arsitektur mikroprosesor 8088	26
3.5.2 Sistem mode minimum	29
3.5.3 Sistem mode maksimum	32
3.5.4 Register mikroprosesor 8088	33
3.6 Programable Periperal Interface PPI 8255 ..	38
3.6.1 Blok diagram PPI 8255	39
3.6.2 Pengoperasian PPI 8255A	41
3.6.3 Menghubungkan PPI 8255 dengan mikro prosesor 8088	43
3.7 Address Decoding	44

BAB	HALAMAN
3.8 Buffering	45
IV. PENGAMBILAN DATA DAN PERHITUNGAN	47
4.1 Pendahuluan	47
4.2 Survey Spot Speed	47
4.3 Peramalan Volume Kendaraan	48
4.3.1 Faktor pertumbuhan kendaraan	48
4.3.2 Penyesuaian volume kendaraan ke tahun 1992	51
4.4 Perhitungan Ekuivalensi Volume Kendaraan	53
4.4.1 Persimpangan A	53
4.4.2 Persimpangan B	54
4.5 Perhitungan Interval Nyala Lampu	56
4.5.1 Persimpangan A	56
4.5.1 Persimpangan B	59
4.6 Pengkoordinasian Antar Persimpangan	60
V PERENCANAAN DAN PEMBUATAN ALAT.....	63
5.1 Pendahuluan	63
5.2 Perencanaan CPU 8088 Dalam Minimum Mode	65
5.2.1 Rangkaian clock	65
5.2.2 Rangkaian reset	66
5.2.3 Buffer	67
5.2.4 Control bus	67
5.2.5 Rangkaian decoder memory	68
5.2.6 Rangkaian decoder unit I/O	71

BAB	HALAMAN
5.3 Rangkaian Detektor	71
5.4 Rangkaian Keluaran	73
5.5 Perencanaan Software	73
5.6 Modifikasi Sistem Rancangan	74
5.6.1 Modifikasi rangkaian detektor	74
5.6.2 Modifikasi rangkaian keluaran	77
5.7 Keadaan Khusus	77
VI. KESIMPULAN DAN SARAN	79
DAFTAR PUSTAKA	81
LAMPIRAN	82
LAMPIRAN A	82
- Listing Program	82
- Gambar dan Rangkaian Alat yang dibuat	93
LAMPIRAN B	94
- Kecepatan dan Volume Kendaraan	94
- Type beberapa Detektor Kendaraan	98
LAMPIRAN C DATA SHEET	109

DAFTAR GAMBAR

GAMBAR	HALAMAN
1. 1 SKEMA LOKASI PENELITIAN	3
2. 1 SKETSA RUAS JALAN DENGAN TIGA PERSIMPANGAN . . .	13
2. 2 BAGIAN RUAS JALAN	14
3. 1 SISTEM MIKROPROSESSOR	17
3. 2 ARSITEKTUR SISTEM 3 BUS	18
3. 3 DIAGRAM PEMBACAAN DATA DARI ROM	22
3. 4 DIAGRAM PEMBACAAN DATA DARI RAM	24
3. 5 DIAGRAM WAKTU PENULISAN DATA DARI ROM	25
3. 6 KONFIGURASI PIN 8088	28
3. 7 MODE MINIMUM 8088 MPU	30
3. 8 REGISTER MIKROPROSESSOR 8088	34
3. 9 FLAG REGISTER 8088	37
3.10 BLOK DIAGRAM PPI 8255	40
3.11 FORMAT CONTROL WORD PPI 8255	42
3.12 HUBUNGAN LENGKAP PPI 8255 DENGAN MIKROPROSESOR 8088	44
4. 1 VOLUME KENDARAAN DI LOKASI PENELITIAN TAHUN 1989	49
4. 2 PENYESUAIAN VOLUME KENDARAAN TAHUN 1992	52
4. 3 EKIVALENSI VOLUME KENDARAAN KE SMP	57
5. 1 BLOK DIAGRAM SISTEM YANG DIBUAT	64
5. 2 RANGKAIN CLOCK DAN RESET 8284	66

GAMBAR	HALAMAN
5. 3 RANGKAIAN CONTROL BUS	68
5. 4 MEMORI MAP SISTEM	69
5. 5 RANGKAIAN DECODER ROM	69
5. 6 RANGKAIAN DECODER RAM	70
5. 7 RANGKAIAN DECODER I/O	72
5. 8 RANGKAIAN DETEKTOR KENDARAAN	72
5. 9 RANGKAIAN UNTUK MENYALAKAN LAMPU LED	73
5.10 FLOW CHART DARI SISTEM YANG DIBUAT	75
5.11 RANGKAIAN KELUARAN 8255 DENGAN JALA-JALA PLN	77
5.12 PENANGANAN KEADAAN KHUSUS	78

DAFTAR TABEL

TABEL	HALAMAN
3.1 OPERASI BUS CYCLE MODE MINIMUM	31
3.2 OPERASI BUS CYCLE MODE MAKSIMUM	33
3.3 STATUS INTRUKSI QUEUE	33
3.4 OPERASI REGISTER MIKROPROSESSOR 8088	35
3.5 KEGUNAAN ADDRESS PIN A0, A1, CS	41
4.1 FAKTOR EKIVALENSI KENDARAAN	53
4.2 INTERVAL NYALA LAMPU PERSIMPANGAN A	59
4.3 INTERVAL NYALA LAMPU PERSIMPANGAN B	60
5.1 TABEL KEBENARAN DEKODER 74LS139	68
5.2 PERENCANAAN DEKODER ROM	70
5.3 PERENCANAAN DECODER RAM	70
5.4 ALAMAT PORT 8255	71

KATA PENGANTAR

Dengan bermacam-macam kesulitan, hambatan-hambatan dan usaha yang tak kenal lelah, akhirnya berkat pertolongan dan rahmat Allah s.w.t, Tugas Akhir ini dapat diselesaikan yang merupakan syarat utama untuk meraih gelar sarjana pada Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember (ITS).

Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

- Bapak DR. Ir. Abdullah Alkaff selaku dosen wali dan dosen pembimbing yang telah banyak meluangkan waktu dan membantu di dalam menyelesaikan tugas akhir ini.
- Bapak DR. Ir. Mochammad Rameli selaku koordinator bidang studi teknik sistem pengaturan.
- Bapak Ir. Katjuk Astrowulan MSEE selaku ketua jurusan Teknik Elektro Fakultas teknologi Industri - ITS.
- Rekan-rekan mahasiswa dan juga semua pihak yang telah membantu, memberikan saran dan dukungan kepada penulis di dalam menyelesaikan tugas akhir ini.

Sesungguhnya hasil Tugas Akhir ini masih jauh dari sempurna, oleh karena terbatasnya kemampuan dan pengetahuan penulis. Untuk itu saran dan kritik pembaca sangat penulis harapkan.

Akhir kata, besar harapan penulis agar hasil Tugas Akhir ini dapat bermanfaat bagi semua pihak yang membutuhkannya.

Surabaya, Juli 1992

Penulis

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan yang pesat di bidang ekonomi, teknologi dan jumlah penduduk mengakibatkan jumlah sarana dan prasarana yang pesat pula. Tetapi umumnya terjadi ketidakseimbangan antara pertumbuhan sarana dan prasarana transportasi sehingga terjadi peningkatan volume lalu lintas yang tinggi. Sehingga kondisi lalu lintas akan semakin padat yang selanjutnya dapat menimbulkan kemacetan atau bahkan kecelakaan. Penyebab lain terjadi kemacetan lalu lintas ini adalah adanya suatu kecenderungan di kalangan masyarakat tertentu, bahwa mereka lebih senang menggunakan kendaraan pribadi dari pada kendaraan umum.

Permasalahan lalu lintas ini umumnya terjadi di persimpangan jalan. Untuk mengatasi permasalahan lalu lintas di persimpangan tersebut sudah dipasang alat pengatur arus lalu lintas yang dinamakan lampu lalu lintas atau traffic light. Namun lampu lalu lintas yang ada tersebut tidak saling mempengaruhi (tidak terkoordinasi). Sehingga bila ada beberapa persimpangan jalan yang berdekatan laju kendaraan akan banyak mengalami hambatan.

Berhubung dengan hal di atas, ada kemungkinan suatu kendaraan yang telah berhenti di salah satu persimpangan jalan (karena lampu merah menyala) harus berhenti lagi dipersimpangan jalan berikutnya (karena lampu merah juga sedang menyala).

Masalah di atas dapat diatasi dengan menerapkan mikroprosesor untuk mengatur nyala lampu lalu lintas pada daerah tersebut. Hal ini mengingat bahwa mikroprosesor dapat diprogram sesuai dengan karakteristik dinamik dari lalu lintas pada persimpangan-persimpangan di atas.

1.2 Permasalahan

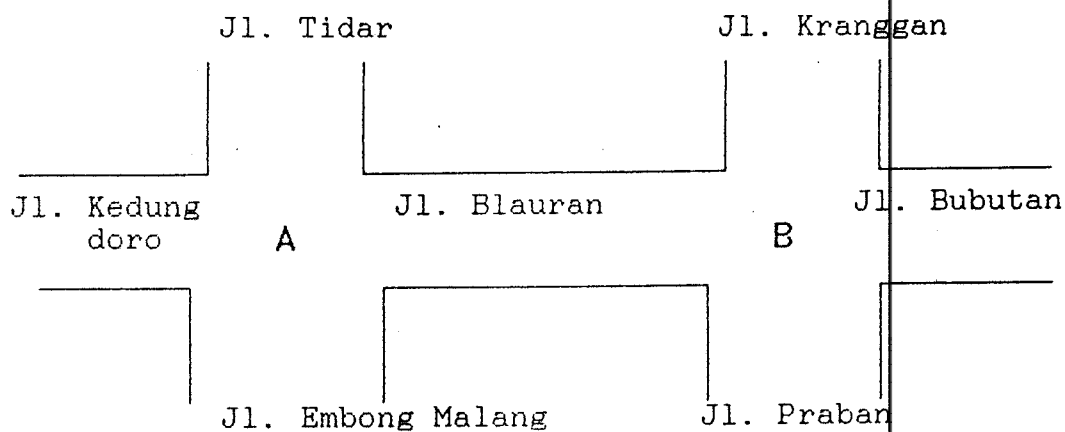
Masalah-masalah di atas harus dapat diatasi sehingga tercipta arus lalu lintas yang lebih lancar. Untuk mengatasi kemacetan lalu lintas pada beberapa persimpangan jalan yang berurutan dan berdekatan, perlu dirancang suatu sistem lampu lalu lintas yang terkoordinasi satu dengan yang lainnya. Pada Tugas Akhir ini akan digunakan sistem progresip yaitu sistem koordinasi beberapa persimpangan yang panjang siklus yang sama, namun phase merah dan hijau dapat bervariasi sesuai dengan kondisi masing-masing. Pengaturan waktu salah satu lampu relatif terhadap lampu berikutnya (yang selanjutnya dinamakan offset), disusun sedemikian rupa sehingga diperoleh gerakan lalu lintas yang kontinyu. Sistem yang dimaksud adalah sistem mikroprosesor yang akan mempunyai keluwesan terhadap kondisi

kepadatan kendaraan pada persimpangan. Yaitu dengan cara memprogram sistem mikroprosesor sesuai dengan karakteristik dinamik dari lalu lintas pada masing-masing persimpangan.

1.3 Batas Permasalahan

Untuk menyelesaikan Tugas Akhir ini, pertama-tama harus ditentukan adanya persimpangan yang berurutan dan berdekatan, yang masing-masing persimpangan terdapat lampu lalu lintasnya. Penulis menentukan lokasi penelitiannya di jantung kota Surabaya, yaitu persimpangan jalan Kedungdoro dengan jalan Embong Malang dan persimpangan jalan Blauran dengan jalan Kranggan. Sketsa lokasi penelitian ini diperlihatkan pada gambar 1.1

Jalan Blauran, pada ruas antara titik A dan titik B mempunyai arus lalu lintas satu arah. Sehingga kendaraan yang berasal dari persimpangan A akan langsung menuju persimpangan B.



GAMBAR 1.1 SKEMA LOKASI PENELITIAN

Sedang mengenai perangkat keras yang akan dibuat sebagai inti dari Tugas Akhir ini adalah berupa suatu model lampu lalu lintas. Yaitu dibuat suatu model jalan dengan persimpangannya dengan lampu LED yang berwarna merah, kuning dan hijau.

1.4 Sistematika Pembahasan

Setelah pendahuluan disajikan dalam BAB I ini, akan dijelaskan :

- BAB II : TEORI DASAR SISTEM LALU LINTAS akan diuraikan tentang penentuan lamanya interval hijau, kuning, merah menyalah. Kemudian dilanjutkan pengkoordinasian antar persimpangan.
- BAB III : SISTEM MIKROPROSESSOR, akan dibahas mengenai pembentukan sistem mikroprosesor yang meliputi hardwarnya yaitu mikroprosesor 8088, 3-sistem bus, memory serta komponen utama lainnya.
- BAB IV : PENGAMBILAN DATA DAN PERHITUNGANNYA akan diuraikan mengenai metode pengambilan data volume lalu lintas, kecepatan kendaraan, serta mengenai data lainnya. Kemudian akan dianalisa data tersebut untuk menghitung interval hijau, kuning, merah pada masing-masing persimpangan serta offset dari antar persimpangan.
- BAB V : PERENCANAAN DAN PEMBUATAN ALAT, akan dibahas

mengenai hardware dan softwarena.

- BAB VI : KESIMPULAN, akan dibahas mengenai kesimpulan-kesimpulan yang didapat setelah mengerjakan Tugas Akhir.

BAB II

TEORI DASAR SISTEM LALU LINTAS

2.1 Pendahuluan

Beberapa istilah yang sering digunakan pada sistem lalu lintas di antaranya ¹ :

- a. Kapasitas yaitu jumlah maksimum kendaraan yang melewati suatu ruas jalan atau jalur dalam satu atau dua arah selama periode normal.
- b. Kecepatan yaitu tingkat pergerakan lalu lintas kendaraan atau komponen lalu lintas lainnya yang dinyatakan dalam jarak per waktu.
- c. Kepadatan yaitu jumlah kendaraan yang menggunakan suatu ruas jalan atau jalur tertentu. Yang dinyatakan dalam kendaraan per panjang.
- d. Cycle time (cycle length) yaitu waktu yang dibutuhkan untuk nyala lampu satu urutan secara lengkap. Dinyatakan dalam detik.
- e. Traffic phase yaitu sebagian dari cycle yang menetapkan waktu dari tiap fhasenya.
- f. Interval yaitu sebagian dari traffic phase yang menentukan lamanya sebuah lampu menyala.

¹). HAPPY BUDHIATI, Studi Dua Persimpangan Di Jalan Kusuma Bangsa, tugas akhir Sipil ITS 1987, hal 19.

- g. Interval kuning sering disebut interval peringatan yaitu waktu nyala lampu kuning yang mengikuti interval hijau sebelum interval merah.
- h. Delay yaitu waktu yang hilang pada saat lalu lintas tertahan atau terhambat oleh beberapa elemen. Di mana pengemudi tidak dapat menguasainya.

2.2 Penentuan Lama Waktu Nyala Lampu

Pada umumnya, warna lampu yang dipakai untuk lampu lalu lintas (dari bawah ke atas atau dari kiri ke kanan) adalah HIJAU-KUNING-MERAH. Warna merah berarti kendaraan dilarang masuk persimpangan, warna kuning berarti peringatan bagi pengemudi bahwa lampu hijau akan segera berganti merah dan warna hijau berarti kendaraan diperbolehkan berjalan menuju persimpangan.

2.2.1 Penentuan interval kuning

Dalam sistem lalu lintas diperlukan interval kuning ini karena :

- a. Mengisyaratkan kepada pengemudi bahwa interval hijau telah berakhir dan supaya berhenti dengan hati-hati, karena lampu merah segera akan menyala.
- b. Mempersilakan kendaraan yang tepat atau sebelum lampu kuning menyala, menyeberang persimpangan dan selama lampu kuning menyala kendaraan-kendaraan ini telah sampai di phase seberangnya. Sehingga tidak mengganggu kendaraan-kendaraan yang akan berjalan dari phase yang lainnya.

Dalam hal ini, interval kuning sering disebut interval pembersihan artinya persimpangan dibersihkan terhadap kendaraan.

Interval lampu kuning ini diberikan oleh rumus ²:

$$t_a = t_d + \frac{U}{2a^2} + \frac{W+L}{U} \quad 2.1$$

dengan : t_a = interval waktu kuning (detik)

t_d = waktu reaksi dari pengemudi (detik)

U = kecepatan rata-rata (meter per detik)

W = lebar persimpangan jalan (meter)

L = panjang kendaraan (meter)

a = perlambatan kendaraan (meter per detik ²)

2.2.2 Penentuan interval hijau

Interval hijau dari kendaraan yang bergerak melitasi persimpangan dapat ditentukan dari panjang siklus dikurangi total waktu yang hilang terjadi pada kendaraan pertama melewati persimpangan, sehingga setiap fasenya waktu yang hilang adalah ³ :

$$K = k_1 + k_2 \quad 2.2$$

dimana : K = total waktu yang hilang (detik)

k_1 = waktu yang hilang saat mulai berjalan
(detik)

²). MARTIN WOHL AND BRIAN V. MARTIN, Traffic System analysis, MCGRAW-HILL BOOK COMPANY, 1967, hal 447

³). Ibit, hal. 450

k_2 = waktu yang hilang saat menunggu kendaraan terakhir melintasi persimpangan (detik)

Harga k_1 dapat diambil secara pendekatan yaitu sebesar 3,7 detik untuk setiap phasanya tetapi k_2 tergantung pada kecepatan kendaraan, lebar persimpangan dan panjang kendaraan. Secara umum, total interval hijau yang efektif adalah⁴ :

$$G_E = C - \sum_i^p K_i \quad 2.3$$

dengan : G_E = total interval hijau efektif (detik)

C = panjang siklus

K_i = total waktu yang hilang pada phase ke i (detik)

p = jumlah phase

Total interval hijau yang efektif ini dapat dihasilkan dengan merencanakan volume lalu lintas setiap phasanya. Untuk distribusi G_E yang proporsional, phase hijau total (termasuk interval kuning) dapat diperoleh dari rumus⁵ :

$$G_i = \frac{Dv_i}{Dv} G_E + K_i \quad 2.4$$

dengan : G_i = panjang waktu nyala hijau dan kuning pada phase ke i .

⁴. Ibid, hal 451

⁵. Ibid, hal 451

Dv_i = volume total kendaraan pada phase ke i

Dv = jumlah total volume kendaraan pada semua phase

Setelah waktu nyala lampu kuning dan hijau dapat ditentukan, maka nyala lampu merah dapat ditentukan yaitu sama dengan panjang siklus dikurangi dengan jumlah waktu nyala lampu kuning dan hijau.

2.3 Koordinasi Antar Persimpangan

Pada umumnya, setiap persimpangan utama dikontrol dengan lampu lalu lintas terutama pada pertemuan yang volume kendaraannya sangat tinggi. Sehingga kendaraan-kendaraan yang akan berubah arah perjalanan akan mengalami kesukaran, kecuali ada sinyal lalu lintas yang mengatur. Terkadang jarak antar persimpangan ini sangat berdekatan letaknya, sehingga kita dapat menganggap kendaraan-kendaraan yang melewati jalan antar persimpangan tersebut merupakan suatu kelompok kendaraan.

Untuk mendapatkan arus lalu lintas yang lancar, tidak cukup hanya dengan pengaturan setiap persimpangan, arus lalu lintas yang lancar dan menerus belum diperoleh. Oleh karena itu, adalah sangat penting untuk mengkoordinasikan lampu lalu lintas yang mempunyai jarak yang relatif dekat, sehingga dapat mengurangi delay.

Beberapa kegunaan pengaturan dari pada pengkoordinasi antar persimpangan antara lain :

1. meningkatkan kapasitas

2. mengurangi delay dan kemacetan
3. mengurangi dan mencegah kecelakaan
4. meningkatkan keamanan pemakai jalan
5. meningkatkan perawatan jalan

Pada daerah perkotaan di mana terdapat banyak persimpangan dan arus lalu lintasnya berurutan, maka dapat diterapkan beberapa koordinasi lampu lalu lintas yaitu ⁶ :

- a. Sistem simultan : Pada sistem ini semua lampu lalu lintas sepanjang jalan akan menyala dengan warna yang sama dalam waktu yang bersamaan. Di samping itu panjang siklusnya juga sama. Sistem ini sangat cocok, untuk jarak antar persimpangan yang sangat dekat serta pada kecepatan yang tinggi.
- b. Sistem bergantian (alternate sistem) : Pada sistem ini satu group (rangkai) lampu lalu lintas akan menyala merah, yang berikutnya menyala hijau dan seterusnya, sepanjang jalan tersebut. Sistem ini dapat menjadikan gerakan lalu lintas berjalan secara kontinyu. sistem ini efektif untuk dua jalan utama yang berpotongan dengan split 50-50 dan dengan kecepatan rencana $U = 2S/C$.
- c. Sistem progresip (progresive sistem) : Sistem ini dapat dipergunakan untuk persimpangan jalan yang berurutan.

⁶). HAPPY BUDIHATI, Op. Cit, hal 40

Masing-masing persimpangan akan mempunyai panjang siklus yang sama, namun interval merah dan hijau dapat bervariasi sesuai dengan kondisi masing-masing. Pengaturan waktu salah satu lampu relatif terhadap lampu berikutnya (yang selanjutnya dinamakan offset), disusun sedemikian rupa sehingga diperoleh gerakan lalu lintas yang kontinyu. Dengan sistem ini memungkinkan untuk menyingkronisasi jalan dua arah yang mempunyai jarak blok dan kecepatan kendaraan yang berbeda.

2.3.1 Sistem sinyal progresip

Seperti yang telah dijelaskan di bagian awal dari sub bab ini, bahwa terdapat berbagai macam sinyalisasi lalu lintas di persimpangan jalan. Namun, sesuai dengan penelitan tugas akhir ini hanya akan dibahas sistem sinyal progresip saja.

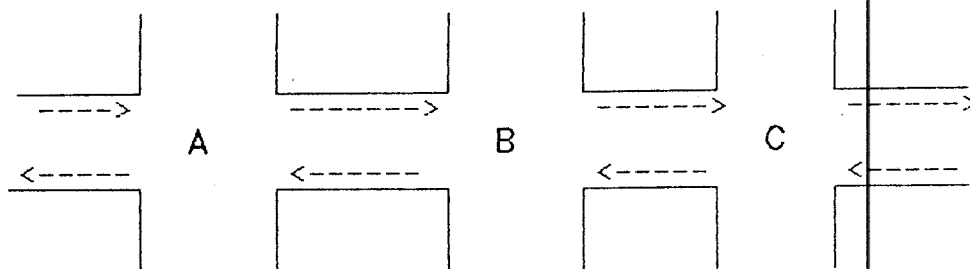
Pada kondisi lalu lintas yang padat, adanya lampu lalu lintas cenderung untuk membentuk kendaraan-kendaraan itu menjadi group-group yang diberi istilah platoon (peleton) dengan headway yang uniform. Bila pada suatu jalan mayor terdapat banyak persimpangan, maka dapat diusahakan agar peleton kendaraan tersebut dapat berjalan secara kontinyu setiap kali melewati persimpangan, yaitu dengan jalan menerapkan sistem sinyal progresip bagi lampu lalu lintas di daerah tersebut.

Keuntungan sistem ini bila dibandingkan dengan sistem individual adalah :

- waktu tunda (delay) dapat diperkecil sehingga perjalanan lebih singkat.
- jumlah pemberhentihan akan berkurang
- kecepatan kendaraan dapat ditingkatkan

Pada umumnya, sistem lalu lintas terkoordinasi semacam ini akan beroperasi dengan baik apabila laju kendaraan pada jalan mayor lebih dominan bila dibandingkan dengan jalan minor. Dan jumlah kendaraan yang masuk ke jalan mayor tersebut relatif kecil.

Gambar 2.2 di bawah memperlihatkan sketsa ruas jalan dengan tiga buah persimpangan yang berdekatan. Arus lalu lintas pada ruas jalan ini mempunyai dua arah yaitu A ke C dan dari C ke A.



GAMBAR 2.1

SKETSA RUAS JALAN DENGAN TIGA PERSIMPANGAN

Misalnya interval "hijau" dimulai dari persimpangan A, yang juga dijadikan sebagai referensi waktu. Namun harus diberikan jarak waktu terhadap waktu referensi. Jarak waktu ini dinamakan "offset" (T). Offset ini akan bergantung pada jarak antar persimpangan yang berurutan

(S) dan berbanding terbalik dengan kecepatan kendaraan (U) yang dirumuskan ⁷:

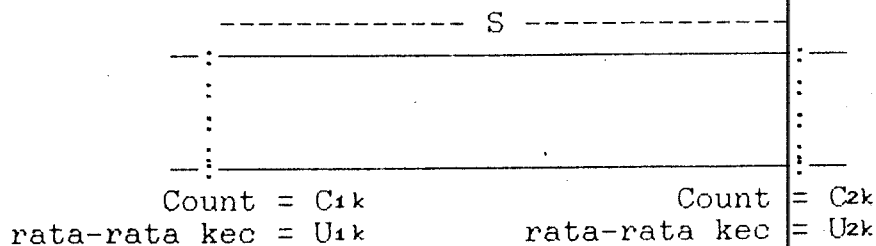
$$T = \frac{S}{U} \quad 2.5$$

di mana : T = offset (detik)

S = jarak antar persimpangan (meter)

U = kecepatan kendaraan (m/det)

Sedangkan kecepatan kendaraan pada suatu ruas jalan akan ditentukan oleh kepadatan kendaraan pada ruas jalan tersebut. Perhatikan gambar 2.2 di bawah ini, bagian ruas jalan sepanjang S meter pada saat $t = kT$ jumlah kendaraan $Y(k)$ yang ada pada bagian jalan tersebut. Untuk mengetahui $Y(k)$ ini digunakan dua set detektor, satu pada bagian masuk untuk menghitung kendaraan C_{1k} dan kecepatan U_{1k} yang masuk pada bagian ini dalam waktu interval T detik antara $kT - k+1T$.



GAMBAR 2.2 BAGIAN DARI RUAS JALAN

Sedangkan bagian yang lain menghitung jumlah kendaraan C_{2k} dan kecepatan U_{2k} yang meninggalkan bagian ruas jalan

⁷. THEODORE M. MATSON DKK, Traffic Engineering, McGRAW-HILL COMPANY, 1955, hal. 345

tersebut dalam interval yang sama t detik. Sehingga ada hubungan antara $Y(k+1)$ dan $Y(k)$ yaitu ⁸:

$$Y(k+1) = Y(k) + C_1k - C_2k + W(k) \quad 2.6$$

di mana $W(k)$ merupakan white noise yang disebabkan oleh kekurangtelitian detektor. Selanjutnya hubungan kecepatan rata-rata dan kepadatan lalu lintas dirumuskan sebagai berikut ⁹:

$$U(k) = b_k e^{-[Y(k)/a_k S]^2} + n(k) \quad 2.7$$

di mana kepadatan diberikan dengan $Y(k)/S$, a_k dan b_k adalah dua parameter yang diestimasi dan $n(k)$ adalah white noise.

⁸. MISCHA SCWARTZ, LEONARD SHAW, Signal Processing, Discrete Spectral Analysis, Detection and estimation, MC. Graw-Hill Inc 1975, hal 380.

⁹. Ibid. hal 381

BAB III

MIKROPROSESSOR 8088 BESERTA KOMPONEN PENUNJANG

3.1 Pendahuluan

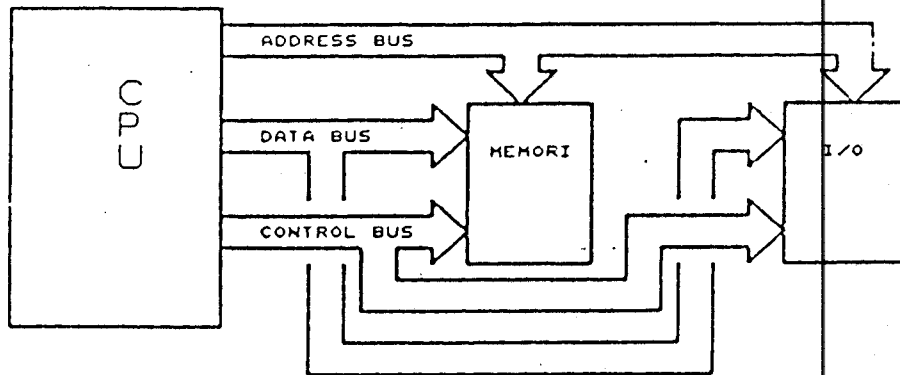
Pada bab ini akan dibahas mengenai sistem mikroprosesor 8088 sebagai unit pengendali beserta komponen-komponen penunjang dari alat yang dibuat, yaitu meliputi mikroprosesor 8088, arsitektur sistem 3 bus, EPROM, RAM, programable peripheral interface (PPI) 8255.

3.2 Sistem Mikroprosesor

Sebuah mikroprosesor adalah merupakan produk dari teknologi LSI yang menghasilkan implementasi CPU dalam satu serpih. Dan sebuah Sistem Mikroprosesor adalah implementasi dari sebuah mikroprosesor (CPU) atau lebih dengan ditunjang oleh beberapa unit fungsional yang lain. Sebuah Sistem Mikroprosesor yang sederhana terlihat dalam gambar 3.1 terdiri atas :

- Unit pengolah pusat (CPU, Central Processing Unit) dalam sebuah chip tunggal yang di dalamnya terdapat ALU (Arithmetic Logic Unit) yang melakukan perhitungan dan unit pengendali yang berfungsi untuk menyerempakkan operasi sistem serta register-register internal.
- Unit Memory yang berfungsi untuk menyimpan informasi.

- Unit Input-Output (I/O) yang berfungsi untuk komunikasi dengan dunia luar.



GAMBAR 3.1 SISTEM MIKROPROSESSOR

Antara unit-unit diatas direalisasikan dalam arsitektur sistem 3 bus yaitu:

- Sistem address bus
- Sistem data bus
- Sistem control bus

3.3 Arsitektur Sistem 3 Bus

Sistem bus didefinisikan sebagai kumpulan dari jalur-jalur sinyal yang mempunyai hubungan fungsi dengan mikroprosessornya. Dan dikelompokkan berdasarkan fungsinya di mana mempunyai fungsi sebagai penghubung dalam suatu sistem yang terlihat dalam gambar 3.2.

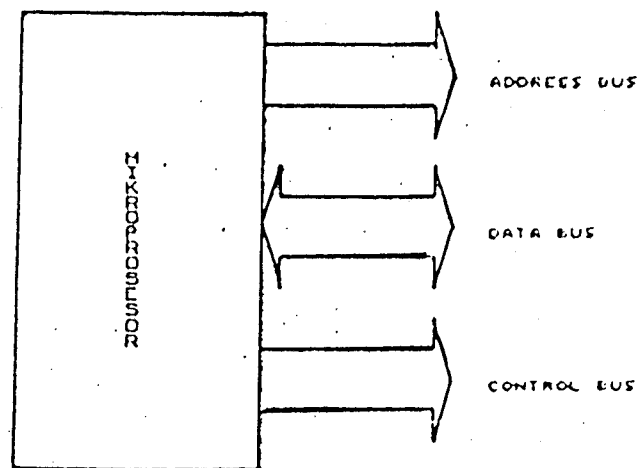
Sinyal-sinyal yang terdapat pada mikroprosesor 8 bit dapat digolongkan menjadi 3 buah system bus, yakni:

1. Sistem Address Bus.

2. Sistem Data Bus.

3. Sistem Kontrol Bus.

Karena terdiri dari 3 buah sistem bus maka disebut arsitektur sistem 3 bus.



GAMBAR 3.2 ARSITEKTUR SISTEM 3 BUS ¹⁰

3.3.1 Sistem address bus

Address bus digunakan untuk mentransmisikan alamat ke dalam lokasi memory atau piranti I/O yang memungkinkan pentransferan data. System address bus merupakan pin output dari mikroprosesor yang akan dihubungkan dengan bus data. Karena address bus merupakan pin output maka hanya

¹⁰, JAMES W. COFFRON, Practical Hardware Detail for
8080, 8085, Z80 dan 6800, Prentice-Hall Inc. 1981
hal 5

mempunyai satu arah saja (Undirectional). Pada mikro-prosesor 8 bit seperti 8088 bus alamat adalah 20 bit yang memungkinkan pengiriman sampai 1024 kbyte (2^{20}) lokasi alamat ekstern. Pada delapan bit address yang mempunyai orde rendah digunakan untuk menentukan lokasi I/O, maka untuk I/O hanya dapat mempunyai lokasi sebanyak $2^8 = 256$ lokasi. Bus alamat digunakan dalam hubungannya dengan bus data untuk menentukan sumber atau tujuan data yang dikirim pada data bus.

3.3.2 Sistem data bus

Sistem data bus merupakan sekumpulan jalur yang berfungsi untuk mengirim sinyal data menuju sistem memory dan ke sistem I/O, sebaliknya dari sistem memory atau I/O ke mikroprosesor. Jadi data bus bersifat dua arah (bidirectional).

Akan tetapi meskipun data bus bersifat bidirectional, dalam waktu yang bersamaan tidak dapat digunakan untuk mengirim dan menerima data. Jadi sinyal hanya dapat dikirim atau diterima pada satu arah saja pada satu saat.

CPU 8088 dilengkapi dengan delapan data biner yang diberi simbol D0-D7 di mana D0 adalah least significant bit sedang D7 adalah most significant bit. Data bus secara serentak membawa data valid dalam 8 bit atau data 1 byte dan oleh karena itu CPU 8088 disebut sebagai mikroprosesor 8 bit.

3.3.3 Sistem Kontrol Bus

Sistem kontrol bus pada mikroprosesor mempunyai empat buah sinyal yang mempunyai fungsi sama yaitu sebagai kontrol pada mikroprosesor. Keempat sinyal tersebut adalah

1. Memory Read (MEMR)
2. Memory Write (MEMW)
3. I/O Read (IOR)
4. I/O Write (IOW)

Kontrol bus ini merupakan sinyal output yang bersifat satu arah (Unidirectional). Sinyal MEMR aktif menunjukkan bahwa mikroprosesor sedang membaca data dari memory, bila sinyal MEMW aktif menunjukkan bahwa mikroprosesor sedang menulis data ke memory sedang untuk IOR dan IOW menyatakan bahwa mikroprosesor sedang menerima atau mengirim data ke peralatan luar. Dengan demikian fungsi sinyal kontrol adalah untuk menentukan hubungan kerja antara mikroprosesor dengan memory atau I/O baik kondisi Read atau Write.

3.4 Unit Memory

Memory merupakan tempat penyimpanan informasi yang dapat berupa data atau instruksi bagi suatu sistem mikroprosesor. Memory menurut fungsinya terdapat dari dua jenis, yaitu : ROM (Read Only Memory) dan RAM (Random Acces Memory).

3.4.1 Read only memory

Dalam sistem mikroprosesor diperlukan suatu memory

untuk menyimpan data yang sulit untuk dihapus (nonvolatile), di mana data yang terdapat pada memory tersebut tidak akan hilang bila power supply dari sistem dimatikan. Begitu isi memory ini ditentukan oleh proses rakitan, maka isinya tidak dapat lagi dirubah. Isinya dapat dibaca tetapi tidak dapat diisi dengan program baru. Karena ROM isinya tidak mudah dihapus (nonvolatile) maka ROM dipakai untuk menyimpan program-program utama. Ada beberapa jenis dari memory yang bersifat non-volatile yang digunakan dalam suatu sistem mikroprosesor, yaitu ¹¹:

1. Read Only Memory (ROM)
2. Programable Read Only Memory (PROM)
3. Erasable Programable Read Only Memory (EPROM)
4. Electricity Alternate Read Only Memory (EAROM)

Untuk PROM, EPROM dan EAROM data atau informasi dimasukkan dalam memory chip dengan suatu alat yang dapat memberikan pulsa tegangan yang cukup tinggi ke dalam sel-sel memory chip. Pada EPROM informasi yang telah disimpan masih dapat dihapus dengan memberikan sinar ultraviolet dengan intensitas tertentu pada window dari IC pada EPROM.

3.4.1.1 Sistematika pembacaan data dari ROM

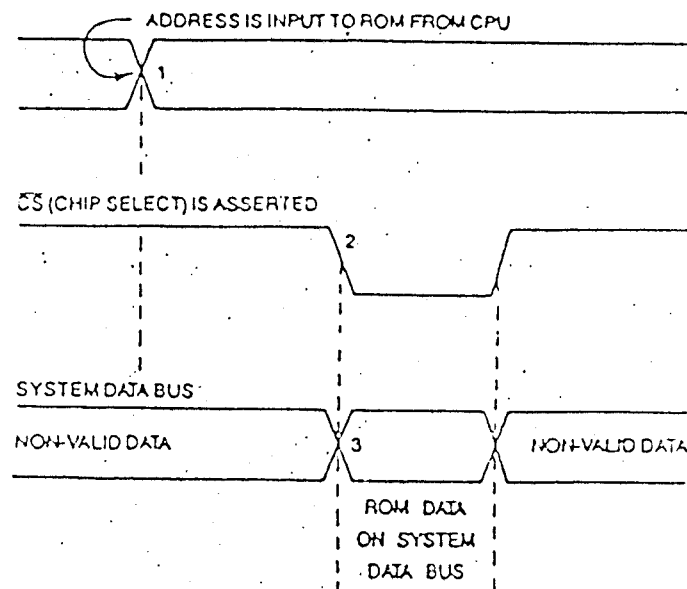
Gambar 3.3 menunjukkan diagram waktu untuk urutan

¹¹, JAMES W. COFFRON, Z80 Applications, Sybex, Inc. 1983, hal. 1

pembacaan data dari ROM.

Urutan-urutan proses pembacaan data dari ROM adalah sebagai berikut ¹² :

1. CPU memberikan input address kepada ROM sesuai dengan lokasi yang akan dibaca .
2. CPU menunggu untuk selang waktu tertentu berkisar antara 100-840 nano detik. Selang waktu ini diperlukan oleh rangkaian didalam ROM untuk mengkode address yang diterima dan data bus berada dalam keadaan siap menerima data.



GAMBAR 3.3 DIAGRAM WAKTU PEMBACAAN DATA DARI ROM ¹³

¹², Ibid. hal. 5

¹³, Ibid. hal. 30

3. Signal chip select diaktifkan, maka data akan dikirim dari memory ke data bus.
4. Sinyal chip select tidak diaktifkan untuk meniadakan data dari ROM pada data bus, sehingga alamat dan jalur data pada ROM pada keadaan impedansi tinggi.

3.4.2 Random access memory

RAM merupakan sistem memory yang menyimpan data hanya untuk sementara (volatile), karena bila power supply dimatikan maka data yang disimpan akan hilang. Dinamakan Random Access Memory karena lokasi manapun dapat dicapai secara langsung dengan menempatkan inputnya.

Terdapat dua jenis memory RAM yang digunakan untuk menyimpan data yaitu :

- a. Static RAM
- b. Dynamic RAM

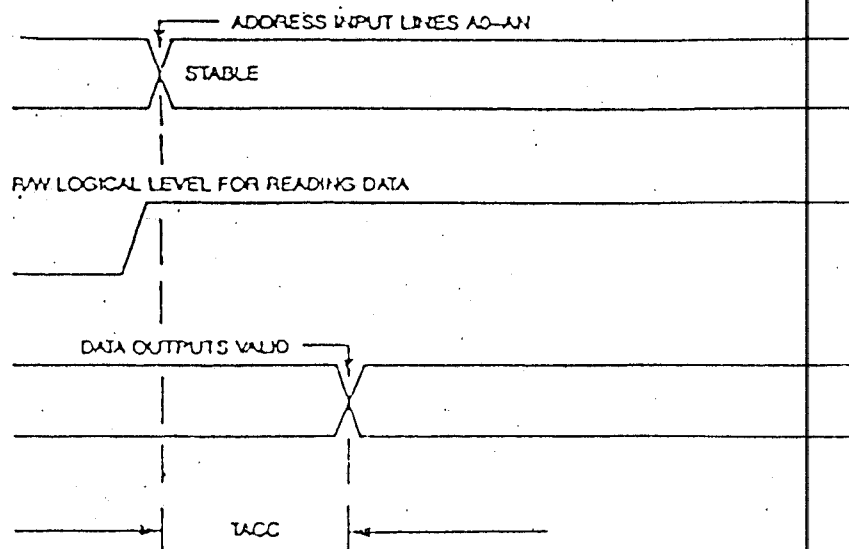
Static RAM dapat menyimpan data dan tetap stabil untuk selamanya selama daya tidak dimatikan, sedang Dynamic RAM harus diberi refresh cycle agar informasi dapat dipertahankan. Dalam perencanaan Tugas Akhir ini digunakan type RAM Static 6116 yang berkapasitas 2048 x 8 bit sebagai tempat untuk menyimpan data sementara.

3.4.2.1 Sistematika pembacaan data dari RAM

Urutan-urutan sinyal yang diperlukan untuk membaca

data dari RAM sebagai berikut ¹⁴:

1. Memory menerima address yang menentukan lokasi tertentu. Rangkaian decoder dalam RAM memilih elemen manakah yang harus diaktifkan.
2. Sinyal MEMR (Memory read) menjadi aktif dan memory langsung menerima sinyal ini.
3. Sistem menunggu dalam selang beberapa waktu tertentu (Read Access Time) sampai rangkaian di dalam RAM menjadi stabil.



GAMBAR 3.4 DIAGRAM PEMBACAAN DATA DARI RAM ¹⁵

¹⁴. Ibid. hal. 90

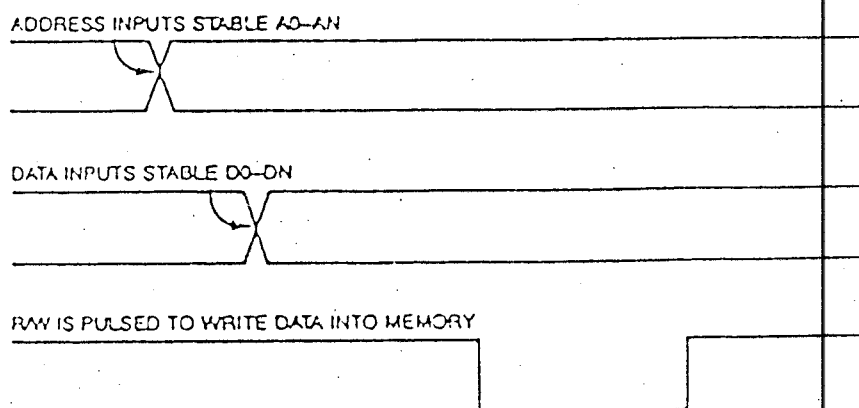
¹⁵. Ibid 90

4. Data akan dikirim dari memory ke data bus dan akan diterima oleh mikroprosesor. Jika mikroprosesor terlalu cepat mengaktifkan sinyal chip select (tidak menunggu Read Access Time), maka mikroprosesor akan mendapat data yang salah.

3.4.2.2. Sistematika penulisan data ke RAM

Diagram waktu penulisan data ke RAM, seperti ditunjukkan dalam gambar 3.5, di mana proses penulisan sebagai berikut ¹⁶:

1. Address dari data yang akan ditulis dimasukkan ke dalam address dari memory yang sesuai dengan lokasi yang ditentukan.



GAMBAR 3.5 DIAGRAM WAKTU PENULISAN DATA DARI RAM ¹⁷

¹⁶,. *ibid.* hal. 31

¹⁷,. *ibid.* hal. 31

2. Data yang akan diisi ke memory diletakkan ke data bus.
3. Mikroprosesor menunggu untuk selang waktu tertentu (Write Access Time) sampai rangkaian dalam RAM menjadi stabil.
4. Setelah Write Access Time, maka sinyal memory write diaktifkan sehingga data yang terdapat pada input RAM tertulis ke dalam elemen penyimpan pada RAM.

3.5 Mikroprosesor 8088

Mikroprosesor ini diproduksi dengan menggunakan teknologi High Performance Metal Oxide (HMOS) dan dalam rangkaiannya ekuivalen dengan 29000 transistor yang dikemas dalam bentuk IC 40 pin.

Mikroprosesor 8088 adalah hasil perubahan dari mikroprosesor 8086 yang pertama kali diperkenalkan oleh intel pada tahun 1979. Perbedaan antara mikroprosesor 8088 dengan mikroprosesor 8086 adalah arsitektur hardware dalamnya. Seperti 8086, mikroprosesor 8088 jumlah jalur bit internalnya adalah 16 bit, tetapi jumlah jalur bit data bus eksternalnya sebanyak 8 bit sedangkan mikroprosesor 8086 sebanyak 16 bit.

Keterangan untuk arsitektur internal dan untuk masing-masing pin, baik input maupun output beserta fungsinya, akan dijelaskan pada penjelasan-penjelasan berikut.

3.5.1 Arsitektur mikroprosesor 8088

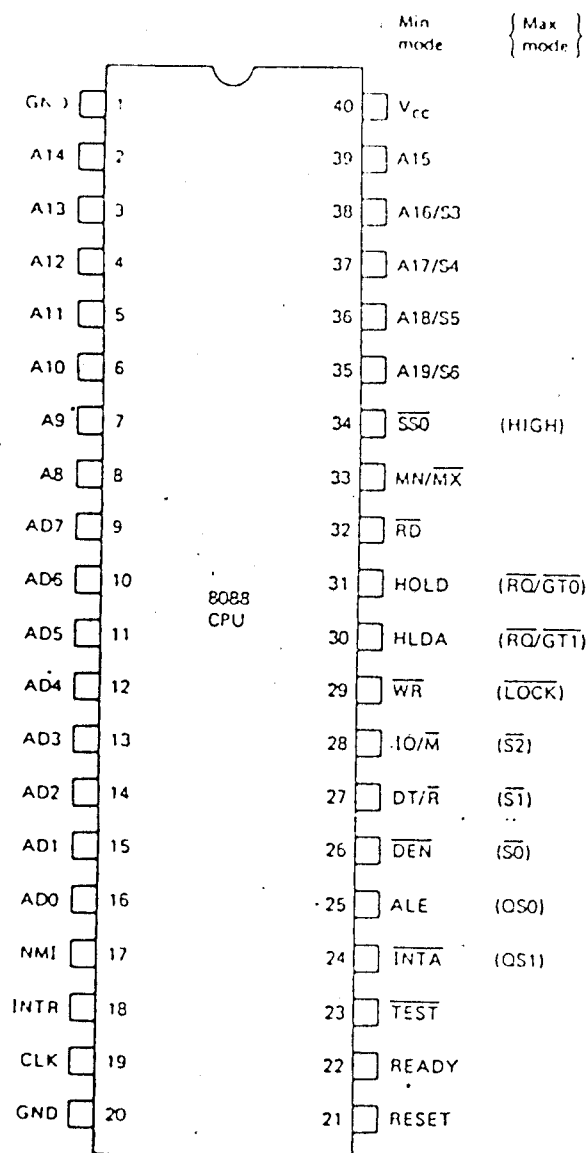
Konfigurasi pin diagram dari mikroprosesor 8088

dapat dilihat pada gambar 3.6. terlihat bahwa beberapa pin mempunyai fungsi ganda seperti address A0 - A7 yang dimultiplex dengan data D0 - D7.

Berikut di bawah ini diberikan penjelasan singkat dari pin-pin 8088 baik dalam mode minimum maupun dalam mode maksimum.

- A0-A7 berfungsi sebagai jalur address dan data yang bekerja secara multiplex. Jalur ini bekerja sebagai address (A0-A7) pada bus cycle yang pertama (T), dan pada bus cycle yang berikutnya (T2, T3 dan T4) jalur ini akan berfungsi sebagai jalur data.
- A8-A15 (Address Bus) berfungsi sebagai address bus.
- A16/S3-A19/S6 (Address/Status) berfungsi sebagai jalur address dan status yang bekerja secara multiplex. selama bus cycle yang pertama (T1), jalur ini berfungsi sebagai address (A16-A19) untuk operasi memory, selama operasi memory, jalur ini akan low. Untuk bus cycle berikutnya jalur ini berfungsi sebagai penunjuk status dari 8088.
- RD (Read) disebut sebagai Read Strobe, bilamana aktif berarti mikroprocessor sedang melaksanakan membaca isi memory atau melakukan pembacaan isi port. Jika IO/M berlogika 0 berarti sedang membaca memory, tetapi jika IO/M 1 berarti membaca port.
- READY berfungsi sebagai isyarat permintaan (acknowledge) dari port atau memory yang akan dibaca isinya oleh mikroprocessor, bahwa pembacaan dapat dilangsungkan. Signal ini aktif "high".

- RESET, digunakan untuk mereset mikroprocessor (input aktif low).



GAMBAR 3.6 KONFIGURASI PIN 8088 ¹⁸

- TEST, merupakan input aktif low, pin ini akan dianalisa

¹⁸, AVTAR SINGH dan W. A. TRIBBLE, The 8088 Microprocessor Programming, Intrefacing, Software Hardware And Application, Prentice Hall Inc. 1988 hal 22.

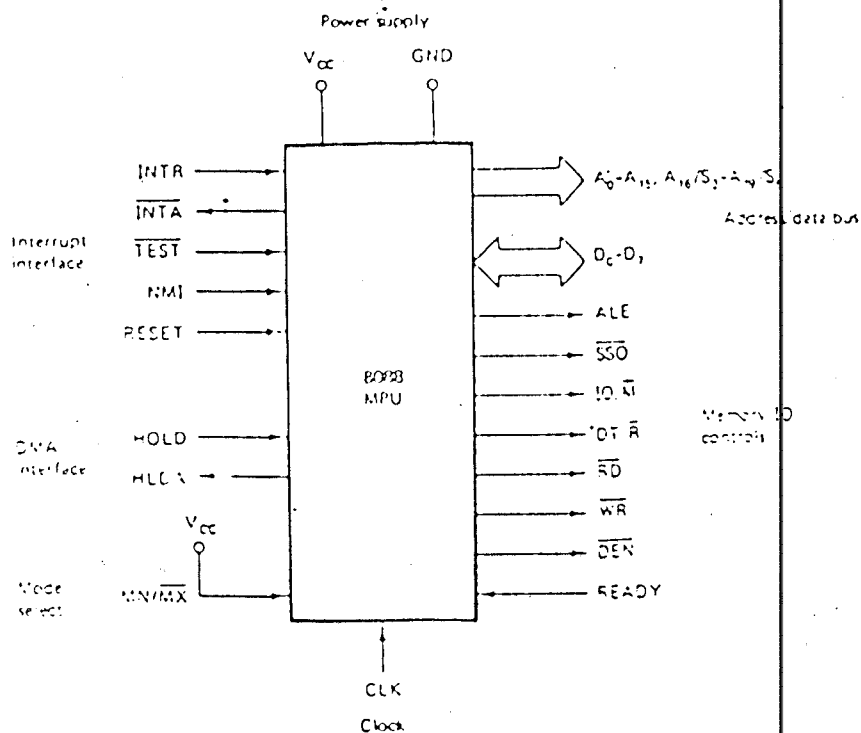
oleh intruksi 'wait for test' jika saat analisa kondisi pin ini low, maka instruksi akan dilanjutkan, tetapi jika kondisinya high, 8088 akan menunggu sampai pin TEST menjadi low.

- NMI, merupakan jenis interrupt yang berasal dari hardware dan interrupt ini tidak dapat dicegah sama sekali. Program induk dalam keadaan apapun akan ditinggalkan untuk melayani interrupt NMI.
- INTR, sinyal ini merupakan sinyal masukan yang berupa isyarat permintaan interup ke mikroprocessor 8088 dari perangkat keras atau di luar mikroprocessor 8088.
- CLK, clock merupakan input clock dari 8088 dan bus controller. Bentuk pulsa harus memiliki rise time minimal 10 nano detik.
- Vcc, merupakan pin power supply. Besar tegangan input yang diberikan pada pin ini adalah +5 Volt dengan toleransi 10 %.
- GND. dihubungkan dengan ground.
- MN/MX, merupakan pin yang menyatakan mode operasi mikroprocessor 8088. Jika input ini low, maka mikroprocessor akan beroperasi dengan mode maksimum. Sebaliknya jika input pin ini berlogika high, maka mikroprocessor 8088 beroperasi dengan mode minimum.

3.5.2 System mode minimum.

Mikroprocessor 8088 dapat dioperasikan pada sistem mode minimum dengan cara memberikan logika 1 pada pin

MN/MX. Pada operasi mode minimum, mikroprosesor 8088 mengendalikan sepenuhnya sinyal kontrol yang diperlukan pada operasi memory dan I/O.



GAMBAR 3.7 MODE MINIMUM 8088 MPU

Berikut ini akan dijelaskan pin-pin 8088 bila dioperasikan dalam mode minimum. Sedangkan blok diagram mode minimum 8088 MPU diberikan pada gambar 3.7 di atas.

- HOLD, merupakan input aktif high. Sinyal ini digunakan pada saat siklus DMA, yaitu jika peralatan eksternal akan mengambil alih kontrol system bus, maka peralatan eksternal harus memberikan logika 1 pada input HOLD.
- HLDA, merupakan output aktif high. sinyal ini digunakan pada siklus DMA, dan berfungsi untuk memberitahukan pada peralatan eksternal bahwa seluruh address bus, data bus

- dan control bus 8088 berada dalam keadaan high.
- WR, sinyal ini menyatakan bahwa mikroprocessor sedang melaksanakan operasi write memory atau I/O tergantung dari pin IO/M. WR akan aktif pada siklus T2, T3 dan T4 dari siklus bus.
 - IO/M, berfungsi untuk membedakan memory acces atau I/O acces. Bila kondisinya low, menyatakan I/O acces dan bila kondisinya high menyatakan acces.
 - INTA (Interrupt acknowledge), menyatakan bahwa mikroprocessor 8088 telah mendeteksi adanya permintaan interrump.

TABEL 3.1 OPERASI BUS CYCLE MODE MINIMUM

IO/M	DT/R	SSO	OPERASI
1	0	0	Interrupt acknowledge
1	0	1	Read I/O port
1	1	0	Write I/O port
1	1	1	Halt
0	0	0	Code access

- DT/R (Data Transmit/receive), berfungsi untuk mengatur arah pengiriman dan penerimaan data bus.
- DEN (Data Enable), merupakan output low. Logika 0 pada pin ini menunjukkan bahwa pin AD0-AD7 saat itu terdapat informasi data D0-D7.
- ALE (Address Latch Enable), merupakan sinyal output dari mikroprocessor 8088 yang digunakan untuk menyimpan address ke dalam address latch.

- SSO (Status line), merupakan status line yang bersama-sama dengan IO/M dan DT/R membentuk suatu sistem decoding dari status bus cycle. Pada tabel 3.1 dapat dilihat fungsi yang dibentuk oleh ketiga sinyal tersebut.

3.5.3 Sistem mode maksimum

Dalam sub bab ini akan dijelaskan pin-pin 8088 yang beroperasi dalam mode maksimum.

- S2, S1, S0 (Status)

Sinyal-sinyal ini digunakan 8088 bus controller untuk membangkitkan seluruh sinyal control untuk I/O dan memory access. Kombinasi ketiga sinyal ini akan digunakan oleh bus control untuk I/O dan memory access. Operasi yang ditunjukkan ketiga sinyal ini dapat dilihat pada tabel 3.2.

- RQ/GT0, RQ/GT1 (Request dan Grant)

Pin ini digunakan oleh pemakai bus lokal untuk meminta agar 8088 melepaskan sistem bus. Setiap pin bersifat bi-directional dan mempunyai fungsi yang sama dan RQ/GT0 mempunyai prioritas yang lebih tinggi dari RQ/GT1.

- Lock

Sinyal lock diaktifkan oleh intruksi tambahan "lock", akan tetapi aktif sampai intruksi berikutnya selesai dilaksanakan. Sinyal Lock ini akan digunakan untuk memberitahukan kepada pemakai bus lokal agar tidak ikut

memakai bus karena sedang digunakan.

TABEL 3.2 OPERASI BUS CYCLE MODE MAKSIMUM

S2	S1	S0	OPERASI
1	0	0	Interrupt acknowledge
1	0	1	Read I/O port
1	1	0	Write I/O port
1	1	1	Halt
0	0	0	Code access
0	0	1	Read memory

- QS1, QS0 (Queue Status)

menyatakan antrian instruksi pada mikroprosesor 8088 seperti yang dinyatakan oleh tabel 3.3.

- Pin 34 selalu aktif pada mode maksimum.

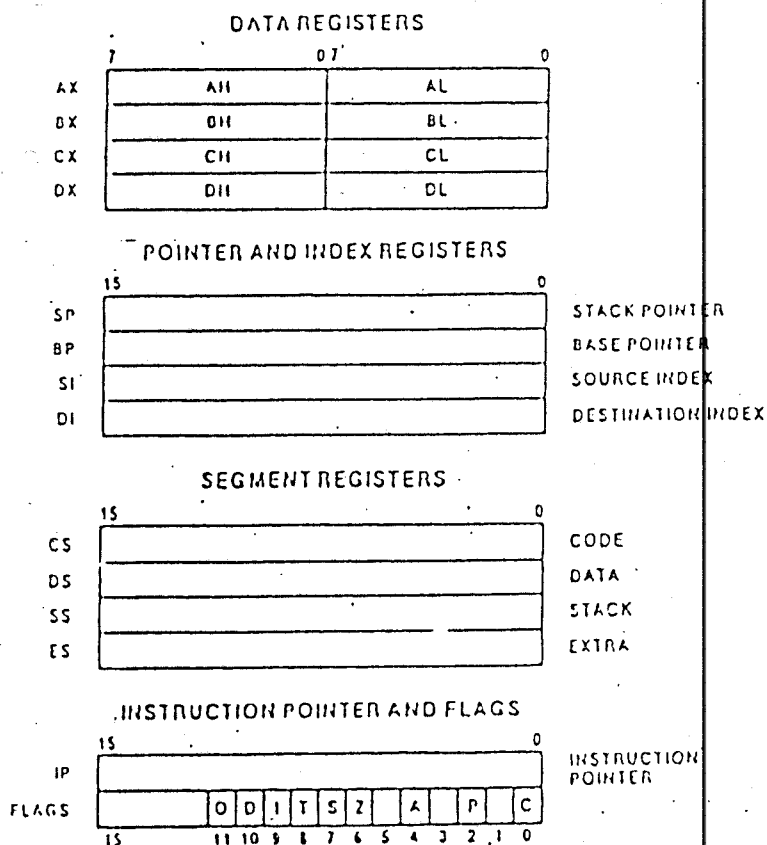
TABEL 3.3 STATUS INSTRUKSI

QS1	QS0	OPERASI
0	0	Tidak ada operasi
0	1	Op-Code byte pertama
1	0	Antrian kosong
1	1	Byte selanjutnya

3.5.4 Register mikroprosesor 8088

Mikroprosesor 8088 mempunyai 14 buah register 16 bit. Register-register ini dikelompokkan atas 5 buah group yaitu :

- Data register
- Instruction pointer register
- Pointer dan index register
- Segment register
- Flag register



GAMBAR 3.8 REGISTER MIKROPROSESSOR 8088

Data register terdiri dari 4 buah register yaitu Accumulator register (AX), Base register (BX), Count

register (CX), Data register (DX). Masing-masing register 16 bit ini dapat dioperasikan baik untuk 16 bit data operasi (word) atau dua 8-bit data operasi (byte). Jika register-register dioperasikan untuk operasi 8 bit, maka setiap register akan terbagi menjadi dua register yaitu : AH, AL, BH, BL, CH, CL, DH dan DL.

Register AX, BX, CX, dan DX memang disiapkan untuk register pengoperasi data, oleh karena itu register ini disebut sebagai register umum untuk mengoperasi data (General purpose register).

Instruction pointer register merupakan register 16 bit yang berfungsi mencatat offset address dari lokasi memory instruksi yang berikutnya pada harga base code segmen saat itu.

TABEL 3.4 OPERASI REGISTER MIKROPROSESSOR 8088

REGESTER	O P E R A S I
AX	Perkalian word, pembagian word, I/O
AL	Perkalian byte, pembagian byte, Translasi, Aritmatik desimal
AH	Perkalian byte, pembagian byte
BX	Translasi
CX	Operasi String
DX	Perkalian word, Pembagian word, I/O
SP	Operasi Stack
SI	Operasi String
DI	Operasi String

Pointer dan index register terdiri dari 4 buah register 16 bit yaitu : SP (Stack Pointer), BP (Base Pointer), SI (Source Index), DI (Destination Index).

Register-register ini hanya dapat dioperasikan untuk 16 bit. Fungsi dari register-register index dan pointer adalah untuk pembentukan memory efektif address (EA). Penggunaan register mikroprosesor 8088 ditunjukkan pada tabel 3.4.

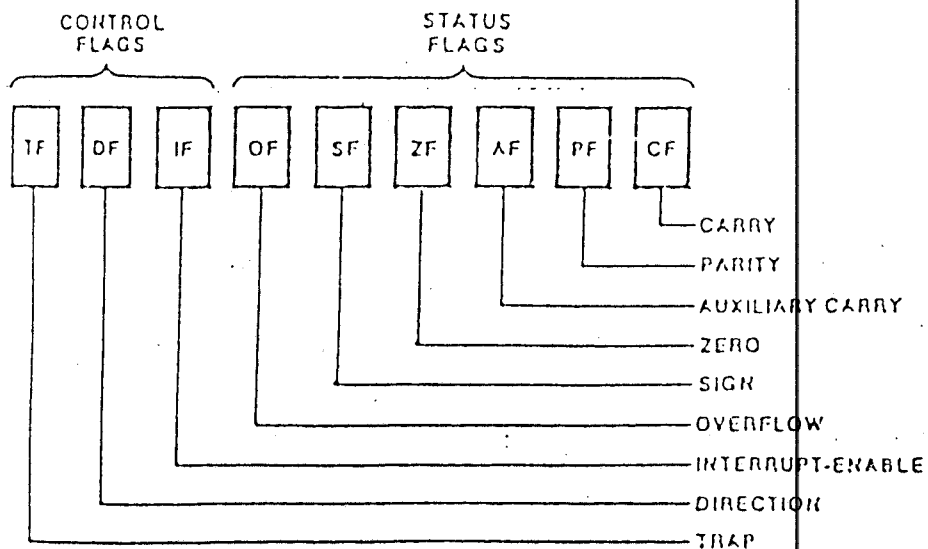
Mikroprosesor 8088 mencatat address untuk menunjukkan lokasi data di memory menggunakan segmen register ditambah dengan offset address atau selisih address.

Kemampuan addressing dari register 8088 adalah 20 bit yaitu 1.048.576 (1M) lokasi address, tetapi ada pula intruksi dari register 8088 yang hanya dilaksanakan untuk 16 bit address. Cara yang dilakukan untuk mendapatkan 20 bit address dari 16 bit address adalah dengan menambahkan 16 bit offset address ke 16 bit segmen address yang digeser ke kiri sebanyak 4 kali atau dengan kata lain segmen addressnya dikalikan dengan 16 dan baru ditambahkan dengan offset address.

Flag register merupakan register pembantu terhadap semua operasi aritmatika dan logika. Bentuk bantuannya berupa menyimpan tanda keadaan operasi atau akibat operasi aritmatik yang terjadi. Jumlah bit akibat operasi itu berjumlah 9 bit dan letak bit tersebut pada flag register sudah ditentukan secara pasti. Susunan bit dari flag register dapat dilihat pada gambar 2.4. Selanjutnya

dijelaskan fungsi masing-masing bit.

- CF (Carry Flag), bit akan diset apabila terjadi carry out atau borrow pada high-order bit sebagai hasil operasi aritmatik.
- OF (Overflow Flag), bit ini akan diset apabila terjadi overflow atau hasil aritmatik terlalu besar untuk daerah yang dituju.
- SF (Sign Flag), menunjukkan tanda dari bilangan hasil operasi aritmatik. bila diset maka tanda bilangan adalah negatif.



GAMBAR 3.9 FLAG REGISTER 8088 ²⁰

- PF (Parity Flag), flag ini akan diset apabila hasil operasi mempunyai jumlah bit yang berlogika satu adalah genap.
- ZF (Zero Flag), flag diset apabila hasil operasi aritmatik nol.
- AF (Auxiliary carry flag), bit ini akan diset apabila dalam operasi aritmatik terdapat carry out, borrow dari nibble perhitungan desimal.
- DF (Direction Flag), jika bit ini diset maka operasi string akan melakukan proses pengurangan secara otomatis dari high address. ke low address. Jika bit ini dibuat nol, maka pada operasi string akan dilakukan penambahan secara otomatis.
- IF (Interrupt Flag), apabila control flag diset, maka memungkinkan 8088 menerima external maskable interrupt.
- TF (Trap Flag), jika bit ini diset maka 8088 dalam mode single step. Setiap sebuah intruksi berakhir, maka interrupt akan dibangkitkan secara otomatis.

3.6 Programmable Peripheral Interface PPI 8255

PPI 8255 merupakan piranti I/O yang luas digunakan, karena mudah pengoperasiannya dan kompatibel dengan berbagai tipe mikroprocessor, yang berfungsi untuk menghubungkan komponen luar dengan sistem mikroprocessor. Dikatakan port I/O karena disinilah informasi keluar masuk dari/ke dalam mikrokomputer. Komponen I/O PPI 8255 merupakan paralel I/O dan dapat diprogram untuk mentransfer data

dalam berbagai kondisi dari I/O yang sederhana sampai dengan interrupt I/O yang kompleks.

PPI 8255 dibentuk dalam 40 pin dual in line package dan dikelompokkan dalam tiga port paralel 8 bit yaitu port A, port B, port C di mana port yang terakhir yaitu port C dapat dipisah dalam 4 bit orde rendah dan 4 bit orde tinggi. Fungsi masing-masing port dapat diprogram sebagai input/output dengan cara menulis control word dalam control register .

PPI 8255 dapat diklasifikasikan dalam 2 mode yaitu mode bit Set/Reset (BSR) dan mode I/O. Mode BSR digunakan untuk men set/reset bit dalam port, sedang mode I/O dibagi dalam 3 mode yaitu mode 0, mode 1 dan mode 2. Dalam mode 0 semua port berfungsi sebagai I/O yang sederhana, mode 1 adalah mode handshake dan mode 2 di mana peripheral dapat menggunakan port A sebagai port data bi-directional.

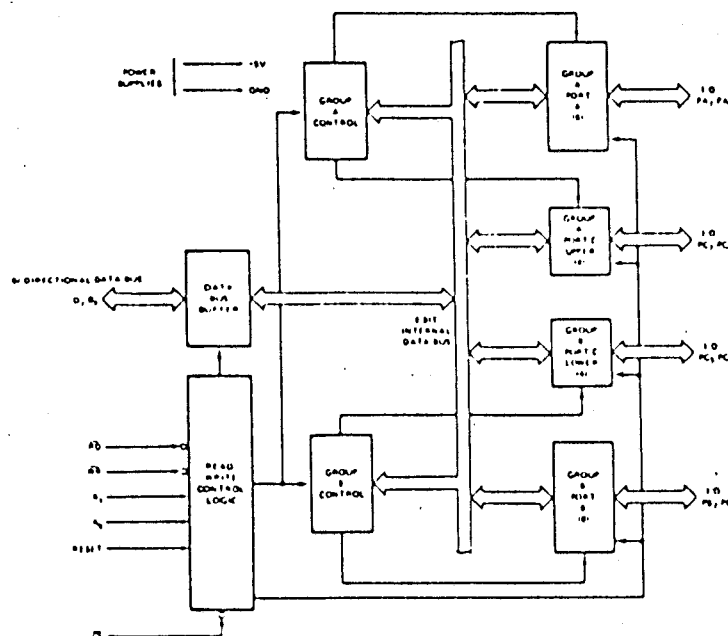
3.6.1 Blok diagram PPI 8255

Secara blok diagram PPI 8255A diperlihatkan dalam gambar 3.10 yang mempunyai dua kelompok 8 bit port A dan B dan kelompok 4 bit port C orde rendah dan orde tinggi, data bus buffer dan control logic.

Pada bagian control terdapat 6 jalur pengontrol sebagai berikut :

1. RD (Read) merupakan sinyal kontrol yang memungkinkan operasi baca, dimana jika sinyal berlogika 0, maka mikroprosesor membaca data pada I/O port.

2. WR (Write) merupakan sinyal kontrol yang memungkinkan operasi penulisan (write operation) dari I/O port yang dipilih.
3. RESET merupakan sinyal aktif high, sinyal yang akan mereset control register dan mensetting semua port ke dalam mode input.



GAMBAR 3.10 BLOK DIAGRAM PPI 8255 21

4. CS (Chip Select) adalah sinyal pemilih, digunakan untuk mengaktifkan chip PPI 8255, bila mendapat logika "0" mikroprosesor dapat mengirim data atau menerima data dari PPI 8255.

5. Address Input (A0-A1) Kombinasi dari kedua address input ini menentukan register mana dari PPI 8255 yang akan menerima atau mengirim data dari atau ke mikroprosesor.

TABEL 3.5 KEGUNAAN ADDRESS PIN A0, A1 DAN CS

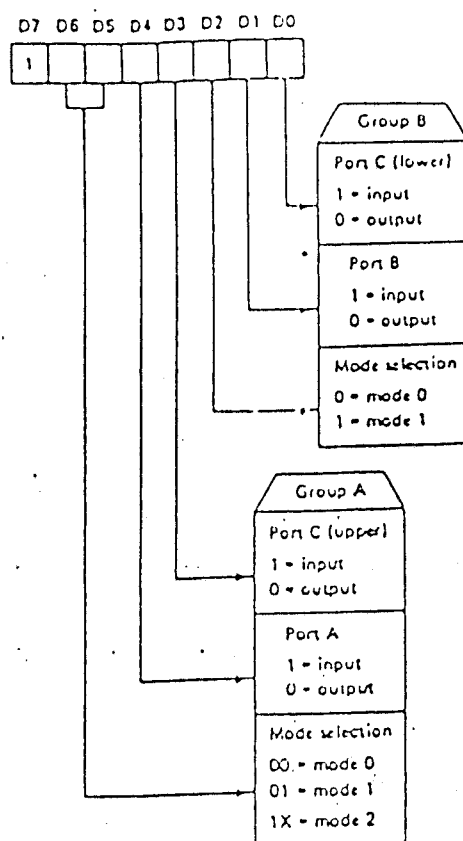
CS	A1	A0	YANG DIPILIH
0	0	0	PORT A
0	0	1	PORT B
0	1	0	PORT C
0	1	1	CONTROL REGISTER
1	x	x	8255 tidak dipilih

3.6.2 Pengoperasian PPI 8255A

Ada tiga macam basic mode operation dari PPI 8255 yang dapat dipilih melalui software (diprogram pada control word register) yaitu mode 0, mode 1 dan mode 2.

Pada gambar 3.11 diperlihatkan definisi bit dari control register, isi dari register ini disebut control register yang akan menentukan fungsi dari masing-masing port yaitu port A, port B dan port C .

Mode kerja dari PPI 8255 ditentukan melalui perangkat lunak. Untuk IC ini dilengkapi dengan kontrol register 8-bit. Ke dalam register-register ini dapat dituliskan logika 0 atau 1 untuk mengkonfigurasi masing-masing port sebagai input atau output dan menentukan mode operasi. Pada gambar terlihat bahwa bit D0 sampai D2 terkorrespondensi dengan group B. Bit D0 mengatur 4 bit bawah dari port C sebagai input atau output. Logika 1 pada D0 menjadikan input dan logika 0 menjadikannya output.



GAMBAR 3.11 FORMAT CONTROL WORD PPI 8255²²

Bit D1 mengatur port B (8-bit) sebagai port input atau sebagai port output. Logika 1 pada bit ini menjadikan input dan logika 0 menjadikannya output.

Bit D2 menentukan mode kerja port B (8-bit) dan 4 bit bawah port B. Terdapat dua operasi yang dapat dipilih, yaitu mode 0 dan mode 1. Logika 1 pada bit ini menjadikan mode 0 dan logika 1 menjadikan mode 1.

²², ibid hal 242.

Bit D3 dan D4 menentukan operasi kerja 4 bit atas port C dan port A. Bit-bit ini bekerja seperti bit D0 dan bit D1. Untuk bit D5 dan bit D6 berfungsi untuk memilih salah satu dari 3 mode operasi, yaitu mode 0, mode 1 dan mode 2.

Bit D7 adalah mode set flag. Bit ini harus berlogika 1 (aktif) bila mode operasi akan diubah. Pada perencanaan tugas akhir ini hanya digunakan 1 mode saja, yaitu mode 0.

Ketiga port dari PPI 8255 ini memberikan 24 jalur input atau output yang digunakan untuk menginterface komponen diluar mikroprosesor. Sekali piranti ini diprogram menurut yang dikehendaki, maka kita dapat mengeluarkan data dengan instruksi OUT atau instruksi IN untuk mengambil data .

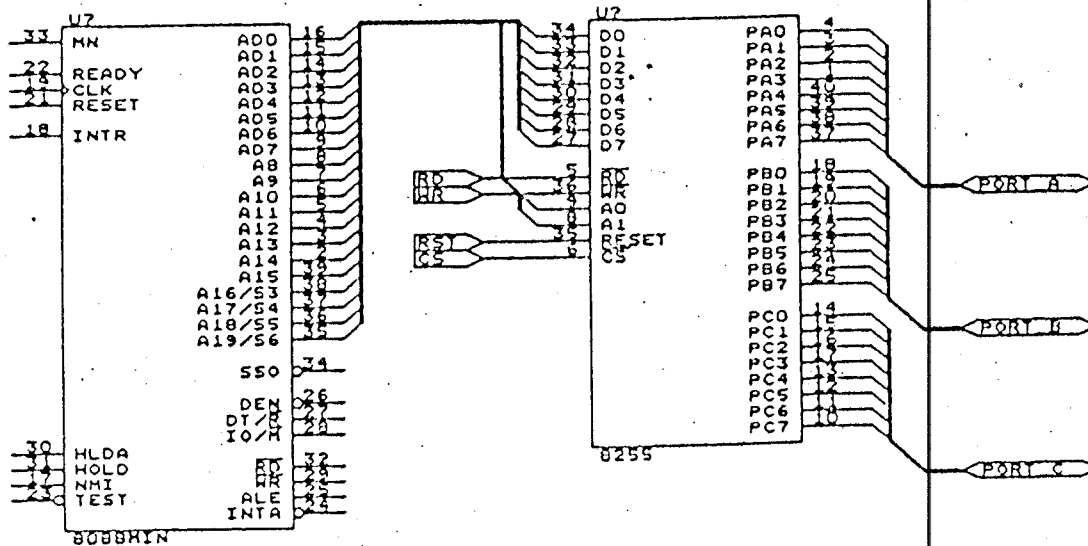
3.6.3 Menghubungkan PPI 8255 dengan mikroprosesor 8088

Telah diketahui bahwa PPI 8255 mempunyai dua buah address input, A0 dan A1 untuk memilih salah satu dari empat buah register yang dimilikinya. Dua address input ini dihubungkan ke A0 dan A1 dari mikroprosesor 8088. CS input line PPI 8255 digunakan untuk menempatkan PPI 8255 ke address I/O yang direncanakan.

Jadi input dari CS adalah output dari decoder yang mendecodekan alamat PPI tersebut, yaitu A0-A9.

Pin yang terakhir PPI 8255 yang perlu dihubungkan adalah RESET input. Input ini membutuhkan logika 1 untuk

bisa bekerja, maka dari mikroprosesor 8088 dapat langsung disambung. Hubungan lengkap PPI 8255 dengan mikroprosesor 8088 ditunjukkan pada gambar 3.12.



GAMBAR 3.12

HUBUNGAN LENGKAP PPI 8255 DENGAN MIKROPROSESSOR 8088

3.7 Address Decoding

Decoding berfungsi untuk mendekodekan saluran address CPU untuk mengaktifkan komponen memory maupun komponen I/O agar data yang akan dibaca atau ditulis pada piranti tersebut tidak kacau. Hal ini dapat dimungkinkan dengan suatu cara yang disebut dengan Address decoding, di mana sinyal-sinyal pada address bus didecode sedemikian rupa sehingga setiap kombinasi pada address akan menghasilkan satu sinyal pilih yang akan mengaktifkan salah satu IC. Untuk berkomunikasi dengan peralatan I/O maka diperlukan suatu cara agar mikroprosesor dapat memilih atau menentukan salah satu IC atau peralatan

tertentu saja dengan mengaktifkan CS (Chip Select).

Beberapa metode yang dipergunakan untuk mendekode suatu address yaitu Gating dan Decoding.

- GATING :

Gating merupakan cara yang paling sederhana untuk mendecodekan suatu kombinasi address dengan menggunakan gate-gate logika seperti AND, OR, dan NOT. Cara ini tidak banyak digunakan karena tidak efisien untuk sinyal pilih yang banyak.

- DECODING :

Cara yang paling mudah dan efisien untuk decoding adalah dengan menggunakan dekoder. Dengan menggunakan dekoder maka setiap n macam kombinasi address dapat diperoleh sebanyak 2^n sinyal pilih, sehingga dekoder ini cocok untuk sistem yang memerlukan sinyal pilih yang banyak .

Dalam perencanaan tugas akhir ini menggunakan dekoder 74LS138 untuk menentukan lokasi address EPROM, RAM, dan PPI 8255 sedang decoder 74LS139 dipakai untuk kontrol bus .

3.8 Buffering

Tiap masukan sebuah alat merupakan beban pada keluaran yang menggerakannya. Sebagian besar komponen menggerakkan mulai dari satu sampai dua puluh komponen lainnya. Setiap komponen harus diperiksa karakteristik penggerakan serta pembebanan dan keluarannya.

Bus mikroprosesor harus berhubungan dengan setiap serpih masukan dan keluaran periferal dan memory sistem.

Semua mikroprosesor MOS kurang mampu untuk menggerakkan keluarannya yang diperlukan dalam suatu sistem besar. Jika mikroprosesor dibebani melebihi fan out, akibatnya level tegangan pada pin-pin bus dapat turun sampai melebihi batas level marginnya yang mengakibatkan mikroprosesor memberikan informasi yang tidak benar, karena itu dipakai buffer atau penggerak untuk menaikkan daya penggerakan bus.

Misalnya jalur address pada mikroprosesor harus mendrive input address dari empat komponen memory dan dua komponen I/O, maka besarnya arus suplay untuk mendrive komponen memory dan komponen I/O harus sama dengan jumlah arus input dari keempat komponen memory ditambah dengan jumlah arus input dari kedua komponen I/O.

4.1 Pendahuluan

Sebelum merencanakan sistem pengatur lampu lalu lintas, terlebih dahulu dilakukan pengambilan data-data tentang volume dan kecepatan kendaraan pada daerah yang dijadikan obyek penelitian. Data-data ini nantinya digunakan untuk menentukan interval waktu penyalaan lampu lalu lintas di lokasi tersebut.

Untuk keperluan tugas akhir ini, dilakukan survey spot speed pada masing-masing ruas jalan di lokasi penelitian. Sedangkan untuk data volume kendaraan diambil data survey kendaraan yang dilakukan oleh laboratorium perhubungan, jurusan teknik sipil FTSP-ITS yang dilakukan pada hari sabtu tanggal 9 Desember 1989. Data-data penunjang lainnya diperoleh dari Biro Pusat Statistik Jawa Timur.

4.2 Survey Spot Speed

Pelaksanaan survey spot speed untuk tugas akhir ini dilaksanakan lima hari pada bulan mei 1992. Metode pelaksanaan spot speed yang digunakan adalah cara yang paling sederhana yaitu dengan prinsip pengukuran waktu yang digunakan untuk menempuh jarak tertentu. Metode ini dipakai

karena lebih murah dan menggunakan peralatan sederhana.

Survey spot speed dilakukan pada lima lokasi yaitu :
Jl.Kedungdoro, Jl.Embong Malang dan Jl.Tidar masuk menuju ke persimpangan A. Sedangkan Jl.Blauran, Jl.Praban sisi utara masuk ke persimpangan B.

4.3 Peramalan Volume Kendaraan

Karena data survey volume kendaraan yang ada adalah data pada tahun 1989 dari laboratorium perhubungan jurusan teknik sipil ITS, yang dapat dilihat pada gambar 4.1. Untuk penyesuaian data volume kendaraan ke tahun 1992 perlu diramalkan dengan menggunakan metode yang telah digunakan oleh Edi Napitulu dalam bukunya 'Studi Pembinaan Arus Lalu Lintas Daerah Wonokromo' tugas akhir jurusan teknik sipil ITS. Yaitu untuk mencari volume lalu lintas dan peramalannya pada daerah penelitian adalah dengan mengalikan volume lalu lintas dari data survey dengan faktor pertumbuhannya.

4.3.1 Faktor pertumbuhan kendaraan

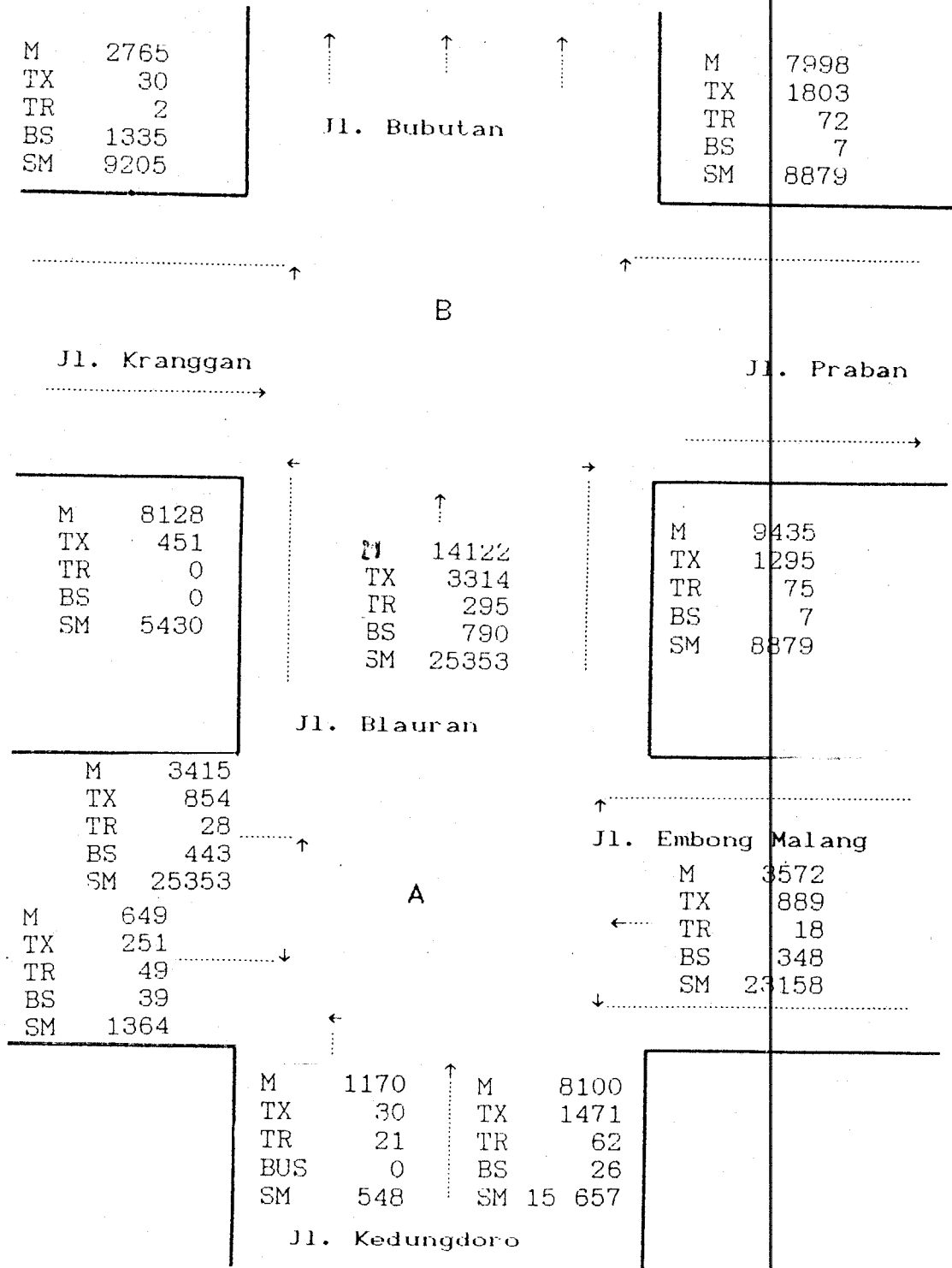
Sesuai dengan data yang sudah ada pada lampiran B maka dapat diperoleh persamaan linier regresi sebagai berikut :

- persamaan linier regresi untuk PDRB (product domestic regional brutto) per penduduk :

$$Y = 9943.76 X - 196726178.24$$

- persamaan linier regresi untuk jumlah penduduk :

$$Y = 16154 X - 29945844$$



GAMBAR 4.1

VOLUME KENDARAAN DI LOKASI PENELITIAN TAHUN 1989

- persamaan linier regresi untuk mobil :

$$Y = 9098 X - 18006215$$
- persamaan linier regresi untuk taxi :

$$Y = 134.3 X - 263594$$
- persamaan linier regresi untuk truk :

$$Y = 2487.8 X - 4898265$$
- persamaan linier regresi untuk bus :

$$Y = 120.4 X - 236706$$
- persamaan linier regresi untuk sepeda motor :

$$Y = 15502.4 X - 304491710$$

di mana X menyatakan tahun.

Selanjutnya untuk mencari besarnya faktor pertumbuhan kendaraan dari masing-masing jenis kendaraan digunakan perumusan sebagai berikut ²³⁾ :

- fp mobil = $f_{psur \text{ mobil}} \times f_{psur \text{ PDRB/penduduk}}$
- fp taxi = $f_{psur \text{ taxi}} \times f_{psur \text{ penduduk}}$
- fp truk = $f_{psur \text{ truk}} \times f_{psur \text{ PDRB/penduduk}}$
- fp bus = $f_{psur \text{ bus}} \times f_{psur \text{ penduduk}}$
- fp spdmtr = $f_{psur \text{ spdmtr}} \times f_{psur \text{ PDRB/penduduk}}$

di mana fp masing-masing kendaraan adalah faktor pertumbuhan masing-masing kendaraan di lokasi penelitian, dan f_{psur} adalah faktor pertumbuhan di Surabaya. Faktor pertumbuhan masing-masing kendaraan di Surabaya adalah jumlah kendaraan tahun 1992 dibagi dengan jumlah masing-masing

23). EDDY NAPITULU, Studi Pembenahan Arus Lalu Lintas Di Daerah Wonokromo, tugas akhir 1989, hal 60.

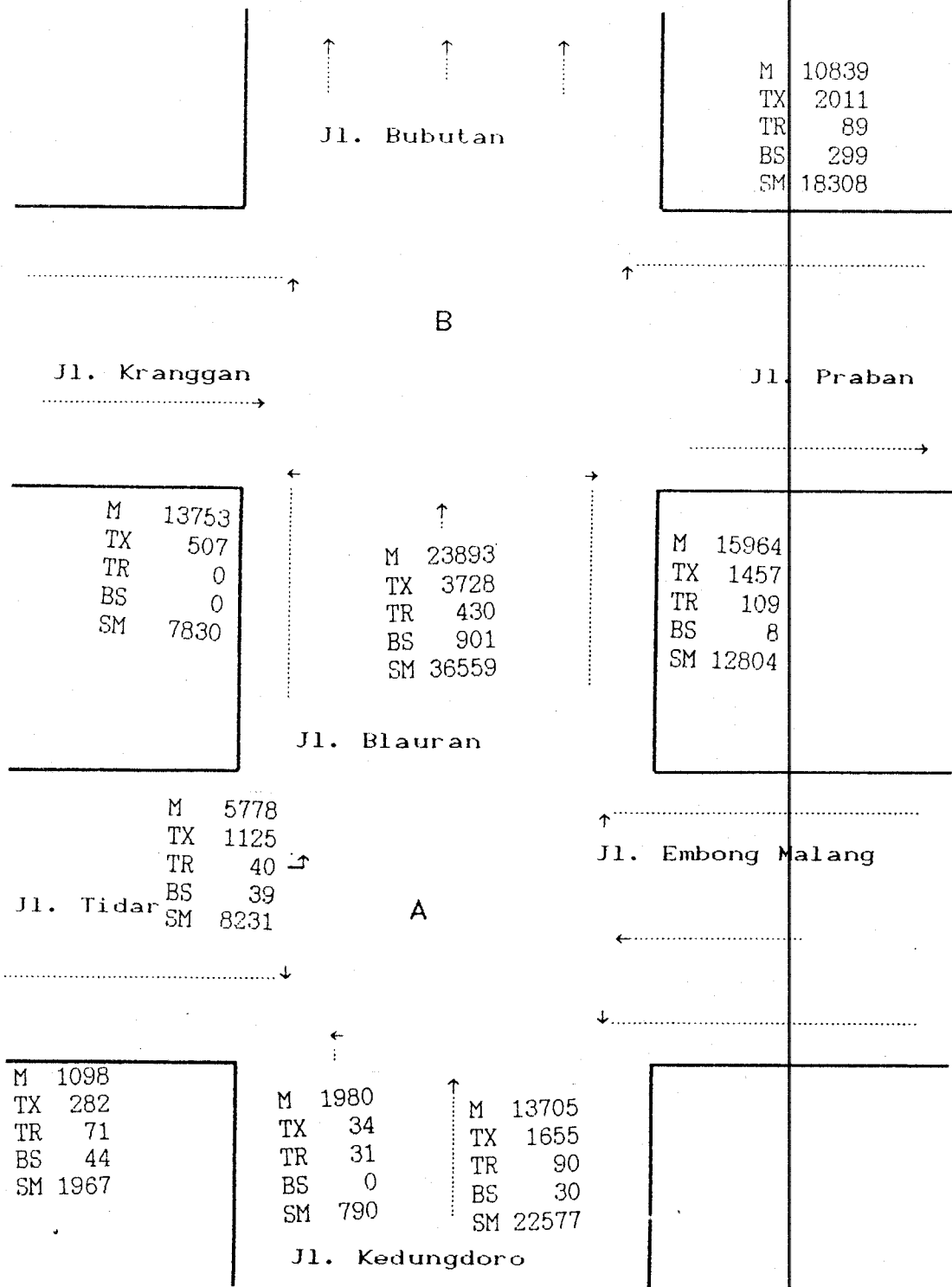
kendaraan tahun 1989. Faktor pertumbuhan PDRB/penduduk adalah PDRB/penduduk tahun 1992 di Surabaya dibagi dengan PDRB/penduduk tahun 1989. Faktor pertumbuhan penduduk di Surabaya adalah jumlah penduduk di Surabaya tahun 1992 dibagi dengan jumlah penduduk tahun 1989.

Dengan memasukan data-data yang ada diperoleh :

- fp mobil = 1.692
- fp taxi = 1.125
- fp truk = 1.456
- fp bus = 1.141
- fp spdmtr = 1.442

4.3.2 Penyesuaian volume kendaraan ke tahun 1992

Untuk menyesuaikan besarnya volume kendaraan ke tahun 1992 pada lokasi penelitian adalah dengan mengalikan volume kendaraan data tahun 1989 yang terlihat pada gambar 4.1 dengan faktor pertumbuhan dari masing-masing jenis kendaraan. Dan hasilnya seperti pada gambar 4.2.



GAMBAR 4.2

PENYESUAIAN VOLUME KENDARAAN KE TAHUN 1992

4.4 Perhitungan Ekuivalen Volume Kendaraan

Setelah mengetahui volume kendaraan pada masing-masing ruas jalan, selanjutnya dicari ekuivalen volume kendaraan ke satuan mobil penumpang (SMP) bagi semua jenis kendaraan yang menjadi obyek penelitian. Hal ini harus dilakukan karena rumus-rumus pada bab 2 hanya diperuntukkan pada kendaraan penumpang saja.

Menurut Laboratorium Perhubungan Jurusan Teknik Sipil ITS, faktor ekivalensi masing-masing kendaraan adalah :

TABEL 4.1 FAKTOR EKIVALENSI KENDARAAN

Jenis Kendaraan	Faktor Ekivalensi
Mobil + Taxi	1 SMP
Truk	3 SMP
Bus	2.5 SMP
Sepeda Motor	0.3 SMP

4.4.1 Persimpangan A

Lokasi penelitian pada persimpangan A dibagi menjadi tiga phase. Phase 1 meliputi Jl.Kedungdoro menuju ke Jl.Tidar dan ke Jl.Blauran, phase 2 meliputi Jl.Embong Malang menuju ke JL.Tidar dan phase 3 meliputi JL.Tidar menuju ke Jl.Blauran dan ke JL.Kedungdoro. Kendaraan yang berasal dari JL.Embong Malang menuju ke JL.Kedungdoro dan ke Jl.Blauran tidak tergantung nyala lampu.

Berdasarkan tabel 4.1, volume yang ditunjukkan gambar 4.2 dirubah ke volume satuan kendaraan penumpang

dengan cara mengalikan masing-masing kendaraan dengan faktor ekivalensinya.

- phase 1 :

- o volume dari Jl.Kedungdoro ke JL.Blauran,
 $(13705+1655) + 90 \times 3 + 30 \times 2.5 + 22577 \times 0.3$
 $= 22478 \text{ SMP}$

- o volume dari Jl.Kedungdoro ke JL.Tidar,
 $(1980+34) + 31 \times 3 + 790 \times 0,3 = 2344 \text{ SMP}$

jumlah total volume kendaraan pada phase 1 adalah

$$DV_1 = 22478 + 2344 = 24822 \text{ SMP.}$$

- phase 2 :

- o volume kendaraan dari Jl.Embong Malang ke Jl.Tidar,
 $DV_2 = (6044+1000) + 26 \times 3 + 397 \times 2.5 + 35894 \times 0.3$
 $= 18732 \text{ SMP.}$

- phase 3 :

- o volume kendaraan dari Jl.Tidar ke Jl.Kedungdoro
 $(1098+282) + 71 \times 3 + 44 \times 2.5 + 1967 \times 0.3$
 $= 2293 \text{ SMP}$

- o volume kendaraan dari Jl.Tidar ke Jl.blauran,
 $(5778+1125) + 40 \times 3 + 39 \times 2.5 + 8231 \times 0.3$
 $= 9590 \text{ SMP}$

Jumlah total volume kendaraan pada phase 3 adalah

$$DV_3 = 2293 + 9590 = 11883 \text{ SMP}$$

4.4.2 persimpangan B

Persimpangan B di bagi menjadi dua phase saja, yaitu phase 1 adalah volume kendaraan berasal dari Jl.Blauran

menuju ke Jl.Kranggan dan ke Jl.Bubutan. Sedangkan phase 2 adalah volume kendaraan berasal dari Jl.Praban menuju ke Jl.Bubutan atau dari JL.Kranggan menuju ke JL.Bubutan dan ke JL.Praban.

- phase 1 :

- o volume kendaraan berasal dari Jl.Blauran ke Jl.Kranggan,

$$(13753+507) + 783 \times 0.3 = 14495 \text{ SMP}$$

- o volume kendaraan dari Jl.Blauran ke Jl.Bubutan,

$$(23893+3728) + 430 \times 3 + 901 \times 2.5 + 36559 \times 0.3 \\ = 42132 \text{ SMP}$$

- o volume kendaraan dari Jl.Blauran ke Jl.Praban,

$$(15964+1457) + 109 \times 3 + 8 \times 2.5 + 12804 \times 0.3 \\ = 21609 \text{ SMP}$$

$$\text{Jumlah volume kendaraan phase 1 DV}^4 = 78236 \text{ SMP.}$$

- Phase 2 :

- o volume kendaraan dari Jl.Kranggan,

$$(4678 + 1502) + 44 \times 3 + 3 \times 2.5 + 13273 \times 0.3 \\ = 10304 \text{ SMP}$$

- o volume kendaraan dari Jl.Praban ke Jl.Bubutan,

$$(13533 + 2028) + 105 \times 3 + 305 \times 2.5 + 18308 \times 0.3 \\ = 22131 \text{ SMP}$$

karena volume kendaraan yang berasal dari Jl.Kranggan lebih kecil jika dibanding dengan Jl.Praban, maka dalam perhitungan interval lampu dipilih yang lebih besar yaitu yang bersasal dari Jl.Praban ke Jl.Bubutan. Sehingga jumlah kendaraan pada phase 2 DV^s = 22131 SMP. Hasil

ekivalensi keseluruhan mobil penumpang dapat dilihat pada gambar 4.3.

4.5 Perhitungan Interval Nyala Lampu

4.5.1 Persimpangan A

Berdasarkan gambar 4.3 di bawah, maka dapat dilakukan perhitungan tentang interval lampu untuk setiap phase pada masing-masing persimpangan. Dari perhitungan yang telah didapat volume kendaraan masing-masing phase yaitu :

- phase 1 $DV_1 = 24882$ SMP
- phase 2 $DV_2 = 18732$ SMP
- phase 3 $DV_3 = 11883$ SMP

sehingga jumlah kendaraan yang masuk pada persimpangan A adalah $DVA = 55437$ SMP. Sedangkan lebar jalan pada setiap phase adalah :

- phase 1 $W_1 = 14.5$ m
- phase 2 $W_2 = 12$ m
- phase 3 $W_3 = 12$ m

kecepatan rata-rata ruas jalan pada masing-masing phase :

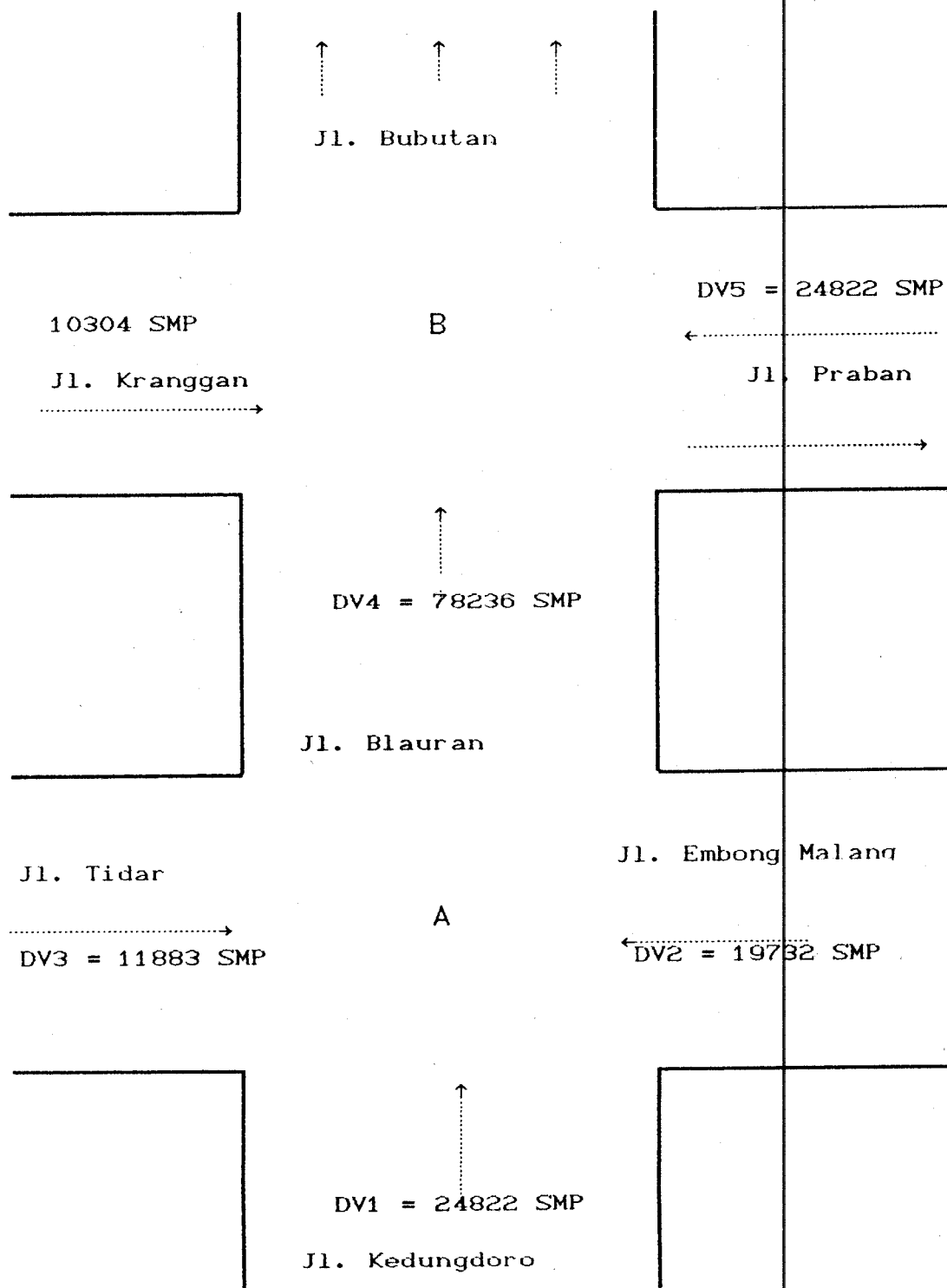
$$U_1 = 6.98 \text{ m/det}$$

$$U_2 = 6.10 \text{ m/det}$$

$$U_3 = 7.21 \text{ m/det}$$

Sedangkan untuk memenuhi data-data untuk perhitungan selanjutnya diambil pendekatan ²⁴⁾ :

24). THEODORE M MATSON, Traffic Engineering, McGraw Hill 1955, hal 420.



GAMBAR 4.3

EKIVALENSI VOLUME KENDARAAN KE SMP

- panjang kendaraan $l = 4$ meter
- waktu reaksi pengemudi $t_d = 1$ detik
- perlambatan mobil $a = 3.6 \text{ m/det}^2$
- waktu hilang saat mulai jalan $k_1 = 3.7$ detik
- panjang siklus $C = 75$ detik

a. Perhitungan waktu hilang

Pada persamaan 2.2 pada bab II telah dirumuskan :

$$K = k_1 + k_2, \text{ sedangkan } k_2 = \frac{W+L}{U}$$

sehingga dengan memasukkan masing-masing harga di atas diperoleh :

- phase 1 $K_1 = 6.35$ detik
- phase 2 $K_2 = 6.32$ detik
- phase 3 $K_3 = 5.92$ detik

b. Perhitungan interval kuning

Pada persamaan 2.1 pada bab II telah dirumuskan

$$t_a = t_d + \frac{U}{2a} + \frac{W+L}{U}$$

- phase 1 $t_{a1} = 1 + \frac{6.98}{2 \times 3.6} + \frac{14.5+4}{6.98} = 4.62 \approx 5$ det
- phase 2 $t_{a2} = 1 + \frac{6.10}{2 \times 3.6} + \frac{12+4}{6.10} = 4.47 \approx 5$ det
- phase 3 $t_{a3} = 1 + \frac{7.21}{2 \times 3.6} + \frac{12+4}{7.21} = 4.22 \approx 5$ det

c. Perhitungan total interval hijau efektif

Pada persamaan 2.3 pada bab II telah dirumuskan

$$G_E = C - \sum_{i=1}^P K_i$$

sehingga didapat :

$$G_E = 75 - 6.35 - 6.32 - 5.92 = 56.41 \text{ detik}$$

d. Perhitungan interval hijau

Pada persamaan 2.4 dalam bab II telah dirumuskan

$$G_i = \frac{DV_i}{DV} G_E + K_i$$

sehingga dapat dihitung pada masing-masing phase adalah

$$\text{- phase 1 } G_1 = \frac{24822}{55437} 56.41 + 6.35 = 31.6 \approx 32 \text{ detik}$$

$$\text{- phase 2 } G_2 = \frac{18732}{55437} 56.41 + 6.32 = 25.3 \approx 25 \text{ detik}$$

$$\text{- phase 3 } G_3 = \frac{11883}{55437} 56.41 + 5.92 = 18.01 \approx 18 \text{ detik}$$

e. Perhitungan interval merah

Interval merah masing-masing phase adalah panjang siklus dikurangi interval hijau. Sehingga hasilnya dapat dilihat pada tabel 4.2.

TABEL 4.2 INTERVAL NYALA LAMPU PERSIMPANGAN A

phase	merah	hijau	kuning
1	43 det	27 det	5 det
2	50 det	20 det	5 det
3	57 det	13 det	5 det

4.5.2 Persimpangan B

Persimpangan B ini hanya terdiri dua phase yaitu

$$\text{-phase 1 } DV_4 = 78\,236 \text{ SMP}$$

$$\text{-phase 2 } DV_5 = 24\,822 \text{ SMP}$$

Jumlah volume kendaraan yang masuk pada persimpangan B DV_B

= 103058. Sedangkan lebar jalan setiap phasanya adalah :

- phase 1 $W_4 = 14.5$ m
- phase 2 $W_4 = 16$ m

Kecepatan rata-rata kendaraan masing-masing phase adalah :

- phase 1 $U_4 = 6.355$ m/det
- phase 2 $U_5 = 6.987$ m/det

TABEL 4.3 INTERVAL NYALA LAMPU PERSIMPANGAN B

phase	merah	hijau	kuning
1	21 det	49 det	5 det
2	54 det	16 det	5 det

Dengan cara perhitungan yang sama dengan yang digunakan pada persimpangan A, didapat tabel 4.3 di atas ini.

4.6 Pengkoordinasian Antar Persimpangan

Telah dijelaskan dalam bab II beberapa macam cara pengkoordinasian antar persimpangan. Pada tugas akhir ini dipilih metode sistem progresip yaitu pengaturan lampu relatif terhadap lampu berikutnya, yang selanjutnya disebut offset. Pada persamaan 2.5 dirumuskan $T = S/U$, sedangkan kecepatan rata-rata kendaraan berubah terhadap kepadatan kendaraan yang telah dirumuskan dalam persamaan 2.7.

Persamaan 2.7 di atas diasumsikan bahwa a_k dan b_k adalah parameter-parameter yang tidak berubah terhadap waktu. Yang berarti bahwa $a_k = a_{k+1}$, a_{k+2} , dan $b_k = b_{k+1}$, b_{k+2} . Sehingga diperlukan dua syarat batas untuk meng-

hitung harga dua parameter yang tidak diketahui tersebut. Syarat batas yang pertama diambil dari hasil perhitungan sub bab sebelumnya. Sedangkan syarat batas yang ke dua diambil dari aturan kecepatan kendaraan dalam kota khususnya pada lokasi penelitian. Yaitu 40 Km per jam.

Dari perhitungan sebelumnya volume kendaraan pada ruas jalan Blauran $DV_4 = 78236$ SMP selama empat belas jam. Sehingga didapat volume kendaraan pada setiap siklusnya 75 detik adalah 116.4 SMP.

Dengan memasukkan ke persamaan $S = U \times t$, rata-rata kecepatan jalan Blauran $U_4 = 6.335$ mper detik, didapat jarak $S = 475$ meter. maka didapat kepadatan kendaraan $Y(k) = 116.4 \text{ SMP}/475 \text{ meter}$ atau $Y(k) = 24.5 \text{ SMP per seratus meter}$. Selanjutnya dimasukkan ke persamaan 2.7 menjadi :

$$6.335 = bk e^{-(24.5/ak)^2} \quad (4.1)$$

Syarat batas yang kedua adalah kecepatan kendaraan maksimum dalam kota khususnya pada jalan Blauran adalah 40 Km per jam atau 11.11 meter per detik, meskipun pada saat $Y(k) = 0$. Sehingga di dapat :

$$11.11 = bk \quad (4.2)$$

Persamaan 4.2 ini dimasukkan ke persamaan 4.1 didapat harga $ak = 0.32.68$. Selanjutnya harga-harga ini dimasukkan ke persamaan 2.7 didapat :

$$U(k) = 11.11 e^{-(Y(k)/32.68)^2} \quad (4.3)$$

Persamaan 2.5 untuk menghitung offset antar persim-

pangan $T = S/U$ di mana jarak antar persimpangan A dan persimpangan B adalah 170 meter. Sedangkan rata-rata kecepatannya berubah terhadap kepadatan kendaraan yang telah dirumuskan pada persamaan (4.3). Sehingga didapat persamaan offset antar persimpangan yang berubah terhadap kepadatan kendaraan yaitu :

$$T = 15.30 e^{(Y(k)/32.68)^2} \quad (4.4)$$

5.1 Pendahuluan

Dalam bab IV telah didapat masing-masing interval nyala lampu pada setiap phase serta persamaan offset antar persimpangan. Sebagai tindak lanjutnya adalah merancang perangkat keras dan perangkat lunaknya.

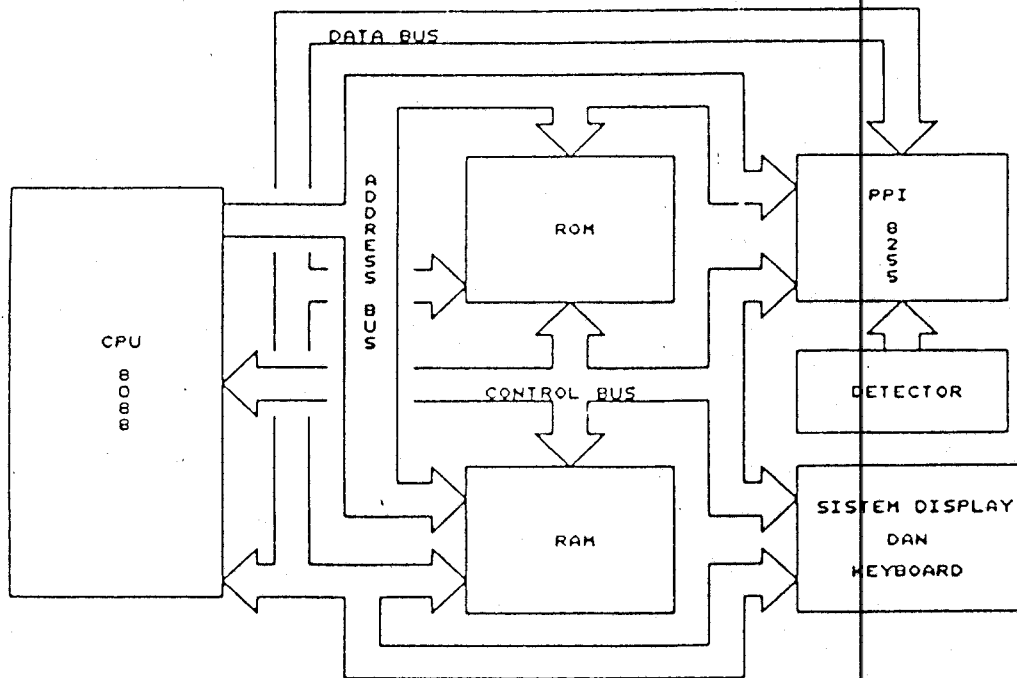
Selanjutnya pada bab ini akan dijelaskan perencanaan alat yang akan dibuat yang meliputi perencanaan perangkat keras dan perangkat lunak.

Untuk memperjelas sistem yang akan dibuat, blok diagram sistem ditunjukkan pada gambar 5.1. Sebagai pusat dari sistem tersebut adalah CPU 8088 akan menyalakan lampu dengan mengirim data tertentu. Sedangkan sebagai umpan balik berupa detektor yaitu suatu rangkaian yang mewakili sensor kepadatan kendaraan pada setiap saat.

Secara terperinci blok diagram tersebut mempunyai fungsi sebagai berikut :

- CPU 8088 sebagai unit pengendali dan pemroses data yang ditunjang dengan buffer yang berfungsi menguatkan dan mengatur arah sinyal.
- ROM sebagai penyimpan program dan ditunjang oleh decoder

yang berfungsi mendekodekan jalur address bus (A0-A19) menjadi suatu address tertentu sehingga suatu unit hanya dapat diaktifkan dengan memberi bilangan address yang benar.



GAMBAR 5.1 BLOK DIAGRAM SISTEM YANG DIBUAT

- RAM sebagai penyimpan data atau media storage yang volatile dan juga ditunjang decoder seperti ROM, tetapi ditunjukan pada alamat RAM.
- Unit PPI yang bekerja sebagai penghubung antara minimum sistem 8088 dengan dunia luar dalam hal ini tugasnya menerima data dari detektor dan mengeluarkan data ke sistem display.

- Sistem display adalah berupa lampu-lampu LED yang mewakili sebagai lampu-lampu lalu lintas. LED-LED ini dipasang pada papan model tersendiri.

5.2 Perencanaan CPU 8088 dalam Minimum Mode

Dalam perencanaan modul minimum sistem 8088 ini, terbagi menjadi beberapa tahap. Tahap tersebut adalah tahap perencanaan rangkaian clock, rangkaian buffer, control bus dan decoder memori dan unit input output.

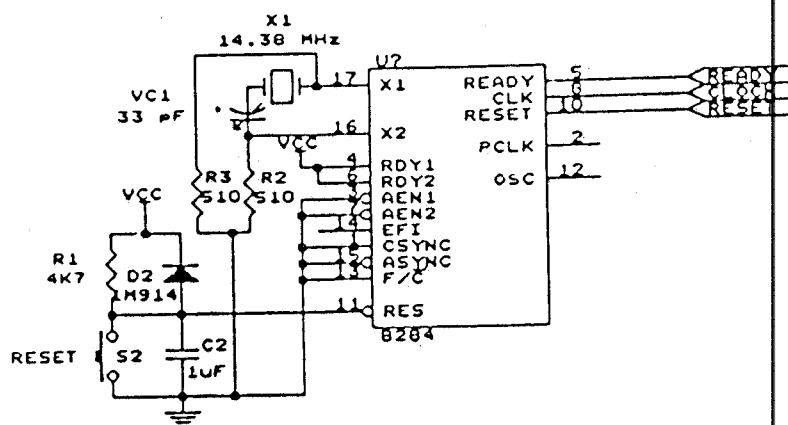
5.2.1 Rangkaian clock

Dasar waktu untuk sinkronisasi operasi-operasi internal dan eksternal mikroprocessor 8088 diberikan oleh input sinyal clock. Mikroprocessor 8088 yang beroperasi pada frekwensi clock 5 MHz. Clock secara eksternal dihasilkan oleh clock generator 8284. Dari gambar rangkaian clock terdapat dua output clock, yakni peripheral clock (PCLK) dan oscillator (OSC). Sinyal-sinyal ini diberikan untuk medrive IC-IC periperal. Sinyal clock output pada PCLK setengah dari frekwensi CLK, atau tergantung pada crystal yang dipasang. Sedangkan output OSC tiga kali lipat frekuensi output CLK.

Dalam perencanaan Tugas Akhir ini frekwensi clock 4.77 MHz digunakan crystal 14.31818 yang diinputkan pada X1 dan X2 dari 8284. Untuk lebih jelasnya dapat dilihat pada gambar 5.2.

5.2.2 Rangkaian reset

Pada waktu catu daya diberikan untuk pertama kali pada mikroprocessor (saat power up), semua register termasuk program counter-nya berada dalam keadaan acak (random). Karena itu diperlukan cara tertentu agar PC menunjuk pada lokasi awal program.



GAMBAR 5.2 RANGKAIAN CLOCK DAN RESET 8284

Sinyal reset juga diperlukan oleh CPU 8088 untuk menghentikan semua aktifitas yang dikerjakan dan memulai mengerjakan aktifitas awal. Secara normal sinyal reset digunakan untuk memulai sistem atau bila sistem gagal. Ketika sinyal reset "high" CPU 8088 akan melakukan instruksi yang ada pada address absolut FFF0H.

Untuk menghasilkan sinyal reset ini direncanakan dengan menggunakan sebuah schmitt trigger berbentuk gate NAND dan switch push button. Saat switch belum ditekan maka, output schmit trigger sistem akan terreset.

5.2.3 Buffer

Buffer digunakan untuk menambah kemampuan mikro-processor agar dapat dibebani beberapa IC.

Dalam perencanaan Tugas Akhir ini dibuat rangkaian buffer 74LS244 unidirectional untuk A8 sampai A15 dimana ke dua pin enable dari 74LS244 diketanahkan, untuk memisahkan antara address data pada pin ADO sampai AD7 menggunakan IC 74LS373 latch, di mana output controlnya diketanahkan, sedang pin latch enablenya dihubungkan langsung ke pin ALE 8088. Untuk address A16 sampai A19 diberikan IC latch 74LS373 di mana pin latch enablenya dihubungkan dengan ALE dari 8088. Untuk memperoleh data maka pin-pin ADO-AD7 juga dihubungkan ke bidirectional buffer 74LS245, dimana pin enablenya dihubungkan dengan pin DEN dari 8088, sedang pin directionalnya dihubungkan ke pin DT/R dari 8088.

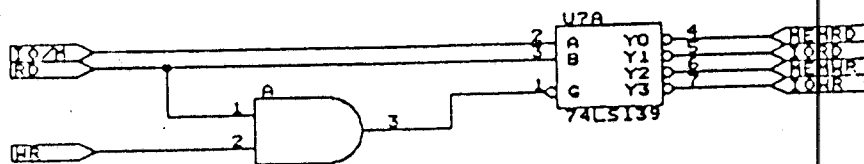
Rangkaian lengkap dari buffer dapat dilihat pada lampiran B pada bagian akhir buku ini.

5.2.4 Control bus

Untuk mendapatkan control bus, pin-pin IO/M dan RD masing-masing dihubungkan ke pin A dan B dari decoder 74LS139. Pin G dari 74LS139 dihubungkan dengan input RD dan WR yang merupakan sinyal aktif low dari 8088. Maka tabel kebenaran dari decoder 74LS139 adalah sebagai berikut :

TABEL 5.1 TABEL KEBENARAN DEKODER 74LS139

RD	IO/M	PIN YANG AKTIF
0	0	Y0 = MEMRD
0	1	Y1 = IORD
1	0	Y2 = MEMWR
1	1	Y3 = IOWR



GAMBAR 5.3 RANGKAIAN CONTROL BUS

5.2.5 Rangkaian decoder memori

Sebelum merencanakan sistem hardware, terlebih dahulu dibuat memori map untuk mengalokasikan komponen memori yang digunakan. Pada gambar 5.4 diperlihatkan memori map dari sistem. Dalam perencanaan ini digunakan dua jenis memori yaitu ROM dan RAM.

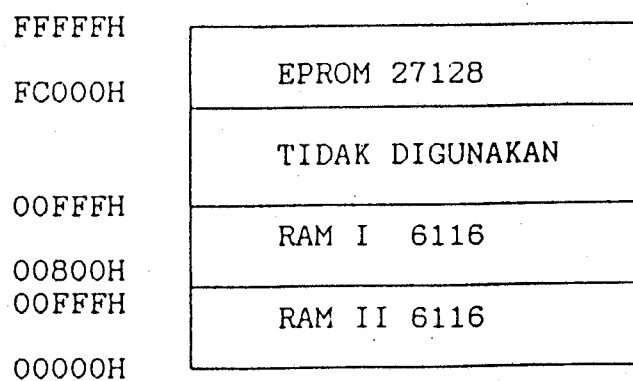
Pada perencanaan tugas akhir ini digunakan chip ROM dengan kapasitas 16KB (27128). Sedang untuk menyimpan data resident dipakai RAM 6116 (2KB)

1. Address 00000H - 007FFH, digunakan untuk RAM 6116 yang berfungsi untuk menyimpan data offset antar persimpangan dan jumlah kendaraan.
2. Address 00800H-00FFFH, yang berfungsi untuk ekspansi

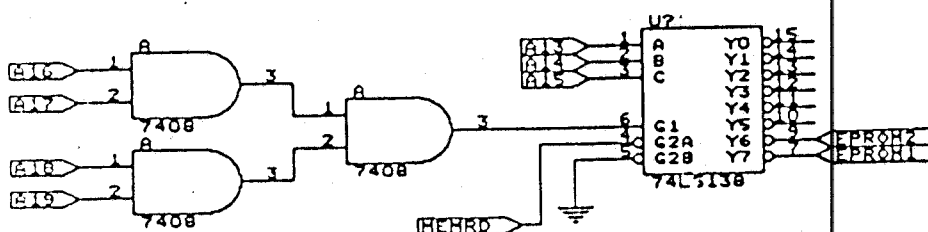
memori.

3. Address 01000H - FBFFFH, tidak digunakan
4. Address FC000H - FFFFFH, digunakan untuk EPROM yang berisi program untuk menjalankan seluruh sistem.

Untuk rangkaian decoder dipakai IC 74LS138 yakni suatu chip decoder 3 input 8 output. Sedangkan rangkaian lengkapnya dapat dilihat pada gambar 5.5 untuk rangkaian decoder ROM dan gambar 5.6 untuk rangkaian decoder RAM.



GAMBAR 5.4 MEMORI MAP SISTEM



GAMBAR 5.5 RANGKAIAN DECODER ROM

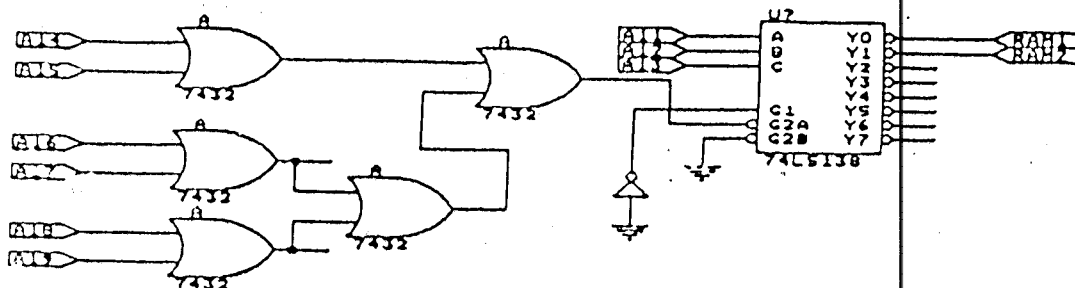
Jika diperhatikan untuk ROM 27128 mempunyai zone A13

sampai A19. Maka bila disusun dalam tabel perencanaan decoder, seperti pada tabel 5.2

TABEL 5.2 PERENCANAAN DECODER ROM.

		A19 - A0
EPROM 27128	FFFFFFH	1111 1111 1111 1111 1111
	FC000H	1111 1100 0000 0000 0000

Sedangkan pada RAM 6116 mempunyai zone A11 sampai A19, maka bila disusun dalam tabel akan dihasilkan tabel 5.3 perencanaan decoder.



GAMBAR 5.6 RANGKAIAN DECODER RAM

TABEL 5.3 PERENCANAAN DECODER RAM

		A19 - A0
RAM 1 6116	00000H	0000 0000 0000 0000 0000
	00FFFH	0000 0000 0111 1111 1111
RAM 2 6116	00800H	0000 0000 1000 0000 0000
	01FFFH	0000 0001 1111 1111 1111

5.2.6 Rangkaian decoder unit input output

Seperti halnya dalam decoding memori, maka komponen I/O perlu untuk di dekoding untuk mengatur lokasi dari beberapa chip I/O. Pada perencanaan Tugas Akhir ini dipakai 8255 yang memerlukan pengaturan alamat. Dalam perencanaan Tugas Akhir ini, untuk port A, B, C, dan CONW diberikan alamat 20, 21, 22, dan 23. Tabel 5.4 memperlihatkan pengalamatan dari PPI 8255.

TABEL 5.4 ALAMAT PORT 8255

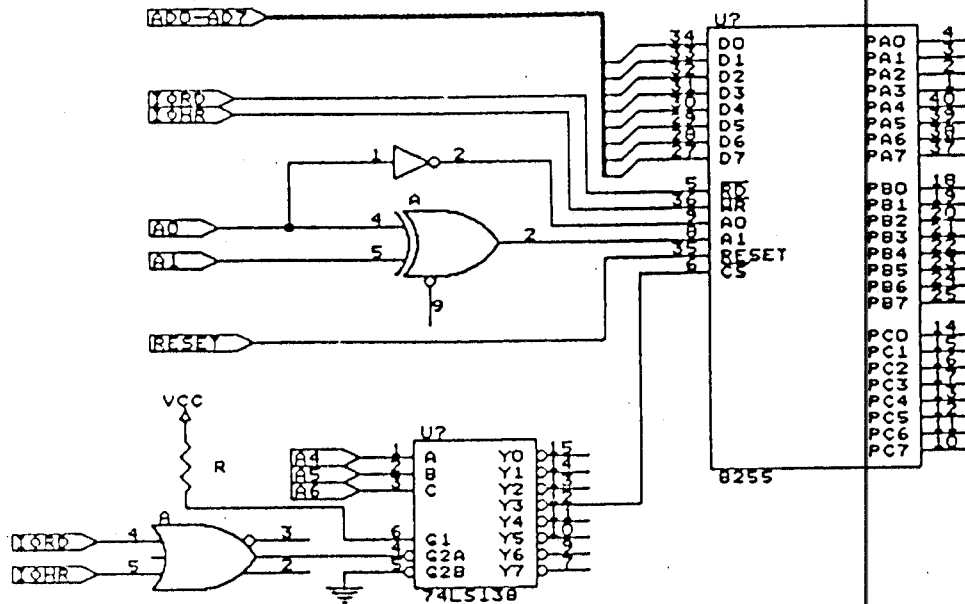
ALAMAT	A7	A6	A5	A4	A3	A2	A1	A0	
20	0	0	1	0	0	0	0	0	PORT A 8255
21	0	0	1	0	0	0	0	1	PORT B 8255
22	0	0	1	0	0	0	1	0	PORT C 8255
23	0	0	0	0	0	0	1	1	CONTROL WORD

Dari hasil tabel konfigurasi alamat maka dapat direncanakan hubungan pin-pin dari decoder yang dipakai yaitu 74LS138 yang mempunyai masukan sebanyak 6 pin. Skema dari rangkaian decoder I/O dapat dilihat pada gambar 5.7 yang menunjukkan hubungan pin-pin masukan dan pin-pin keluaran dari decoder 74LS138.

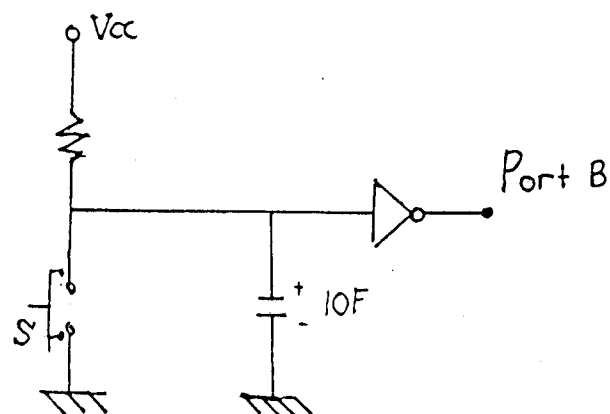
5.3 Rangkaian Detektor

Pada pembuatan model lampu lalu lintas ini untuk mendeteksi kepadatan kendaraan pada ruas JL. Blauran, digunakan detektor pust button yang bekerja berdasarkan

atas penekanan. Pada detektor masuk maka jika pust button



GAMBAR 5.7 RANGKAIAN DECODER I/O



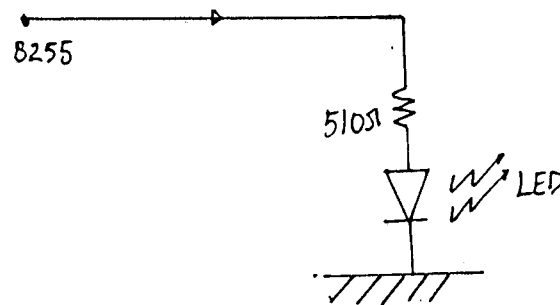
GAMBAR 5.8 RANGKAIAN DETEKTOR KENDARAAN

ditekan berarti ada kendaraan masuk dan pada detektor keluar bila pust button ditekan berarti ada kendaraan

lewat. Dimana pada setiap penekanan adalah satu SMP, sedangkan rangkaiannya dapat dilihat pada gambar 5.8.

5.4 Rangkaian keluaran

Seperti yang telah dijelaskan di bab terdahulu, bahwa sistem yang dibuat hanya berupa model bagi lampu-lampu lalu lintas maka, keluaran dari sistem ini berupa sinyal untuk menghidupkan atau mematikan lampu LED. Rangkaian untuk menghubungkan PPI 8255 dengan lampu LED ditunjukkan pada gambar 5.9



GAMBAR 5.9 RANGKAIAN UNTUK MENYALAKAN LED

5.5 Perencanaan Software

Perencanaan perangkat keras seperti yang telah diuraikan di depan tidak akan dapat berfungsi bila tidak didukung oleh perangkat lunak. Perangkat lunak ini merupakan suatu program untuk mengaktifkan perangkat keras. Jadi perangkat lunak ini digunakan untuk mengatur kerja dari sistem secara keseluruhan mulai dari inisialisasi, pembacaan data dari detektor, pengolahan data, dan akhirnya mengendalikan nyala lampu LED.

Langkah pertama untuk merencanakan perangkat lunak adalah membuat suatu diagram pewaktuan bagi urutan penyalaan masing-masing LED dan kemudian diagram alirnya (flow chart bagi sistem yang dibuat). Selanjutnya baru disusun programnya.

Flow chart dari perencanaan alat ini dapat dilihat pada gambar 5.10 di bawah ini. Sedangkan program dari sistem ini dapat dilihat dilampiran A.

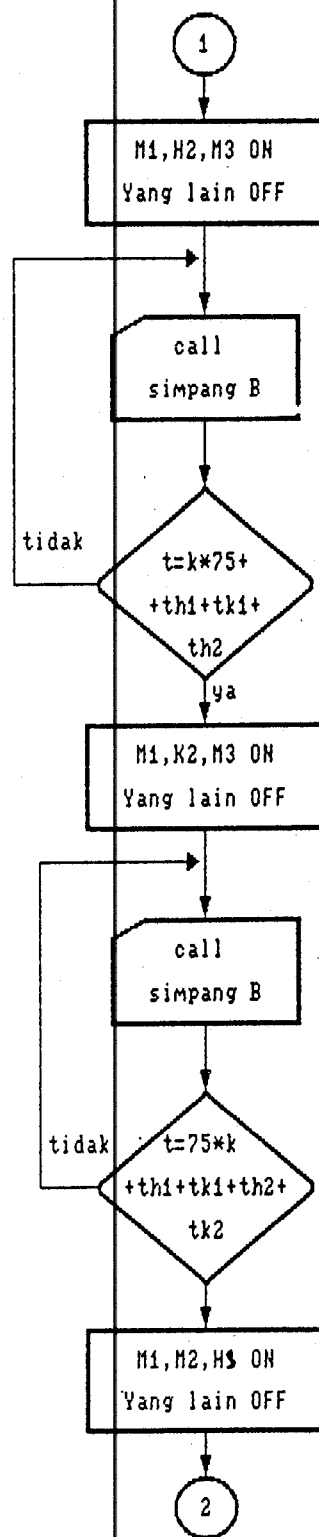
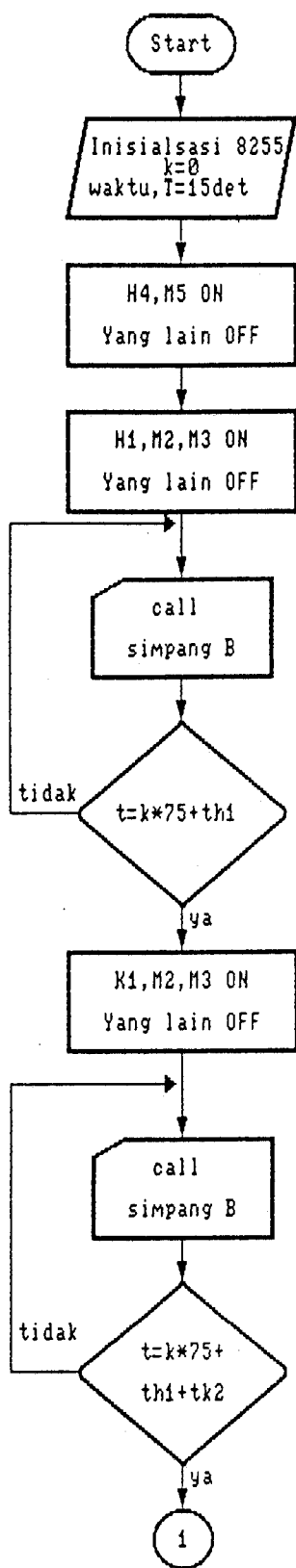
5.6 Penerapan Sistem Rancangan

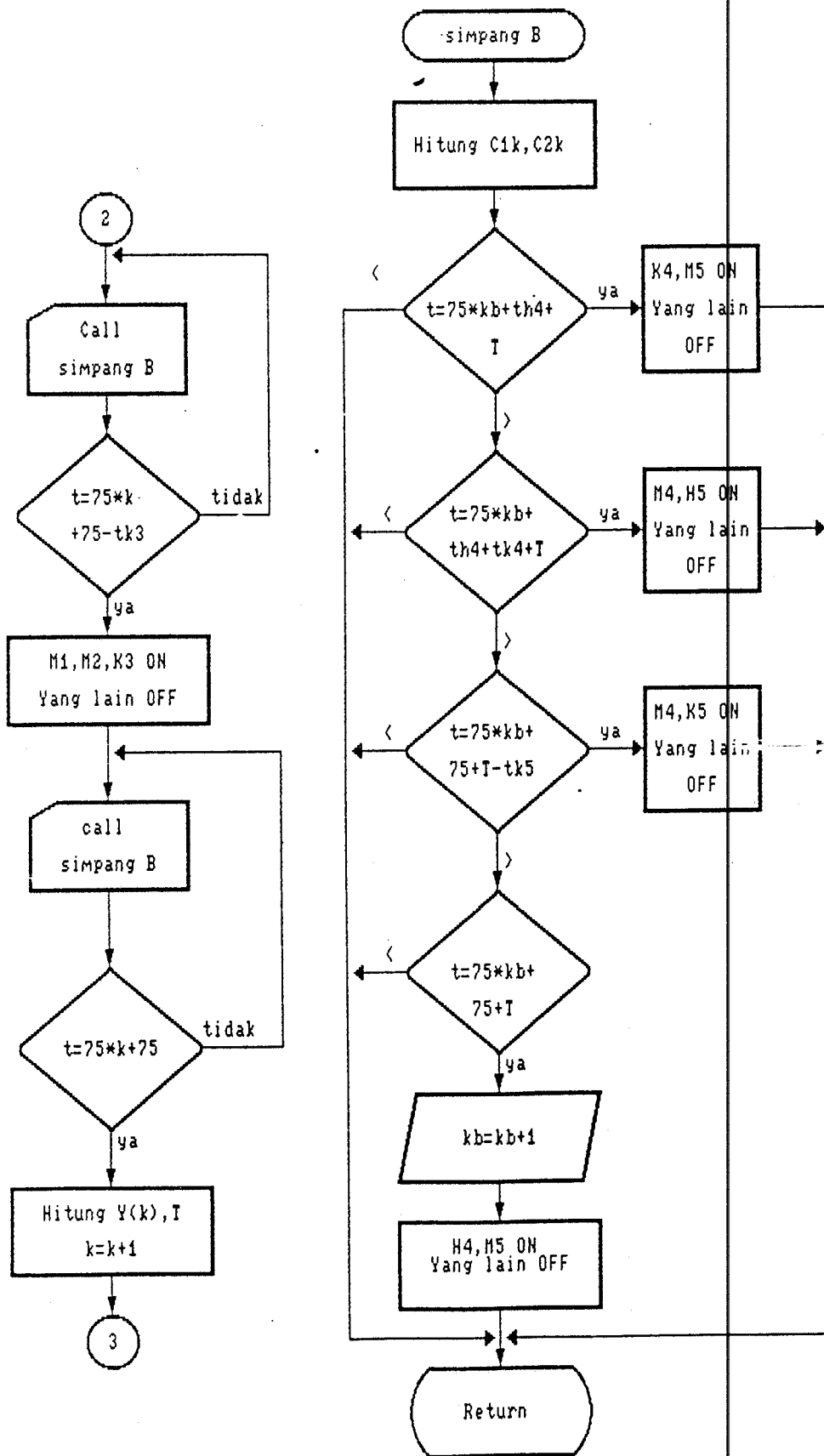
Untuk menerapkan hasil rancangan model lampu lalu lintas sistem progresip tersebut, maka perlu diadakan beberapa modifikasi pada model yang ada, meliputi modifikasi sistem detektor dan rangkaian keluaran.

5.6.1 Modifikasi rangkaian sistem detektor

Sistem detektor yang dipakai pada modul rancangan model tentunya tidak dapat dipakai untuk keperluan yang sebenarnya. Ada beberapa sistem detektor kepadatan kendaraan yang dapat digunakan, yang bekerja berdasarkan sinar infra merah, ultrasonik, kamera, dan lain-lain. Pada lampiran dapat dilihat dengan jelas perbandingan beberapa jenis detektor serta keuntungan dan kerugiannya.

Sistem detektor yang harus dipasang tersebut harus memenuhi persyaratan bahwa detektor tidak mudah terpengaruh oleh kondisi cuaca lingkungan sekitar di mana detektor dipasang.

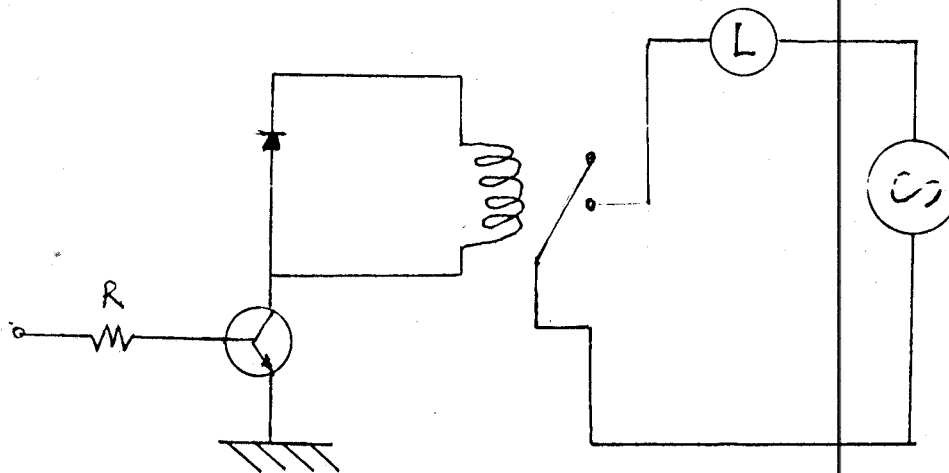




GAMBAR 5.10 FLOW CHART DARI SISTEM YANG DIBUAT

5.6.2 Modifikasi rangkaian keluaran

Untuk keperluan yang sebenarnya, maka lampu yang dipakai adalah lampu AC yang dihubungkan dengan jala-jala PLN. Dengan demikian diperlukan rangkaian tersendiri yang dapat memisahkan antara sinyal digital dengan sinyal analog. Rangkaian ini dapat dilihat gambar 5.11



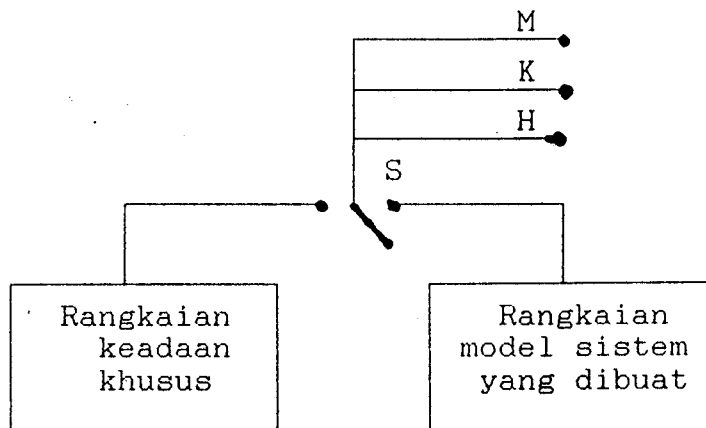
GAMBAR 5.11 RANGKAIAN KELUARAN 8255 DENGAN JALA-JALA PLN

5.7 Keadaan Khusus.

Bila ada suatu keadaan khusus yang harus mendapat prioritas, misalnya ada rombongan pejabat tinggi yang akan lewat, maka diperlukan suatu penanganan nyala lampu lalu lintas secara tersendiri pula pada daerah tersebut.

Kalau diambil contoh bahwa rombongan kendaraan yang mendapat prioritas tersebut akan melewati persimpangan A Jl. Blauran ke Jl. Bubutan, maka pada saat rombongan ter-

sebut tiba, lampu H1, M2, M3 harus menyala pada persimpangan A. Sedangkan pada persimpangan B, lampu H4 dan M5 harus menyala. Hal ini dapat dilakukan dengan memindahkan saklar S ke keadaan khusus.



GAMBAR 5.12 PENANGANAN KEADAAN KHUSUS

Dari uraian Tugas Akhir ini dapat diambil kesimpulan sebagai berikut :

- Pada suatu daerah di mana terdapat banyak deretan persimpangan jalan, penerapan sistem lampu lalu lintas sistem progresip lebih menguntungkan dibanding dengan sistem lain yang berdiri sendiri.
- Penggunaan minimum sistem dengan mikroprocessor 8088 untuk mengatur nyala lampu lalu lintas yang dilengkapi dengan rangkaian detektor kepadatan lalu lintas ini sangat tepat, karena setelah kepadatan lalu lintas terukur di umpan balikkan ke mikroprocessor. Sehingga mikroprocessor akan memberikan suatu keputusan yang tepat pada harga offset yang sesuai.
- Setiap kali offset berubah naik maka memperpanjang nyala lampu hijau pada phase empat pada siklus tersebut. Juga setiap kali offset turun, maka memperpanjang nyala lampu hijau phase lima pada siklus tersebut.
- Pemakaian minimum sistem dalam berbagai bidang dapat memberikan suatu kemudahan. Untuk keperluan yang berbeda hanya diperlukan sedikit modifikasi pada perangkat keras dan lunaknya.

Sudah menjadi tuntutan saat ini untuk mengkoordinasi antar persimpangan yang berdekatan dan saling mempengaruhi, selanjutnya dari koordinasi antar persimpangan dikoordinasikan persimpangan-persimpangan dalam suatu kota yang diawasi oleh komputer sebagai pusat pengendalian.

DAFTAR PUSTAKA

1. Dauglas V. Hall, Microprocessor and Interfacing programming and Hardware, McGraw Hill Book Co. Singapore, 1986.
2. Eddy Napitulu, Studi Pembenahan Arus lalu lintas di Daerah Wonokromo, Tugas Akhir Sipil ITS 1989.
3. Happy Budhiati, Studi Dua persimpangan di Jalan Kusuma bangsa, Tugas Akhir Sipil ITS 1987.
4. James W. Coffron, Z80 Applications, Berkeley 1983.
5. James W. Coffron, Programming The 8086/8088, Sybex Inc, Berkeley California, 1984.
6. Lewis C. Eggerbrecht, Interfacing to IBM Personal Computer, Horward M. Sams & Co. Inc. 1985.
7. Martin Wohldan Brian V. Martin, Traffic System Analysis, McGRAW-HILL New York 1967.
8. Mischa Schwartz & Leonard Shaw, Signal processing, Discrete Spectral Analysis, Detections and Estimation.
9. Singh Avtar and Walter A. triebel, The 8088 Micro-processor Programming, Interfacing, Software, Hardware, and Applications, Prentice-Hall internasional Inc., Englewood Cliffs, 1989.
- 10 Theodore M. Matson dan Wilbur S. Smith dkk., Traffic Engineering, McGRAW-HILL, New York 1955.

```

;=====
;=
;=      Program simulasi traffic light
;=
;=      oleh :
;=
;=      Mokhammad Solehuddin
;=      2852200390
;=
;=      -----
;=      PORT A      | D7 D6 D5 D4 D3 D2 D1 D0 |
;=                  |-----|
;=                  | K3 H3 H1 K1 M1 H2 K2 M2 |
;=                  |-----|
;=
;=      PORT C      | D7 D6 D5 D4 D3 D2 D1 D0 |
;=                  |-----|
;=                  | - M5 M3 H4 K4 M4 H5 K5 |
;=                  |-----|
;=
;=====

```

```

Counter      equ 020h      ;Variabel
AllCounter   equ 030h
Offset       equ 040h
Periode      equ 050h
StatusLama   equ 060h
StatusBaru   equ 070h
StatusMed    equ 080h
Kendaraan    equ 090h
LED1         equ 0A0h
LED2         equ 0E0h
Digit13      equ 0C0h
Digit24      equ 0D0h
DataKu       equ 0E0h
PortA1       equ 250h      ;PPI pertama
PortB1       equ 251h
PortC1       equ 252h
PortCW1      equ 253h
PortA2       equ 020h      ;PPI kedua
PortB2       equ 021h
PortC2       equ 022h
PortCW2      equ 023h
th1          equ 27        ;Waktu Hijau 1
th2          equ 20        ;Waktu Hijau 2
th3          equ 13        ;Waktu Hijau 3
th4          equ 49        ;Waktu Hijau 4
th5          equ 16        ;Waktu Hijau 5
tk           equ 5         ;Waktu Kuning

```

```

Sys8088 segment
  assume cs:sys8088,ds:sys8088,ss:sys8088,es:sys8088
  org 0
  db 2000h dup (0FFh)
  org 0

Awal:
  Cli
  mov ax,0 ;Inisialisasi Stack
  mov ss,ax
  mov ax,07FFh
  mov sp,ax
  mov ax,0
  mov ds,ax
  mov al,10000010b ;Inisialisasi PPI8255
  mov dx,PortCW1 ;Port A - K3H3H1K1M1H2K2M2
  out dx,al ;Port B - Input keyboard
  ;Port B7 - Input Time Base 1 Hz
  ;Port C - xxM5M3H4K4M4H5K5
  mov al,10000000b ;Port B - Data 7's Segment
  mov dx,PortCW2 ;Port C - Control 7's Segment
  out dx,al

BalikNol: ;Clear buffer
  mov si,0
  mov cx,0ffh
  mov al,0

NolKan:
  mov [ds:si],al
  inc si
  loop NolKan
  mov [ds:Counter],al ;Counter=0
  mov al,15
  mov cx,6
  mov bx,0
  mov si,DataKu

Isi05:
  mov [ds:si+bx],al
  inc bx
  loop Isi05
  mov cx,5
  mov al,16
  mov bx,6

Isi610:
  mov [ds:si+bx],al
  inc bx
  loop Isi610
  mov al,17
  mov cx,5
  mov bx,11

```


Isi1118:

```
mov [ds:si+bx],al
inc bx
mov [ds:si+bx],al
inc bx
inc al
loop Isi1118
mov cx,8
mov al,21
mov bx,19
```

Isi1926:

```
mov [ds:si+bx],al
inc bx
inc al
Loop Isi1926
mov al,30
mov [ds:si+bx],al
mov al,31
mov cx,4
mov bx,28
```

Isi2831:

```
mov [ds:si+bx],al
inc bx
inc al
inc al
Loop Isi2831
mov cx,4
mov al,41
mov bx,32
```

Isi3235:

```
mov [ds:si+bx],al
inc al
inc al
inc al
inc bx
loop Isi3235
mov al,54
mov cx,10
mov bx,36
```

Isi36L:

```
mov [ds:si+bx],al
inc al
inc al
inc al
inc al
inc bx
Loop Isi36L
mov dx,PortB1
in al,dx
and al,80h
```

```

mov     [ds:StatusLama],al
mov     ax,0
mov     [ds:Kendaraan],ax           ;Kendaraan=0
mov     al,15
mov     [ds:Offset],al             ;offset (T) = 15
mov     ax,0
mov     [ds:AllCounter],ax         ;Waktu = 0
mov     [ds:Periode],al           ;periode = 0
mov     dx,PortB1
mov     ah,[ds:StatusLama]         ;Menunda 1 dt

Tunda1Hz:
in      al,dx
and     al,80h
cmp     al,ah
jz      Tunda1Hz
mov     [ds:statusLama],al
mov     [ds:StatusMed],al
mov     byte ptr [ds:LED1],00100001b ;H1M2M3H4M5
mov     byte ptr [ds:LED2],01110000b
call    NyalaLED
call    TampilKendaraan

LoopAwal:
call    CheckTime                 ;Bandingkan Counter dengan
mov     ah,th1                    ;Waktu Hijau 1
mov     al,[ds:Counter]
cmp     al,ah
jb      LoopAwal
mov     byte ptr [ds:LED1],00010001b ;H1-K1
call    NyalaLED

Awal2:
call    CheckTime                 ;Bandingkan Counter dengan
mov     ah,th1                    ;Waktu Hijau 1 +
add     ah,tk                     ;Waktu Kuning +
mov     al,[ds:Counter]
cmp     al,ah
jb      Awal2
mov     byte ptr [ds:LED1],00001100b ;K1-M1 = M2-H2
call    NyalaLED

Awal3:
call    CheckTime                 ;Bandingkan Counter dengan
mov     ah,th1                    ;Waktu Hijau 1 +
add     ah,tk                     ;Waktu Kuning +
add     ah,th2                    ;Waktu Hijau 2 +
mov     al,[ds:Counter]
cmp     al,ah
jb      Awal3
mov     byte ptr [ds:LED1],00001010b ;H2-K2
call    NyalaLED

```

```

Awal4:
  call CheckTime           ;Bandingkan Counter dengan
  mov ah,th1              ;Waktu Hijau 1 +
  add ah,tk               ;Waktu Kuning +
  add ah,th2              ;Waktu Hijau 2 +
  add ah,tk               ;Waktu Kuning
  mov al,[ds:Counter]
  cmp al,ah
  jb Awal4
  mov byte ptr [ds:LED1],01001001b ;K2-M2 = M3-H3
  mov al,[ds:LED2]
  and al,11011111b       ;M3-H3
  mov [ds:LED2],al
  call NyalaleD

Awal5:
  call CheckTime           ;Bandingkan Counter dengan
  mov ah,th1              ;Waktu Hijau 1 +
  add ah,tk               ;Waktu Kuning +
  add ah,th2              ;Waktu Hijau 2 +
  add ah,tk               ; Waktu Kuning +
  add ah,th3              ;Waktu Hijau 3
  mov al,[ds:Counter]
  cmp al,ah
  jb Awal5
  mov byte ptr [ds:LED1],10001001b ;H3-K3
  call NyalaleD

Awal6:
  Call CheckTime           ;Check counter apakah
  mov al,[ds:Counter]     ;sudah 75 detik ?
  cmp al,75
  jb Awal6
  mov byte ptr [ds:Counter],0 ;Counter = 0
  call hitung             ;Hitung T (offset)
  mov Byte ptr [ds:LED1],00100001b
  mov al,[ds:LED2]       ;H1M2M3
  or al,00100000b
  mov [ds:LED2],al
  Call NyalaleD
  jmp LoopAwal

CheckTime proc near
  push ax
  push bx
  push dx

CT0:
  mov bx,[ds:Kendaraan] ;Check kendaraan
  mov dx,PortB1          ;Port B untuk kendaraan
  in al,dx               ;B7 - 1 Hz
  mov ah,al              ;B5 - Input 1
  and al,20h             ;B4 - Input 2
  jnz CT1                ;B3 - Input 3

```

```
    inc    bx                ;B2 - Output 1

CT1:                                ;B1 - Output 2
    mov    al,ah            ;B0 - Output 3
    and    al,10h
    jnz    CT2
    inc    bx

CT2:
    mov    al,ah
    and    al,8
    jnz    CT3
    inc    bx

CT3:
    mov    al,ah
    and    al,4
    jnz    CT4
    cmp    bx,0
    je     CT6
    dec    bx

CT4:
    mov    al,ah
    and    al,2

    jnz    CT5
    cmp    bx,0
    je     CT6
    dec    bx

CT5:
    mov    al,ah
    and    al,1
    jnz    CT6
    cmp    bx,0
    je     CT6
    dec    bx

CT6:
    mov    [ds:Kendaraan],bx
    Call   TampilKendaraan
    mov    dx,PortB1
    in     al,dx
    and    al,7Fh
    cmp    al,7Fh
    je     CT7
    Call   delay

CT7:
    mov    ah,[ds>StatusMed]
    mov    dx,PortB1
    in     al,dx
    and    al,80h
```

```

cmp     al,ah
je      CTO                ;Check satu detik
mov     [ds:StatusMed],al
mov     ah,[ds:StatusLama]
cmp     al,ah
jne     CTO
mov     al,[ds:Counter]   ;Naikkan Counter
inc     al
mov     [ds:Counter],al
mov     ax,[ds:AllCounter] ;Naikkan waktu
inc     ax
mov     [ds:AllCounter],ax
call    Simpang           ;Simpangan ke dua
pop     dx
pop     bx
pop     ax
ret
CheckTime endp

```

```

Hitung proc near
push   ax
push   bx
mov     bx,[ds:Kendaraan]
cmp     bx,42
jbe    HT1
mov     bx,42

```

```

HT1:
mov     si,DataKu
mov     al,[ds:si+bx]
cmp     al,75
jbe    HT2
mov     al,75

```

```

HT2:
mov     [ds:Offset],al
pop     bx
pop     ax
ret
Hitung endp

```

```

Simpang proc near
push   ax
push   dx
mov     al,[ds:Periode]   ;Periode
mov     ah,0
mov     dl,75              ; * 75
mul     dl
mov     dx,ax
mov     al,th4
mov     ah,0
add     dx,ax              ; + waktu hijau 4
mov     al,[ds:Offset]
mov     ah,0

```

```

add    dx,ax                ; + offset (T)
mov    ax,[ds:AllCounter]
cmp    ax,dx                ; Bandingkan waktu dengan
jb     sip                  ; Hasil diatas
jne    Simpang1
mov    ah,[ds:LED2]        ; H4-K4
and    ah,00100000b
or     ah,01001000b        ; K4M5
mov    [ds:LED2],ah
call   NyalaLED
Sip:
      jmp    Simpang5

```

```

Simpang1:
mov    ah,0
mov    al,tk
add    dx,ax                ; + waktu kuning
mov    ax,[ds:AllCounter]  ; Bandingkan waktu dengan
cmp    ax,dx                ; Hasil diatas
jb     Simpang5
jne    Simpang2
mov    ah,[ds:LED2]        ; K4-M4
and    ah,00100000b
or     ah,00000110b        ; M4H5
mov    [ds:LED2],ah
call   NyalaLED
jmp    Simpang5

```

```

Simpang2:
mov    al,[ds:Periode]     ; Periode
inc    al                  ; Periode = Periode + 1
mov    dl,75
mul    dl                  ; * 75
mov    dx,ax
mov    al,tk
mov    ah,0
sub    dx,ax                ; - waktu kuning
mov    al,[ds:Offset]
mov    ah,0
add    dx,ax                ; + T
mov    ax,[ds:AllCounter]  ; Bandingkan waktu dengan
cmp    ax,dx                ; Hasil diatas
jb     Simpang5
jne    Simpang3
mov    ah,[ds:LED2]        ; H5-K5
and    ah,00100000b
or     ah,00000101b        ; M4K5
mov    [ds:LED2],ah
call   NyalaLED
jmp    Simpang5

```

```

Simpang3:
mov    ah,0
mov    al,tk

```

```

add    dx,ax                ; + Waktu kuning
mov    ax,[ds:AllCounter]  ; Bandingkan Waktu dengan
cmp    ax,dx                ; Hasil diatas
jb     Simpang5
mov    al,[ds:Periode]    ; Periode = Periode + 1
inc    al
mov    [ds:Periode],al
mov    ah,[ds:LED2]        ; H4M5
and    ah,00100000b
or     ah,01010000b
mov    [ds:LED2],ah
call   NyalaLED

Simpang5:
pop    dx
pop    ax
Ret
Simpang endp

NyalaLED proc near
push  dx
push  ax
mov   al,[ds:LED1]
mov   dx,PortA1
out   dx,al                ;Nyalakan LED group 1
mov   al,[ds:LED2]
mov   dx,PortC1
out   dx,al                ;Nyalakan LED group 2
pop   ax
pop   dx
ret
NyalaLED endp

TampilKendaraan proc near
Push  ax
Push  dx
call  Conversi
mov   al,[ds:Digit13]
mov   dx,PortB2
out   dx,al
mov   al,1
mov   dx,PortC2
out   dx,al
mov   al,0
mov   dx,PortC2
out   dx,al
mov   al,[ds:Digit24]
mov   dx,PortB2
out   dx,al
mov   al,2
mov   dx,PortC2
out   dx,al
mov   al,0
mov   dx,PortC2

```

```

    out    dx,al
    pop    dx
    pop    ax
    ret
TampilKendaraan endp

Conversi proc near
    Push  ax
    Push  bx
    Push  cx
    Push  dx
    mov   ax,[ds:Kendaraan]
    mov   dx,0
    mov   bx,1000
    div   bx
    cmp   al,9
    jbe   CN1
    mov   al,9

CN1:
    mov   bh,al                ;bh - digit 1
    mov   ax,dx
    mov   dl,100
    div   dl
    mov   bl,al                ;bl - digit 2
    mov   al,ah
    mov   ah,0
    mov   dl,10
    div   dl                    ;dl - digit 3
    mov   dx,ax                ;dh - digit 4
    mov   cl,4
    shl   dl,cl
    or    bh,dl
    mov   [ds:Digit13],bh
    mov   cl,4
    shl   dh,cl
    or    bl,dh
    mov   [ds:Digit24],bl
    Pop   dx
    Pop   cx
    Pop   bx
    Pop   ax
    ret

Conversi endp

delay proc near                ;Delay Time
    push ax
    push cx
    push dx
    mov   cx,20000
Delay1:
    mov   ah,[ds:StatusMed]
    mov   dx,PortB1
    in    al,dx

```



```
and    al,80h
cmp    al,ah
jne    Delay2
loop   Delay1
```

```
Delay2:
```

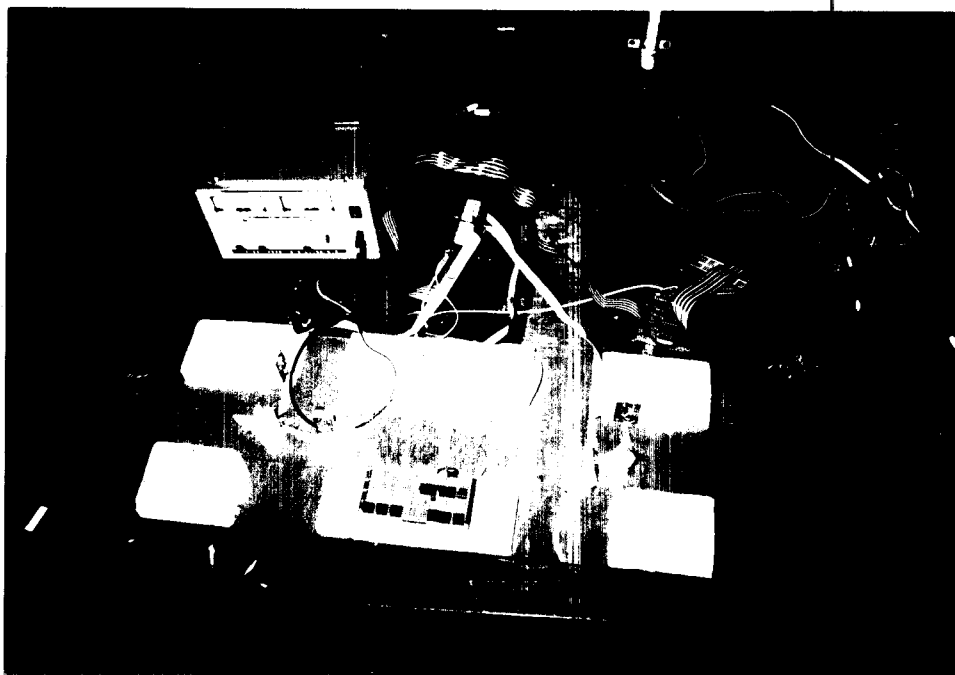
```
pop    dx
pop    cx
pop    ax
ret
delay endp
```

```
org 1FF0h
jmp Far ptr Awal
```

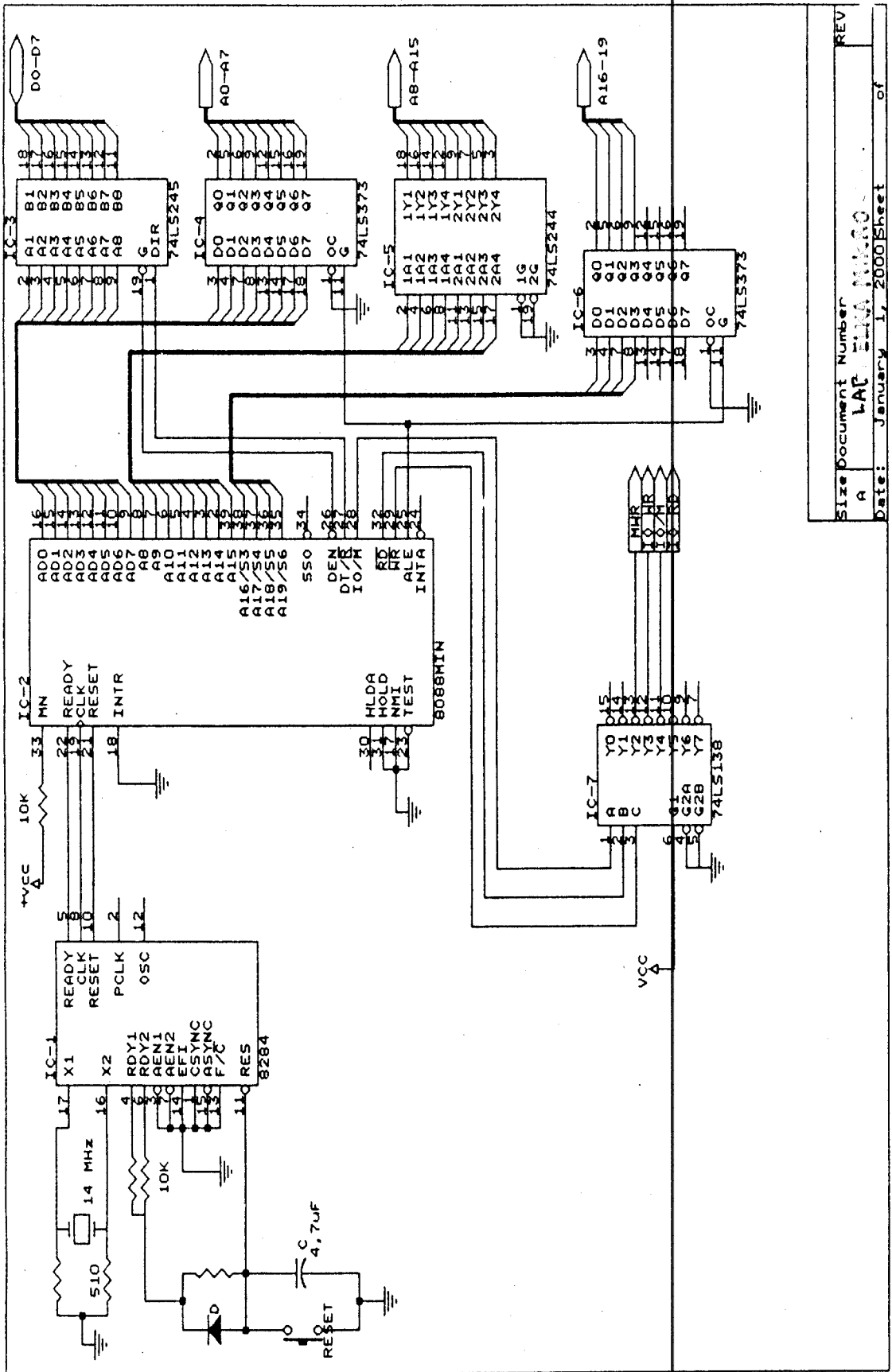
```
sys8088 ends
end
```



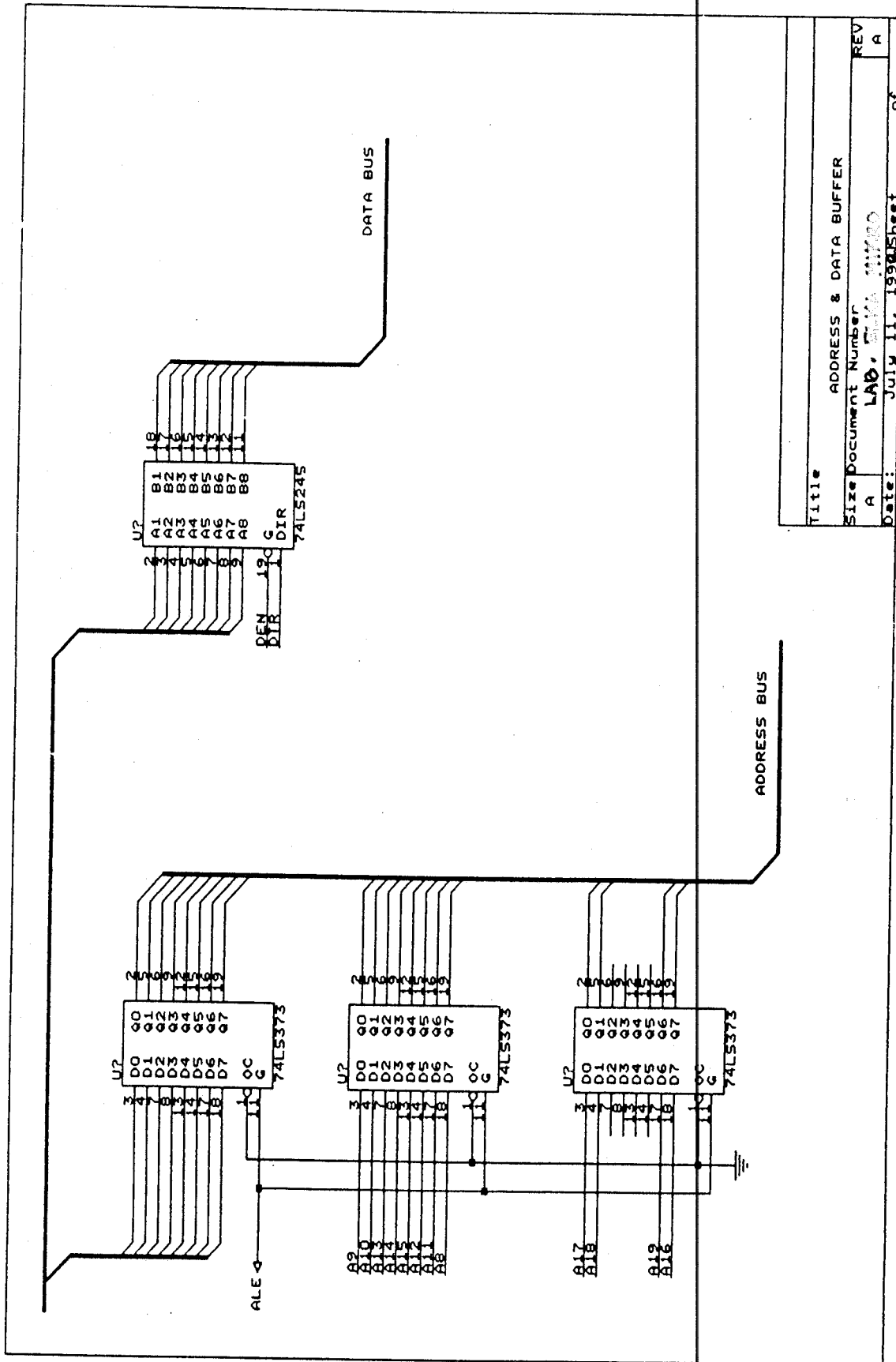
GAMBAR ALAT PENGATUR LAMPU LALU LINTAS
DIINTERFACEKAN KE KOMPUTER



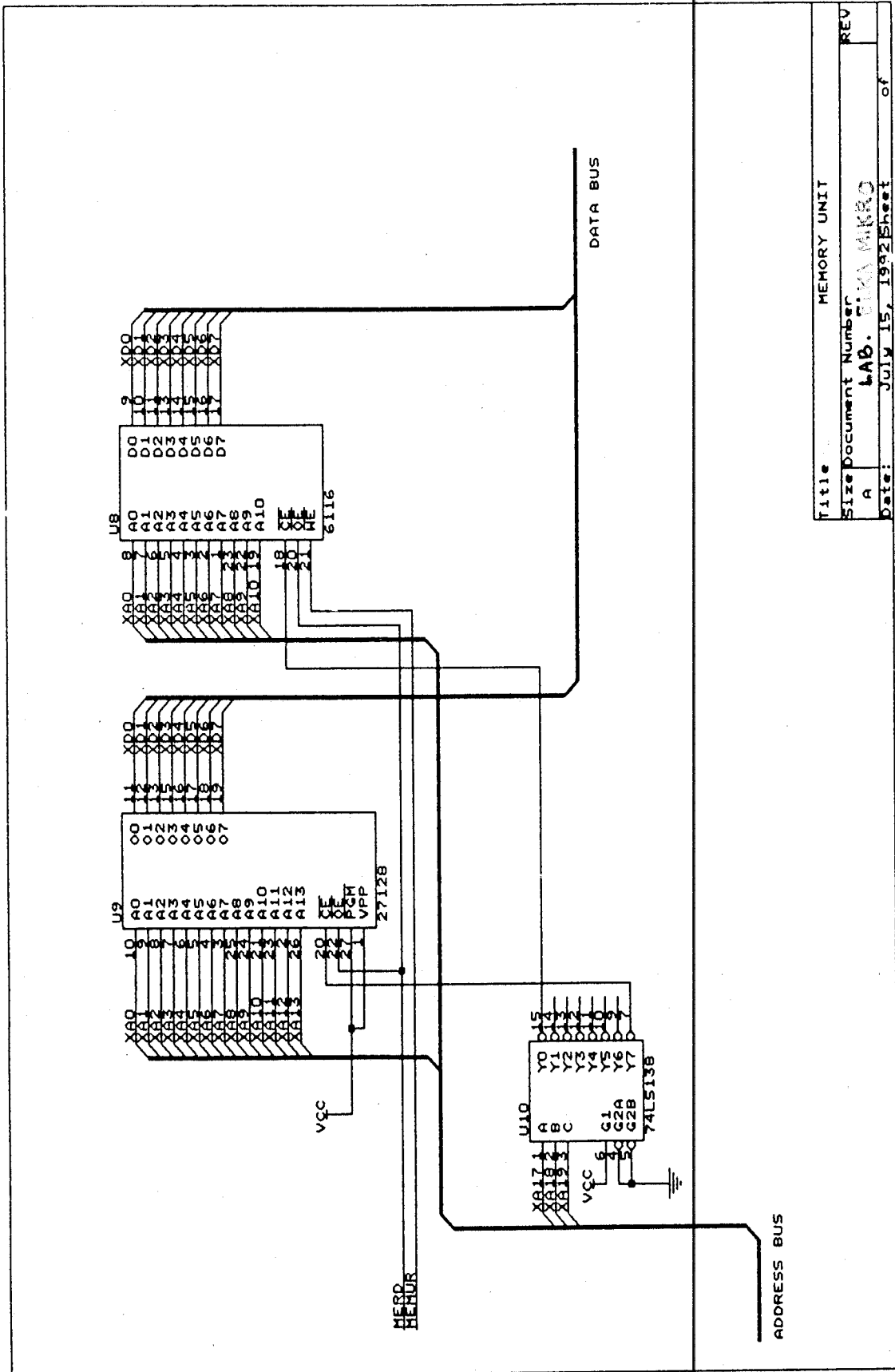
GAMBAR ALAT PENGATUR LALU LINTAS
DENGAN MIKROPROSESSOR 8088



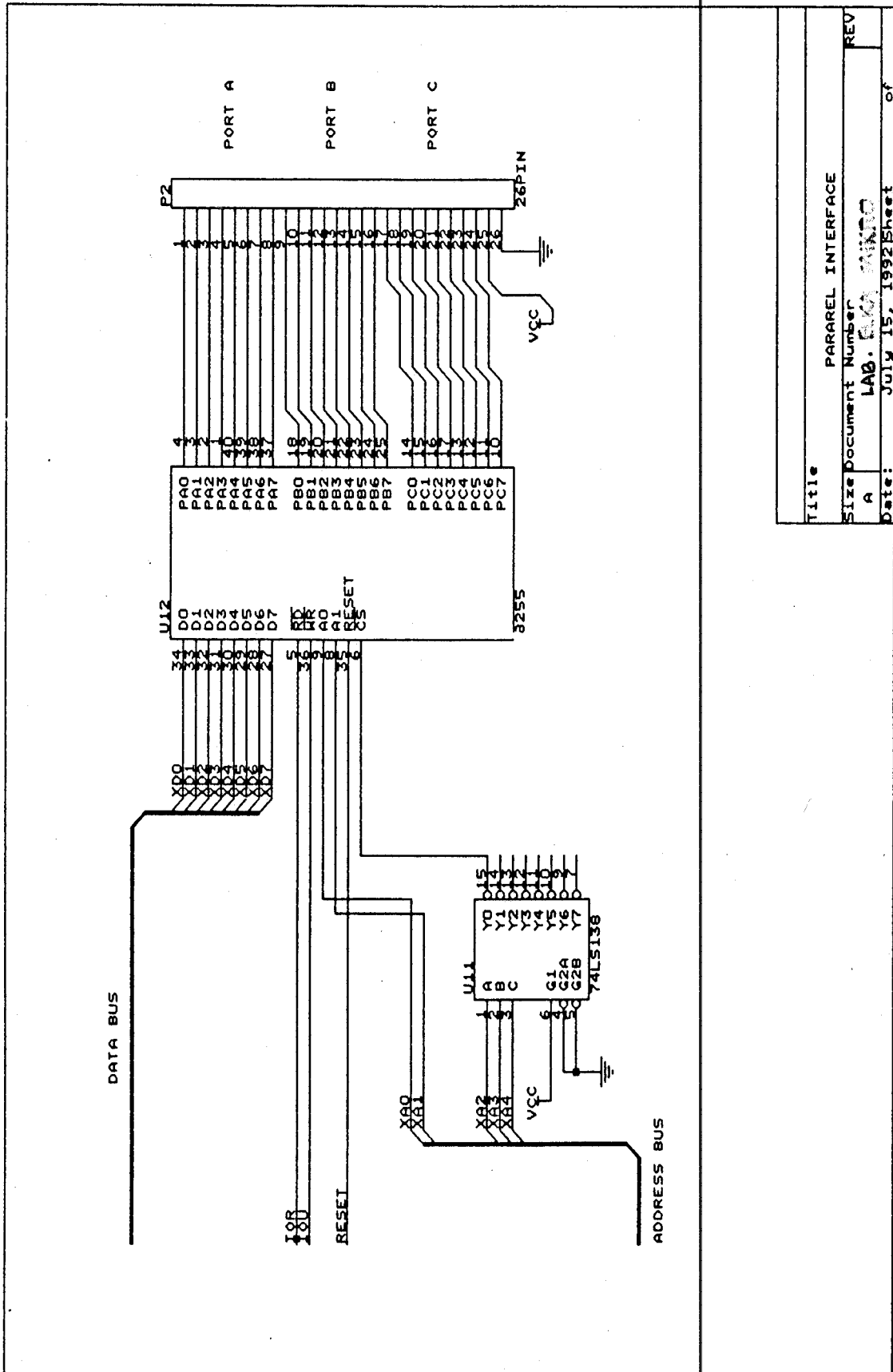
Size	Document Number	REV
A	LAC 2114 M100	
Date:	January 1, 2000	Sheet of



Title	ADDRESS & DATA BUFFER
Size Document Number	A
REV	A
Date:	JULY 11, 1994
Sheet	of



Title	MEMORY UNIT
Size	Document Number
A	LAB. SIMA MICRO
Date:	July 15, 1992
Sheet	of
REV	



Title	PARAREL INTERFACE
Size Document Number	A
LAB. FILE NUMBER	
Date:	JULY 15, 1992
Sheet	of

DATA SPOT SPEED DI JL.BLAURAN
TANGGAL SURVE:SELASA, 2 JUNI 1992

No	WAKTU (dt)	KEC (m/dt)	KEC (km/jam)	No	WAKTU (dt)	KEC (m/dt)	KEC (km/jam)	
1	8.50	5.882	21.176	51	7.90	6.329	22.785	
2	7.25	6.897	24.828	52	7.80	6.410	23.077	
3	7.50	6.667	24.000	53	7.50	6.667	24.000	
4	7.75	6.452	23.226	54	6.75	7.407	26.667	
5	7.75	6.452	23.226	55	7.75	6.452	23.226	
6	7.75	6.452	23.226	56	9.20	5.435	19.565	
7	8.50	5.882	21.176	57	7.00	7.143	25.714	
8	7.25	6.897	24.828	58	7.50	6.667	24.000	
9	7.75	6.452	23.226	59	8.30	6.024	21.687	
10	7.75	6.452	23.226	60	8.25	6.061	21.818	
11	7.75	6.452	23.226	61	8.30	6.024	21.687	
12	7.75	6.452	23.226	62	7.25	6.897	24.828	
13	7.50	6.667	24.000	63	8.50	5.882	21.176	
14	7.50	6.667	24.000	64	8.60	5.814	20.930	
15	7.50	6.667	24.000	65	8.50	5.882	21.176	
16	7.75	6.452	23.226	66	8.25	6.061	21.818	
17	7.75	6.452	23.226	67	7.80	6.410	23.077	
18	7.25	6.897	24.828	68	7.75	6.452	23.226	
19	7.50	6.667	24.000	69	7.50	6.667	24.000	
20	7.75	6.452	23.226	70	7.70	6.494	23.377	
21	8.00	6.250	22.500	71	8.20	6.098	21.951	
22	8.25	6.061	21.818	72	7.20	6.944	25.000	
23	7.75	6.452	23.226	73	8.00	6.250	22.500	
24	7.80	6.410	23.077	74	7.50	6.667	24.000	
25	7.75	6.452	23.226	75	8.50	5.882	21.176	
26	8.50	5.882	21.176	76	7.50	6.667	24.000	
27	8.25	6.061	21.818	77	7.50	6.667	24.000	
28	8.25	6.061	21.818	78	8.50	5.882	21.176	
29	7.00	7.143	25.714	79	7.50	6.667	24.000	
30	8.50	5.882	21.176	80	7.75	6.452	23.226	
31	7.75	6.452	23.226	81	7.55	6.623	23.841	
32	8.25	6.061	21.818	82	8.50	5.882	21.176	
33	8.50	5.882	21.176	83	7.25	6.897	24.828	
34	8.20	6.098	21.951	84	7.25	6.897	24.828	
35	7.75	6.452	23.226	85	7.50	6.667	24.000	
36	8.20	6.098	21.951	86	9.00	5.556	20.000	
37	8.20	6.098	21.951	87	8.25	6.061	21.818	
38	8.20	6.098	21.951	88	8.50	5.882	21.176	
39	8.65	5.780	20.809	89	9.00	5.556	20.000	
40	7.50	6.667	24.000	90	7.50	6.667	24.000	
41	7.75	6.452	23.226	91	7.75	6.452	23.226	
42	8.30	6.024	21.687	92	8.00	6.250	22.500	
43	8.25	6.061	21.818	93	8.00	6.250	22.500	
44	6.20	8.065	29.032	94	8.80	5.682	20.455	
45	7.50	6.667	24.000	95	850	5.882	21.176	
46	8.25	6.061	21.818	96	8.50	5.882	21.176	
47	7.50	6.667	24.000	97	7.50	6.667	24.000	
48	8.00	6.250	22.500	98	8.40	5.952	21.429	
49	8.25	6.061	21.818	99	7.60	6.579	23.684	
50	7.50	6.667	24.000	100	8.25	6.061	21.818	
RATA-RATA				KECEPATAN				6.355 : 22.877

DATA SPOT SPEED JL.PRABAN SISI UTARA
TANGGAL SURVE:RABU, 3 JUNI 1992

No	WAKTU (dt)	KEC (m/dt)	KEC (km/jam)	No	WAKTU (dt)	KEC (m/dt)	KEC (km/jam)	
1	7.70	6.494	23.377	51	6.90	7.246	26.087	
2	7.60	6.579	23.684	52	6.90	7.246	26.087	
3	8.00	6.250	22.500	53	7.40	6.757	24.324	
4	8.20	6.098	21.951	54	7.60	6.579	23.684	
5	7.50	6.667	24.000	55	7.20	6.944	25.000	
6	7.60	6.579	23.684	56	7.30	6.849	24.658	
7	8.00	6.250	22.500	57	7.90	6.329	22.785	
8	8.40	5.952	21.429	58	7.50	6.667	24.000	
9	7.20	6.944	25.000	59	6.90	7.246	26.087	
10	7.00	7.143	25.714	60	6.20	8.065	29.032	
11	8.90	5.618	20.225	61	7.20	6.944	25.000	
12	8.00	6.250	22.500	62	6.30	7.937	28.571	
13	7.50	6.667	24.000	63	6.80	7.353	26.471	
14	8.40	5.952	21.429	64	6.20	8.065	29.032	
15	8.60	5.814	20.930	65	6.90	7.246	26.087	
16	8.20	6.098	21.951	66	6.50	7.692	27.692	
17	7.20	6.944	25.000	67	7.50	6.667	24.000	
18	6.90	7.246	26.087	68	7.20	6.944	25.000	
19	6.20	8.065	29.032	69	7.90	6.329	22.785	
20	7.00	7.143	25.714	70	8.00	6.250	22.500	
21	6.70	7.463	26.866	71	7.20	6.944	25.000	
22	7.50	6.667	24.000	72	7.50	6.667	24.000	
23	7.00	7.143	25.714	73	7.30	6.849	24.658	
24	6.50	7.692	27.692	74	7.40	6.757	24.324	
25	7.10	7.042	25.352	75	6.90	7.246	26.087	
26	6.40	7.813	28.125	76	7.30	6.849	24.658	
27	6.20	8.065	29.032	77	7.50	6.667	24.000	
28	6.80	7.353	26.471	78	7.20	6.944	25.000	
29	6.00	8.333	30.000	79	6.70	7.463	26.866	
30	8.90	5.618	20.225	80	7.10	7.042	25.352	
31	7.10	7.042	25.352	81	7.90	6.329	22.785	
32	7.00	7.143	25.714	82	6.20	8.065	29.032	
33	6.10	8.197	29.508	83	8.10	6.173	22.222	
34	6.00	8.333	30.000	84	6.20	8.065	29.032	
35	8.10	6.173	22.222	85	7.50	6.667	24.000	
36	6.90	7.246	26.087	86	6.50	7.692	27.692	
37	7.90	6.329	22.785	87	7.10	7.042	25.352	
38	7.40	6.757	24.324	88	7.20	6.944	25.000	
39	6.30	7.937	28.571	89	7.50	6.667	24.000	
40	6.80	7.353	26.471	90	7.30	6.849	24.658	
41	7.20	6.944	25.000	91	6.90	7.246	26.087	
42	7.30	6.849	24.658	92	8.00	6.250	22.500	
43	6.90	7.246	26.087	93	7.20	6.944	25.000	
44	7.10	7.042	25.352	94	7.30	6.849	24.658	
45	7.90	6.329	22.785	95	7.50	6.667	24.000	
46	7.20	6.944	25.000	96	6.40	7.813	28.125	
47	6.90	7.246	26.087	97	7.40	6.757	24.324	
48	6.90	7.246	26.087	98	6.20	8.065	29.032	
49	7.30	6.849	24.658	99	6.50	7.692	27.692	
50	7.30	6.849	24.658	100	7.20	6.944	25.000	
RATA-RATA				KECEPATAN				6.987 : 25.153

DATA SPOT SPEED JL.KEDUNGORO
TANGGAL SURVE:KAMIS, 4 JUNI 1992

No	WAKTU (dt)	KEC (m/dt)	KEC (km/jam)	No	WAKTU (dt)	KEC (m/dt)	KEC (km/jam)	
1	7.70	6.494	23.377	51	6.90	7.246	26.087	
2	7.60	6.579	23.684	52	6.90	7.246	26.087	
3	8.00	6.250	22.500	53	7.40	6.757	24.324	
4	8.20	6.098	21.951	54	7.60	6.579	23.684	
5	7.50	6.667	24.000	55	7.20	6.944	25.000	
6	7.60	6.579	23.684	56	7.30	6.849	24.658	
7	8.00	6.250	22.500	57	7.90	6.329	22.785	
8	8.40	5.952	21.429	58	7.50	6.667	24.000	
9	7.20	6.944	25.000	59	6.90	7.246	26.087	
10	7.00	7.143	25.714	60	6.20	8.065	29.032	
11	8.90	5.618	20.225	61	7.20	6.944	25.000	
12	8.00	6.250	22.500	62	6.30	7.937	28.571	
13	7.50	6.667	24.000	63	6.80	7.353	26.471	
14	8.40	5.952	21.429	64	6.20	8.065	29.032	
15	8.60	5.814	20.930	65	6.90	7.246	26.087	
16	8.20	6.098	21.951	66	6.50	7.692	27.692	
17	7.20	6.944	25.000	67	7.50	6.667	24.000	
18	6.90	7.246	26.087	68	7.20	6.944	25.000	
19	6.20	8.065	29.032	69	7.90	6.329	22.785	
20	7.00	7.143	25.714	70	8.00	6.250	22.500	
21	6.70	7.463	26.866	71	7.20	6.944	25.000	
22	7.50	6.667	24.000	72	7.50	6.667	24.000	
23	7.00	7.143	25.714	73	7.30	6.849	24.658	
24	6.50	7.692	27.692	74	7.40	6.757	24.324	
25	7.10	7.042	25.352	75	6.90	7.246	26.087	
26	6.40	7.813	28.125	76	7.30	6.849	24.658	
27	6.20	8.065	29.032	77	7.50	6.667	24.000	
28	6.80	7.353	26.471	78	7.20	6.944	25.000	
29	6.00	8.333	30.000	79	6.70	7.463	26.866	
30	8.90	5.618	20.225	80	7.10	7.042	25.352	
31	7.10	7.042	25.352	81	7.90	6.329	22.785	
32	7.00	7.143	25.714	82	6.20	8.065	29.032	
33	6.10	8.197	29.508	83	8.10	6.173	22.222	
34	6.00	8.333	30.000	84	6.20	8.065	29.032	
35	8.10	6.173	22.222	85	7.50	6.667	24.000	
36	6.90	7.246	26.087	86	6.50	7.692	27.692	
37	7.90	6.329	22.785	87	7.10	7.042	25.352	
38	7.40	6.757	24.324	88	7.20	6.944	25.000	
39	6.30	7.937	28.571	89	7.50	6.667	24.000	
40	6.80	7.353	26.471	90	7.30	6.849	24.658	
41	7.20	6.944	25.000	91	6.90	7.246	26.087	
42	7.30	6.849	24.658	92	8.00	6.250	22.500	
43	6.90	7.246	26.087	93	7.20	6.944	25.000	
44	7.10	7.042	25.352	94	7.30	6.849	24.658	
45	7.90	6.329	22.785	95	7.50	6.667	24.000	
46	7.20	6.944	25.000	96	6.40	7.813	28.125	
47	6.90	7.246	26.087	97	7.40	6.757	24.324	
48	6.90	7.246	26.087	98	6.20	8.065	29.032	
49	7.30	6.849	24.658	99	6.50	7.692	27.692	
50	7.10	7.042	25.352	100	7.20	6.944	25.000	
RATA-RATA				KECEPATAN				6.987 : 25.153

DATA SPOT SPEED JL.TIDAR
TANGGAL SURVE:JUMAT, 5 JUNI 1992

No	WAKTU (dt)	KEC (m/dt)	KEC (km/jam)	No	WAKTU (dt)	KEC (m/dt)	KEC (km/jam)	
1	7.10	7.042	25.352	51	7.80	6.410	23.077	
2	6.30	7.937	28.571	52	7.20	6.944	25.000	
3	6.00	8.333	30.000	53	8.00	6.250	22.500	
4	7.50	6.667	24.000	54	8.00	6.250	22.500	
5	6.80	7.353	26.471	55	6.80	7.353	26.471	
6	6.40	7.813	28.125	56	8.10	6.173	22.222	
7	6.80	7.353	26.471	57	6.10	8.197	29.508	
8	7.80	6.410	23.077	58	7.90	6.329	22.785	
9	6.90	7.246	26.087	59	6.40	7.813	28.125	
10	8.10	6.173	22.222	60	7.50	6.667	24.000	
11	6.10	8.197	29.508	61	8.10	6.173	22.222	
12	6.20	8.065	29.032	62	5.70	8.772	31.579	
13	5.70	8.772	31.579	63	6.70	7.463	26.866	
14	7.00	7.143	25.714	64	7.60	6.579	23.684	
15	7.20	6.944	25.000	65	9.20	5.435	19.565	
16	6.30	7.937	28.571	66	7.30	6.849	24.658	
17	5.80	8.621	31.034	67	6.60	7.576	27.273	
18	6.20	8.065	29.032	68	5.20	9.615	34.615	
19	8.00	6.250	22.500	69	7.80	6.410	23.077	
20	5.20	9.615	34.615	70	7.20	6.944	25.000	
21	6.80	7.353	26.471	71	8.10	6.173	22.222	
22	5.30	9.434	33.962	72	6.20	8.065	29.032	
23	7.20	6.944	25.000	73	7.00	7.143	25.714	
24	5.50	9.091	32.727	74	5.80	8.621	31.034	
25	7.10	7.042	25.352	75	6.80	7.353	26.471	
26	8.10	6.173	22.222	76	6.30	7.937	28.571	
27	7.20	6.944	25.000	77	5.30	9.434	33.962	
28	7.20	6.944	25.000	78	7.20	6.944	25.000	
29	8.10	6.173	22.222	79	8.10	6.173	22.222	
30	5.70	8.772	31.579	80	6.78	7.375	26.549	
31	6.70	7.463	26.866	81	8.00	6.250	22.500	
32	7.60	6.579	23.684	82	6.60	7.576	27.273	
33	9.20	5.435	19.565	83	7.50	6.667	24.000	
34	7.30	6.849	24.658	84	6.40	7.813	28.125	
35	6.60	7.576	27.273	85	7.80	6.410	23.077	
36	5.20	9.615	34.615	86	8.50	5.882	21.176	
37	7.80	6.410	23.077	87	6.30	7.937	28.571	
38	7.20	6.944	25.000	88	7.40	6.757	24.324	
39	8.00	6.250	22.500	89	8.50	5.882	21.176	
40	8.00	6.250	22.500	90	7.30	6.849	24.658	
41	6.80	7.353	26.471	91	7.40	6.757	24.324	
42	8.10	6.173	22.222	92	6.80	7.353	26.471	
43	6.10	8.197	29.508	93	7.40	6.757	24.324	
44	7.90	6.329	22.785	94	7.20	6.944	25.000	
45	6.40	7.813	28.125	95	7.00	7.143	25.714	
46	7.50	6.667	24.000	96	6.40	7.813	28.125	
47	6.80	7.353	26.471	97	7.50	6.667	24.000	
48	7.40	6.757	24.324	98	7.60	6.579	23.684	
49	6.80	7.353	26.471	99	6.80	7.353	26.471	
50	6.10	8.197	29.508	100	8.00	6.250	22.500	
RATA-RATA				KECEPATAN				7.214 : 25.971

DATA SPOT SPEED JL.EMBONG MALANG
TANGGAL SURVE:SABTU, 6 JUNI 1992

No	WAKTU (dt)	KEC (m/dt)	KEC (km/jam)	No	WAKTU (dt)	KEC (m/dt)	KEC (km/jam)
1	8.10	6.173	22.222	51	7.80	6.410	23.077
2	8.30	6.024	21.687	52	8.00	6.250	22.500
3	8.50	5.882	21.176	53	8.00	6.250	22.500
4	7.80	6.410	23.077	54	8.00	6.250	22.500
5	9.00	5.556	20.000	55	8.50	5.882	21.176
6	9.00	5.556	20.000	56	8.10	6.173	22.222
7	9.00	5.556	20.000	57	9.50	5.263	18.947
8	8.00	6.250	22.500	58	7.90	6.329	22.785
9	8.10	6.173	22.222	59	7.80	6.410	23.077
10	8.10	6.173	22.222	60	8.90	5.618	20.225
11	9.10	5.495	19.780	61	8.10	6.173	22.222
12	9.10	5.495	19.780	62	7.50	6.667	24.000
13	9.00	5.556	20.000	63	8.50	5.882	21.176
14	8.00	6.250	22.500	64	8.00	6.250	22.500
15	7.20	6.944	25.000	65	9.20	5.435	19.565
16	9.00	5.556	20.000	66	7.30	6.849	24.658
17	8.00	6.250	22.500	67	7.50	6.667	24.000
18	8.90	5.618	20.225	68	8.50	5.882	21.176
19	8.00	6.250	22.500	69	7.80	6.410	23.077
20	7.50	6.667	24.000	70	8.00	6.250	22.500
21	8.00	6.250	22.500	71	8.10	6.173	22.222
22	7.79	6.418	23.107	72	9.00	5.556	20.000
23	7.80	6.410	23.077	73	7.90	6.329	22.785
24	8.00	6.250	22.500	74	7.50	6.667	24.000
25	8.50	5.882	21.176	75	7.50	6.667	24.000
26	8.10	6.173	22.222	76	7.80	6.410	23.077
27	8.50	5.882	21.176	77	9.00	5.556	20.000
28	9.00	5.556	20.000	78	7.20	6.944	25.000
29	8.30	6.024	21.687	79	8.10	6.173	22.222
30	7.80	6.410	23.077	80	7.80	6.410	23.077
31	9.90	5.051	18.182	81	8.00	6.250	22.500
32	7.60	6.579	23.684	82	9.00	5.556	20.000
33	9.20	5.435	19.565	83	7.50	6.667	24.000
34	9.00	5.556	20.000	84	7.50	6.667	24.000
35	8.00	6.250	22.500	85	7.80	6.410	23.077
36	8.10	6.173	22.222	86	8.50	5.882	21.176
37	7.80	6.410	23.077	87	9.00	5.556	20.000
38	8.00	6.250	22.500	88	9.10	5.495	19.780
39	8.00	6.250	22.500	89	8.50	5.882	21.176
40	8.00	6.250	22.500	90	9.70	5.155	18.557
41	9.00	5.556	20.000	91	7.40	6.757	24.324
42	8.10	6.173	22.222	92	7.50	6.667	24.000
43	9.00	5.556	20.000	93	7.40	6.757	24.324
44	9.00	5.556	20.000	94	9.00	5.556	20.000
45	8.90	5.618	20.225	95	7.00	7.143	25.714
46	8.90	5.618	20.225	96	8.50	5.882	21.176
47	7.90	6.329	22.785	97	7.50	6.667	24.000
48	8.50	5.882	21.176	98	7.10	7.042	25.352
49	9.90	5.051	18.182	99	8.00	6.250	22.500
50	8.00	6.250	22.500	100	7.50	6.667	24.000
RATA-RATA				KECEPATAN : 6.100 : 21.959			

DATA VOLUME KENDARAAN DI TEMPAT LOKASI PENELITIAN SELAMA
14 JAM MULAI JAM 06.00-20.00 TANGGAL 9-DESEMBER 1989
PENELITIAN LABORATORIUM TEKNIK SIPIL ITS

- DARI JL.KEDUNGORO KE JL.BLAURAN

o MOBIL = 8100
o TAXI = 1471
o TRUK = 62
o BUS = 26
o SEPEDA MOTOR = 15 657

- DARI JL.KEDUNGORO KE JL.TIDAR

o MOBIL = 1170
o TAXI = 30
o TRUK = 21
o BUS = 0
o SEPEDA MOTOR = 548

- DARI JL.EMBONG MALANG KE JL.TIDAR

o MOBIL = 3572
o TAXI = 889
o TRUK = 18
o BUS = 348
o SEPEDA MOTOR = 23158

- DARI JL.TIDAR KE KEDUNGORO

o MOBIL = 649
o TAXI = 251
o TRUK = 49
o BUS = 39
o SEPEDA MOTOR = 1364

- DARI JL.TIDAR KE JL.BLAURAN

o MOBIL = 3415
o TAXI = 854
o TRUK = 28
o BUS = 443
o SEPEDA MOTOR = 25353

- DARI JL.BLAURAN KE JL KRANGGAN

o MOBIL = 8128
o TAXI = 451
o TRUK = 0
o BUS = 0
o SEPEDA MOTOR = 5430

- DARI JL.BLAURAN KE JL.BUBUTAN

o MOBIL = 14122
o TAXI = 3314

- o TRUK = 295
- o BUS = 790
- o SEPEDA MOTOR = 25353

- DARI JL. BLAURAN KE JL. PRABAN

- o MOBIL = 9435
- o TAXI = 1295
- o TRUK = 75
- o BUS = 7
- o SEPEDA MOTOR = 8879

- DARI JL. PRABAN KE BUBUTAN

- o MOBIL = 7998
- o TAXI = 1803
- o TRUK = 72
- o BUS = 7
- o SEPEDA MOTOR = 8879

- DARI JL. KRANGGAN

- o MOBIL = 2765
- o TAXI = 30
- o TRUK = 2
- o BUS = 1335
- o SEPEDA MOTOR = 9205

DATA BANYAKNYA BUS DI SURABAYA

TAHUN (X)	BUS (Y)	XY	XX	YY
1986	2403	4772358	3944196	5774409
1987	2483	4933721	3948169	6165289
1988	2701	5369588	3952144	7295401
1989	2793	5555277	3956121	7800849
1990	2850	5671500	3960100	8122500
9940	13230	26302444	19760730	35158448
		B = 120.4		
		A = -236709.2		

JUMLAH MOBIL DI KODYA SURABAYA

TAHUN (X)	MOBIL(Y)	XY	XX	YY
1986	58723	116623878	3944196	3448390729
1987	76089	151188843	3948169	5789535921
1988	83641	166278308	3952144	6995816881
1989	86652	172350828	3956121	7508569104
1990	98932	196874680	3960100	9787540624
9940	404037	803316537	19760730	33529853259
		B = 9098.1		
		A = -18006215.4		

PRODUK DOMESTIK REGIONAL BRUTO PER KAPITA DI KODYA SURABAYA

TAHUN(X)	PDRB(Y)	XY	XX	YY
1986	806911.2365	1602525715.7	3944196	651105743590
1987	871184.4218	1731043446.1	3948169	758962296787
1988	876496.8029	1742475644.2	3952144	768246645494
1989	1091656.5550	2171304887.9	3956121	1191714034074
1990	1193894.0000	2375849060.0	3960100	1425382883236
9940	4840143.0162	9623198753.9	19760730	4795411603181
		B =	99443.76602	
		A =	-196726178.24	

JUMLAH TAXI DI KODYA SURABAYA

TAHUN (X)	BUS (Y)	XY	XX	YY
1986	3076	6108936	3944196	9461776
1987	3247	6451789	3948169	10543009
1988	3496	6950048	3952144	12222016
1989	3562	7084818	3956121	12687844
1990	3590	7144100	3960100	12888100
9940	16971	33739691	19760730	57802745
		B =	134.3	
		A =	-263594.2	

JUMLAH TRUK DI KODYA SURABAYA

TAHUN (X)	TRUK (Y)	XY	XX	YY
1986	42749	84899514	3944196	1827477001
1987	45211	89834257	3948169	2044034521
1988	46981	93398228	3952144	2207214361
1989	49345	98147205	3956121	2434929025
1990	53121	105710790	3960100	2821840641
9940	237407	471989994	19760730	11335495549
		B =	2487.8	
		A =	-4898265	

TABLE 24-5
Types of Detectors

Detector	Measuring Capability					Method of Operation	Advantages	Disadvantages
	Count	Presence	Speed	Occupancy	Queue Length			
Pressure	Yes	No	No	No	No	Weight of vehicle causes closure of metallic contacts to complete a circuit	<ol style="list-style-type: none"> 1. Well-defined detection zone 2. Rugged construction 3. Reliable 4. Relatively inexpensive 5. Capable of detecting all moving vehicles regardless of speed 6. Low maintenance and easy to repair 	<ol style="list-style-type: none"> 1. Counts axles which yields poor count accuracy 2. Does not measure presence 3. Installation may disrupt traffic for excessive period of time 4. Major resurfacing will render it inoperative 5. Susceptible to damage by snowplow 6. Cannot be easily relocated
Magnetic, non-directional	Yes	No	No	No	No	Vehicle passage over wire coil embedded in roadway disturbs earth's lines of flux passing through coil and induces a voltage in the coil; voltage is amplified by high-gain amplifier to operate detector relay	<ol style="list-style-type: none"> 1. Under roadway location and not subject to damage 2. Relatively easy to install 3. Does not necessitate closing of traffic lanes 4. Relative ease of relocation 5. Low maintenance 	<ol style="list-style-type: none"> 1. Nondirectional 2. Difficult to set detection zone 3. Subject to false calls where located near large dc lines 4. Cannot detect presence
Magnetic, directional (two-coil version)	Yes	No	No	No	No	Same method of operation as nondirectional magnetic detector	<ol style="list-style-type: none"> 1. Directional 2. Not affected by dc lines in vicinity 3. Well-defined detection zone 4. Low maintenance 5. Under roadway location and not subject to damage 	<ol style="list-style-type: none"> 1. Requires closing of traffic lane for installation 2. More expensive than nondirectional magnetic detector 3. Cannot detect presence 4. Cannot be easily relocated
Magnetometer	Yes	Yes	No	Yes	No	Similar method of operation as nondirectional and directional magnetic detectors; makes use of small cylindrical sensing head that is placed below pavement surface	<ol style="list-style-type: none"> 1. Relatively easy to install 2. Capable of measuring count or presence 3. Reliable 4. Not affected by dc lines in vicinity 5. Under roadway location and not subject to damage 6. Relative ease of relocation 	<ol style="list-style-type: none"> 1. Requires closing of traffic lane for installation 2. More expensive than nondirectional 3. May double count some vehicles due to magnetic material distribution 4. Poorly defined detection zone
Loop, manual-tuned	Yes	Yes	Yes	Yes	Yes	Vehicle passage cuts magnetic lines of flux that are generated around the loop thereby increasing or decreasing the inductance so that a change is detected and transmitted to an amplifying circuit	<ol style="list-style-type: none"> 1. Size and shape of detection zone can be easily set by size of loop 2. Excellent presence detector 3. Capable of measuring all traffic parameters 4. Relatively easy to install 5. Relatively inexpensive to abandon loop and reuse amplifier at new location 6. Capable of detecting small vehicles 7. Under roadway location and not subject to damage 	<ol style="list-style-type: none"> 1. Cost of installation may be excessive 2. Requires closing of traffic lane or lanes for short periods of time 3. Difficult to tune in order to detect small and large vehicles
Loop, self-tuning	Yes	Yes	Yes	Yes	Yes	Same method of operation as manual-tuned loop	Same advantages as manual-tuned loop except no initial or periodic tuning required	Same disadvantages as manual-tuned loop except that several poorly shielded amplifiers located in same cabinet may interfere with each other and response time may be slower than manual-tuned loop

TABLE 24-5 (continued)

Detector	Measuring Capability					Method of Operation	Advantages	Disadvantages
	Count	Presence	Speed	Occupancy	Queue Length			
Radar	Yes	No	Yes	No	No	Passage of vehicle reflects radar microwaves (Doppler principle) back to antenna to operate detector relay	<ol style="list-style-type: none"> 1. Immune to electromagnetic interference 2. Does not necessitate closing of traffic lanes to install 	<ol style="list-style-type: none"> 1. Relatively expensive to purchase and install, particularly if existing installation available for use 2. Requires FCC license to operate 3. Requires experienced personnel for installation and maintenance 4. Does not measure presence
Sonic, pulsed	Yes	Yes	Yes	Yes	Yes	Emits bursts of energy at a rate of approximately 20 times per second; vehicle reduces wavelength resulting in the return signal arriving when receiver is open	<ol style="list-style-type: none"> 1. Does not necessitate closing of traffic lanes to install 2. Does not require FCC license to operate 3. Can be used at locations with unstable pavement 4. Can classify vehicle by height 	<ol style="list-style-type: none"> 1. Same as 1 for radar detector 2. Somewhat inaccurate due to conical detection zone and wide variations in vehicle configurations and heights 3. Nondirectional 4. Sensitive to environmental conditions 5. Somewhat inaccurate under congested conditions
Sonic, continuous wave	Yes	No	Yes	No	No	Operates on Doppler principle—same as radar	<ol style="list-style-type: none"> 1. Same as 1, 2, and 3 for pulsed sonic detector plus improved accuracy for speed measurement 	<ol style="list-style-type: none"> 1. Same as 1, 4, 5 for pulsed sonic detector 2. Cannot detect presence
Radio frequency	Special application for UTCS Bus Priority System					Bus transmitter sends signal to loop in pavement which relays signal to receiver unit; transmits information on two separate radio frequencies	<ol style="list-style-type: none"> 1. Can be used to select special vehicles 2. Accurate and reliable 	<ol style="list-style-type: none"> 1. Relatively expensive to install
Light emission photo electric	Yes	Yes	Yes	Yes	No	Passage of vehicle between light emitter and photoelectric cell interrupts transmitted beam which operates a detector relay	<ol style="list-style-type: none"> 1. Accurate for vehicle passage in a single lane 2. Most suitable for conditions of uniform light 	<ol style="list-style-type: none"> 1. Inaccurate for detection of more than one traffic lane
Infrared, interrupted beam	Yes	Yes	Yes	Yes	No	Same as photoelectric detector using infrared part of spectrum	Same as photoelectric	Same as photoelectric
Infrared, reflected beam	Yes	Yes	Yes	Yes	No	Overhead transmitter-receiver notes vehicle passage by change in reflectivity between vehicle and pavement	<ol style="list-style-type: none"> 1. Most suitable for conditions of uniform light 	<ol style="list-style-type: none"> 1. Expensive 2. Sensitive to ambient light and color of pavement 3. Sensitive to weather conditions 4. Inaccurate due to reflectivity difference
High-intensity light	Yes	No	No	No	No	High-intensity light emitted from a device mounted on a priority vehicle is received by a light-sensitive detector indicating its presence	<ol style="list-style-type: none"> 1. Provides a means to recognize selected vehicles 	<ol style="list-style-type: none"> 1. Expensive

SOURCE: *Traffic Control Systems Handbook*, U.S. Department of Transportation, Federal Highway Administration, U.S. Government Printing Office, Washington, D.C., 1976.



PRELIMINARY

IAPX 88/10 8-BIT HMOS MICROPROCESSOR 8088/8088-2

- 8-Bit Data Bus Interface
- 16-Bit Internal Architecture
- Direct Addressing Capability to 1 Mbyte of Memory
- Direct Software Compatibility with IAPX 88/10 (8088 CPU)
- 14-Word by 16-Bit Register Set with Symmetrical Operations
- 24 Operand Addressing Modes
- Byte, Word, and Block Operations
- 8-Bit and 16-Bit Signed and Unsigned Arithmetic in Binary or Decimal, Including Multiply and Divide
- Compatible with 8155-2, 8755A-2 and 8185-2 Multiplexed Peripherals
- Two Clock Rates:
5 MHz for 8088
8 MHz for 8088-2
- Available in EXPRESS
 - Standard Temperature Range
 - Extended Temperature Range

The Intel® IAPX 88/10 is a new generation, high performance microprocessor implemented in N-channel, depletion load, silicon gate technology (HMOS), and packaged in a 40-pin CerDIP package. The processor has attributes of both 8- and 16-bit microprocessors. It is directly compatible with IAPX 86/10 software and 8080/8085 hardware and peripherals.

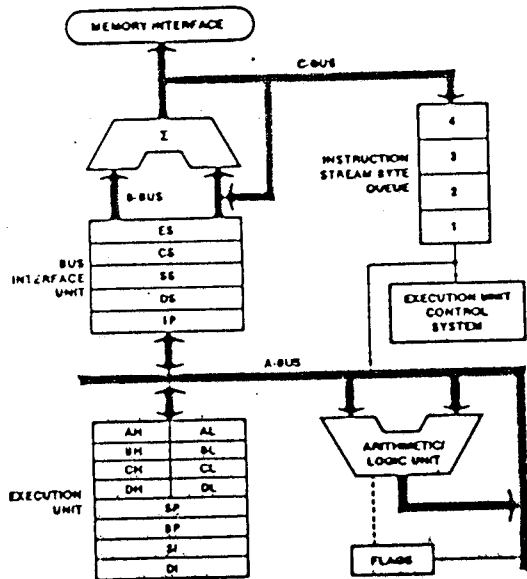


Figure 1. IAPX 88/10 CPU Functional Block Diagram

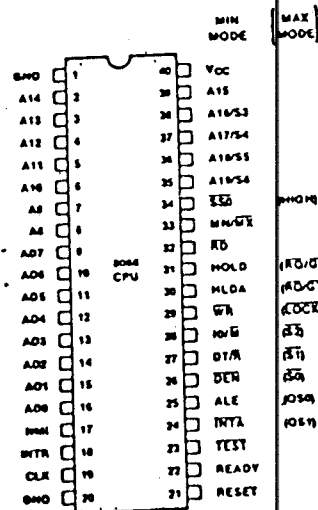


Figure 2. IAPX 88/10 Pin Configuration

intel

IAPX 88/10

PRELIMINARY

Table 1. Pin Description

The following pin function descriptions are for 8088 systems in either minimum or maximum mode. The "local bus" in these descriptions is the direct multiplexed bus interface connection to the 8088 (without regard to additional bus buffers).

Symbol	Pin No.	Type	Name and Function															
AD7-AD0	9-16	VO	Address Data Bus: These lines constitute the time multiplexed memory/IO address (T1) and data (T2, T3, Tw, and T4) bus. These lines are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge".															
A15-A8	2-8, 30	O	Address Bus: These lines provide address bits 8 through 15 for the entire bus cycle (T1-T4). These lines do not have to be latched by ALE to remain valid. A15-A8 are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge".															
A19/S6, A18/S5, A17/S4, A16/S3	35-38	O	Address/Status: During T1, these are the four most significant address lines for memory operations. During IO operations, these lines are LOW. During memory and IO operations, status information is available on these lines during T2, T3, Tw, and T4. S6 is always low. The status of the interrupt enable flag bit (S5) is updated at the beginning of each clock cycle. S4 and S3 are encoded as shown. <table border="1" data-bbox="1029 913 1241 1008"> <thead> <tr> <th>S4</th> <th>S3</th> <th>Segment</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>System Bus</td> </tr> <tr> <td>0</td> <td>1</td> <td>Cache</td> </tr> <tr> <td>1</td> <td>0</td> <td>Cache or Memory</td> </tr> <tr> <td>1</td> <td>1</td> <td>Memory</td> </tr> </tbody> </table> <p>This information indicates which segment register is presently being used for data accessing. These lines float to 3-state OFF during local bus "hold acknowledge".</p>	S4	S3	Segment	0	0	System Bus	0	1	Cache	1	0	Cache or Memory	1	1	Memory
S4	S3	Segment																
0	0	System Bus																
0	1	Cache																
1	0	Cache or Memory																
1	1	Memory																
RD	32	O	Read: Read strobe indicates that the processor is performing a memory or IO read cycle, depending on the state of the I/O pin or S2. This signal is used to read devices which reside on the 8088 local bus. RD is active LOW during T2, T3 and Tw of any read cycle, and is guaranteed to remain HIGH in T1 until the 8088 local bus has floated. This signal floats to 3-state OFF in "hold acknowledge".															
READY	22	I	READY: is the acknowledgement from the addressed memory or IO device that it will complete the data transfer. The RDY signal from memory or IO is synchronized by the 8284 clock generator to form READY. This signal is active HIGH. The 8088 READY input is not synchronized. Correct operation is not guaranteed if the set up and hold times are not met.															
INTR	18	I	Interrupt Request: is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.															
TEST	23	I	TEST: input is examined by the "wait for test" instruction. If the TEST input is LOW, execution continues, otherwise the processor waits in an "idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.															
NMI	17	I	Non-Maskable Interrupt: is an edge triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.															

Table 1. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
RESET	21	I	RESET: causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the Instruction set description, when RESET returns LOW. RESET is internally synchronized.
CLK	19	I	Clock: provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
V _{CC}	40		V _{CC} : is the +5V ±10% power supply pin.
GND	1, 20		GND: are the ground pins.
MN/MX	33	I	Minimum/Maximum: indicates what mode the processor is to operate in. The two modes are discussed in the following sections.

The following pin function descriptions are for the 8088 minimum mode (i.e., MN/MX = V_{CC}). Only the pin functions which are unique to minimum mode are described; all other pin functions are as described above.

IO/M	28	O	Status Line: is an inverted maximum mode $\overline{S2}$. It is used to distinguish a memory access from an I/O access. IO/M becomes valid in the T4 preceding a bus cycle and remains valid until the final T4 of the cycle (VO=HIGH, M=LOW). IO/M floats to 3-state OFF in local bus "hold acknowledge".
WR	29	O	Write: strobe indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the IO/M signal. WR is active for T2, T3, and Tw of any write cycle. It is active LOW, and floats to 3-state OFF in local bus "hold acknowledge".
\overline{INTA}	24	O	INTA: is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T2, T3, and Tw of each interrupt acknowledge cycle.
ALE	25	O	Address Latch Enable: is provided by the processor to latch the address into the 8282/8283 address latch. It is a HIGH pulse active during clock low of T1 of any bus cycle. Note that ALE is never floated.
DT/R	27	O	Data TransmI/Receive: is needed in a minimum system that desires to use an 8286/8287 data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically, DT/R is equivalent to $\overline{S1}$ in the maximum mode, and its timing is the same as for IO/M (T=HIGH, R=LOW). This signal floats to 3-state OFF in local "hold acknowledge".
\overline{DEN}	26	O	Data Enable: is provided as an output enable for the 8286/8287 in a minimum system which uses the transceiver. \overline{DEN} is active LOW during each memory and I/O access, and for \overline{INTA} cycles. For a read or \overline{INTA} cycle, it is active from the middle of T2 until the middle of T4, while for a write cycle, it is active from the beginning of T2 until the middle of T4. \overline{DEN} floats to 3-state OFF during local bus "hold acknowledge".
HOLD, HLDA	30,31	I, O	HOLD: indicates that another master is requesting a local bus "hold". To be acknowledged, HOLD must be active HIGH. The processor receiving the "hold" request will issue HLDA (HIGH) as an acknowledgement, in the middle of a T4 or T1 clock cycle. Simultaneous with the issuance of HLDA the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor lowers HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines. Hold is not an asynchronous input. External synchronization should be provided if the system cannot otherwise guarantee the set up time.
SSO	34	O	Status line: is logically equivalent to $\overline{S0}$ in the maximum mode. The combination of SSO, IO/M and DT/R allows the system to completely decode the current bus cycle status.

IO/M	DT/R	SSO	Decode/Function
0	0	0	Memory Access
0	0	1	Memory Access
0	1	0	Memory Access
0	1	1	Memory Access
1	0	0	I/O Access
1	0	1	I/O Access
1	1	0	I/O Access
1	1	1	I/O Access

Table 1. Pin Description (Continued)

The following pin function descriptions are for the 8088, 8228 system in maximum mode (i.e., MNIMX=GND.) Only the pin functions which are unique to maximum mode are described; all other pin functions are as described above.


Symbol	Pin No.	Type	Name and Function
$\overline{S2}, \overline{S1}, \overline{S0}$	26-28	O	<p>Status: Is active during clock high of T4, T1, and T2, and is returned to the passive state (1,1,1) during T3 or during Tw when READY is HIGH. This status is used by the 8288 bus controller to generate all memory and IO access control signals. Any change by $\overline{S2}$, $\overline{S1}$, or $\overline{S0}$ during T4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T3 or Tw is used to indicate the end of a bus cycle.</p> <p>These signals float to 3-state OFF during "hold acknowledge". During the first clock cycle after RESET becomes active, these signals are active HIGH. After this first clock, they float to 3-state OFF.</p> 
$\overline{RQ}/\overline{GT0}$, $\overline{RQ}/\overline{GT1}$	30, 31	VO	<p>Request/Grant: pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with $\overline{RQ}/\overline{GT0}$ having higher priority than $\overline{RQ}/\overline{GT1}$. $\overline{RQ}/\overline{GT}$ has an internal pull-up resistor, so may be left unconnected. The request/grant sequence is as follows (See Figure 8):</p> <ol style="list-style-type: none"> 1. A pulse of one CLK wide from another local bus master indicates a local bus request ("hold") to the 8088 (pulse 1). 2. During a T4 or T1 clock cycle, a pulse one clock wide from the 8088 to the requesting master (pulse 2), indicates that the 8088 has allowed the local bus to float and that it will enter the "hold acknowledge" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "hold acknowledge". The same rules as for HOLD/HOLDA apply as for when the bus is released. 3. A pulse one CLK wide from the requesting master indicates to the 8088 (pulse 3) that the "hold" request is about to end and that the 8088 can reclaim the local bus at the next CLK. The CPU then enters T4. <p>Each master-master exchange of the local bus is a sequence of three pulses. There must be one idle CLK cycle after each bus exchange. Pulses are active LOW.</p> <p>If the request is made while the CPU is performing a memory cycle, it will release the local bus during T4 of the cycle when all the following conditions are met:</p> <ol style="list-style-type: none"> 1. Request occurs on or before T2. 2. Current cycle is not the low bit of a word. 3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence. 4. A locked instruction is not currently executing. <p>If the local bus is idle when the request is made the two possible events will follow:</p> <ol style="list-style-type: none"> 1. Local bus will be released during the next clock. 2. A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.

Table 1. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function															
LOCK	29	O	<p>LOCK: Indicates that other system bus masters are not to gain control of the system bus while LOCK is active (LOW). The LOCK signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and floats to 3-state off in "hold acknowledge".</p>															
QS1, QS0	24, 25	O	<p>Queue Status: provide status to allow external tracking of the internal 8086 instruction queue.</p> <p>The queue status is valid during the CLK cycle after which the queue operation is performed.</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;"> <table border="1"> <tr> <td>QS1</td> <td>QS0</td> <td>Queue Status</td> </tr> <tr> <td>0</td> <td>0</td> <td>Queue empty</td> </tr> <tr> <td>0</td> <td>1</td> <td>Queue full</td> </tr> <tr> <td>1</td> <td>0</td> <td>Queue full</td> </tr> <tr> <td>1</td> <td>1</td> <td>Queue full</td> </tr> </table> </div>	QS1	QS0	Queue Status	0	0	Queue empty	0	1	Queue full	1	0	Queue full	1	1	Queue full
QS1	QS0	Queue Status																
0	0	Queue empty																
0	1	Queue full																
1	0	Queue full																
1	1	Queue full																
—	34	O	Pin 34 is always high in the maximum mode.															

FUNCTIONAL DESCRIPTION

Memory Organization

The processor provides a 20-bit address to memory which locates the byte being referenced. The memory is organized as a linear array of up to 1 million bytes, addressed as 00000(H) to FFFFF(H). The memory is logically divided into code, data, extra data, and stack segments of up to 64K bytes each, with each segment falling on 16-byte boundaries. (See Figure 3.)

All memory references are made relative to base addresses contained in high speed segment registers. The segment types were chosen based on the addressing needs of programs. The segment register to be selected is automatically chosen according to the rules of the following table. All information in one segment type share the same logical attributes (e.g. code or data). By structuring memory into relocatable areas of similar characteristics and by automatically selecting segment registers, programs are shorter, faster, and more structured.

Word (16-bit) operands can be located on even or odd address boundaries. For address and data operands, the least significant byte of the word is stored in the lower valued address location and the most significant byte in

the next higher address location. The BIU will automatically execute two fetch or write cycles for 16-bit operands.

Certain locations in memory are reserved for specific CPU operations. (See Figure 4.) Locations from addresses FFFFF(H) through FFFF(H) are reserved for operations including a jump to the initial system initialization routine. Following RESET, the CPU will always begin execution at location FFFF(H) where the jump must be located. Locations 0000(H) through 00FF(H) are reserved for interrupt operations. Four-byte pointers consisting of a 16-bit segment address and a 16-bit offset address direct program flow to one of the 256 possible interrupt service routines. The pointer elements are assumed to have been stored at their respective places in reserved memory prior to the occurrence of interrupts.

Minimum and Maximum Modes

The requirements for supporting minimum and maximum 8088 systems are sufficiently different that they cannot be done efficiently with 40 uniquely defined pins. Consequently, the 8088 is equipped with a strap pin (MN/MX) which defines the system configuration. The definition of a certain subset of the pins changes, dependent on the condition of the strap pin. When the MN/MX pin is strapped to GND, the 8088 defines pins 24 through 31 and 34 in maximum mode. When the MN/MX pin is strapped to VCC, the 8088 generates bus control signals itself on pins 24 through 31 and 34.

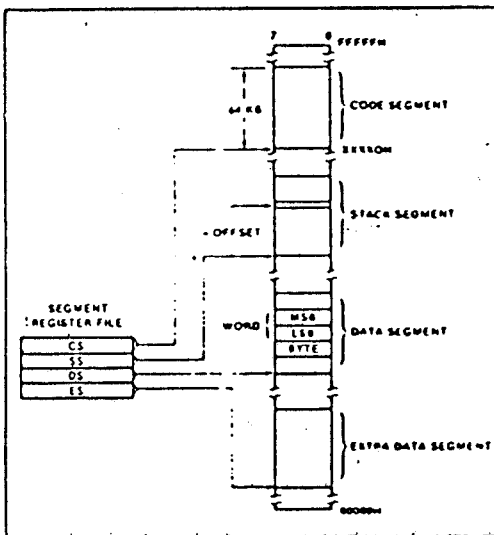


Figure 3. Memory Organization

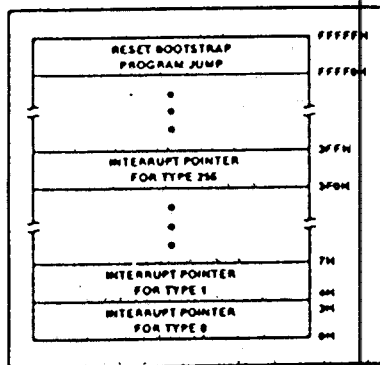


Figure 4. Reserved Memory Locations

Memory Reference Need	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic with all instruction prefetch.
Stack	STACK (SS)	All stack pushes and pops. Memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references when: relative to stack, destination of string operation, or explicitly overridden.
External (Global) Data	EXTRA (ES)	Destination of string operations: Explicitly selected using a segment override.

intel

IAPX 88/10

PRELIMINARY

The minimum mode 8088 can be used with either a multiplexed or demultiplexed bus. The multiplexed bus configuration is compatible with the MCS-85™ multiplexed bus peripherals (8155, 8156, 8355, 8755A, and 8185). This configuration (See Figure 5) provides the user with a minimum chip count system. This architecture provides the 8088 processing power in a highly integrated form.

The demultiplexed mode requires one latch (for 64K addressability) or two latches (for a full megabyte of addressing). A third latch can be used for buffering if the address bus loading requires it. An 8286 or 8287 transceiver can also be used if data bus buffering is required. (See Figure 6.) The 8088 provides \overline{DEN} and $\overline{DT/R}$ to con-

trol the transceiver, and ALE to latch the addresses. This configuration of the minimum mode provides the standard demultiplexed bus structure with heavy bus buffering and relaxed bus timing requirements.

The maximum mode employs the 8288 bus controller. (See Figure 7.) The 8288 decodes status lines $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$, and provides the system with all bus control signals. Moving the bus control to the 8288 provides better source and sink current capability to the control lines, and frees the 8088 pins for extended large system features. Hardware lock, queue status, and two request grant interfaces are provided by the 8088 in maximum mode. These features allow co-processors in local bus and remote bus configurations.

8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel® Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- Reduces System Package Count
- Improved DC Driving Capability.
- Available in EXPRESS
 - Standard Temperature Range
 - Extended Temperature Range

The Intel® 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

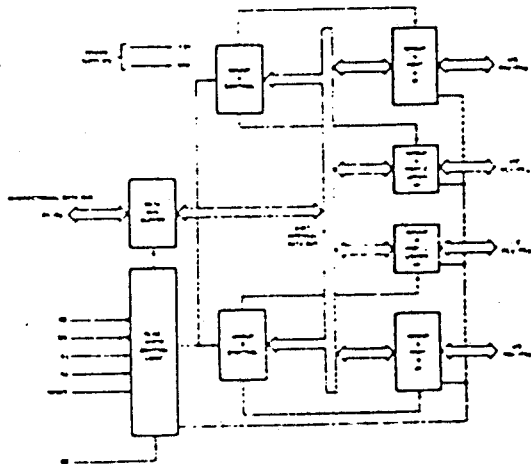


Figure 1. 8255A Block Diagram

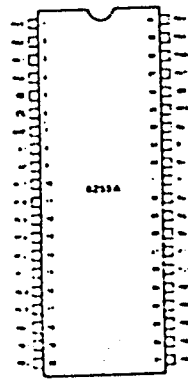


Figure 2. Pin Configuration



8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel[®] microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A 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 8255A and the CPU.

(RD)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

(WR)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A₀ and A₁)

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

8255A BASIC OPERATION

A ₁	A ₀	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
					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 - 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS - 3-STATE

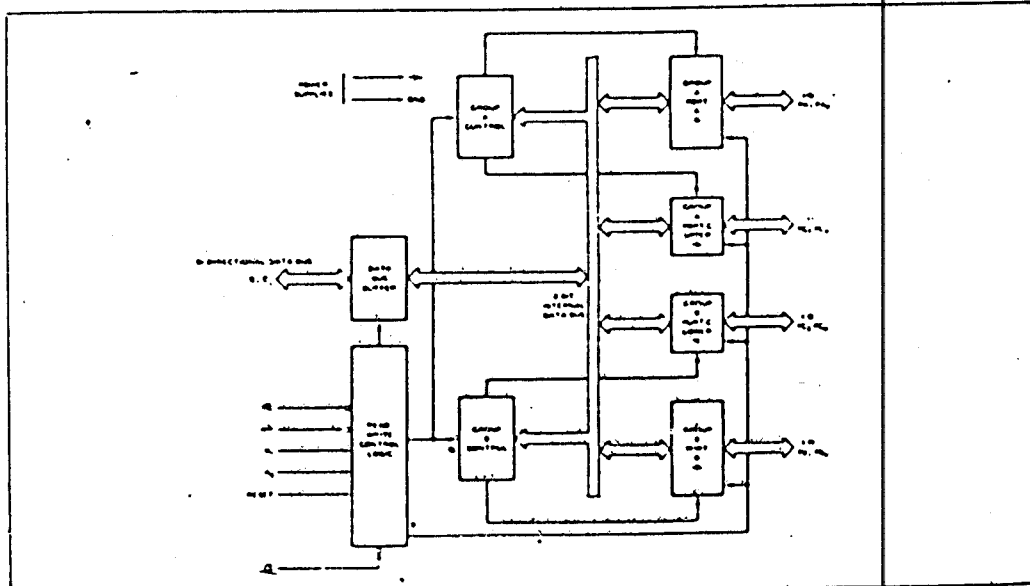


Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions



(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

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 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 55A.

The Control Word Register (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 only be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in 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 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

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

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 outputs and status signal inputs in conjunction with ports A and B.

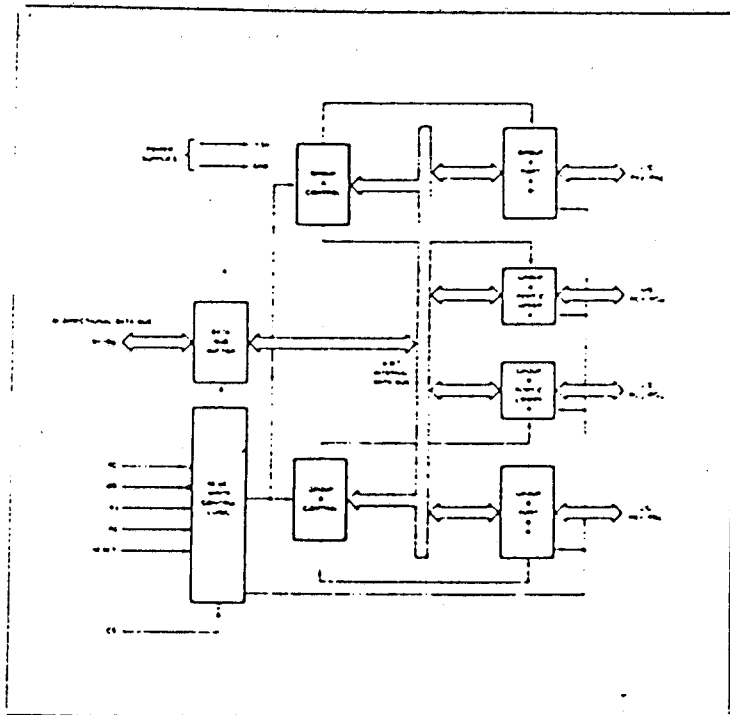
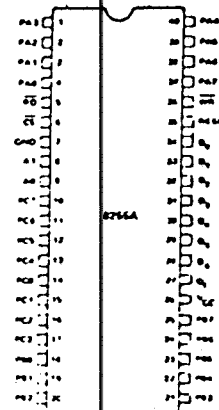


Figure 4. 8255A Block Diagram Showing Group A and Group B Control Functions

PIN CONFIGURATION



PIN NAMES

- 1, 16 DATA BUS (BIDIRECTIONAL)
- RESET RESET INPUT
- CS CHIP SELECT
- RD RD INPUT
- WR WR INPUT
- MS/MT MODE ADDRESS
- PA7/PA6 PORT A (BIT)
- PB7/PB6 PORT B (BIT)
- PC7/PC6 PORT C (BIT)
- VCC +5 VOLTS
- GND 0 VOLTS

8255A 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 (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

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.

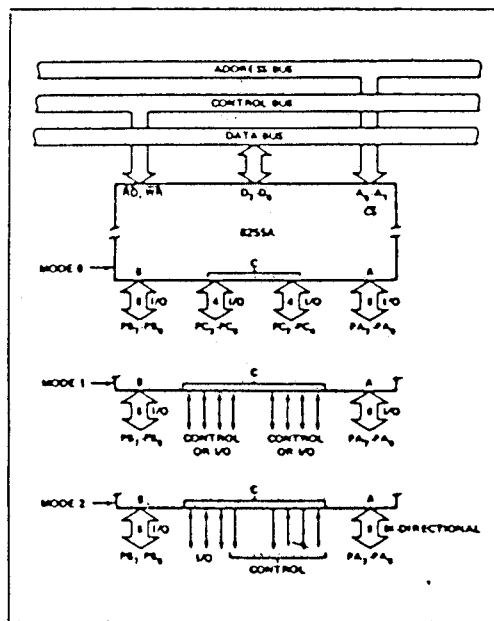


Figure 5. Basic Mode Definitions and Bus Interface

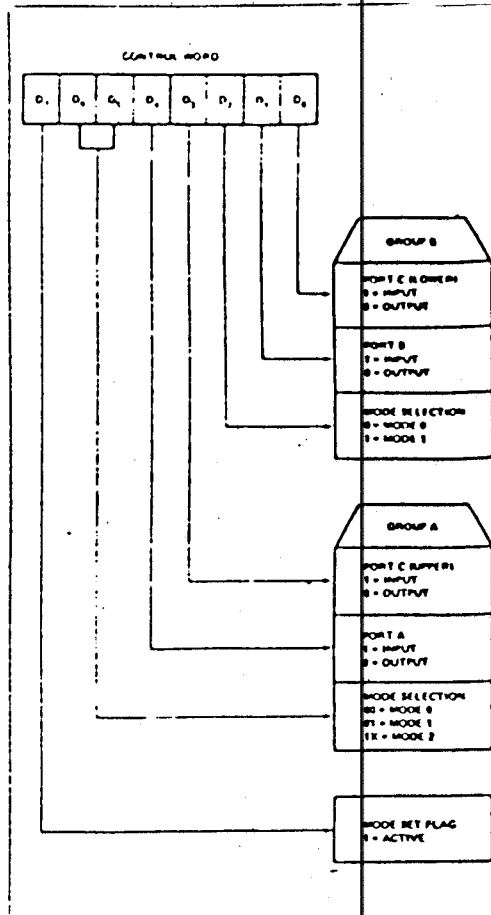


Figure 6. Mode Definition Format

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 8255A 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

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.

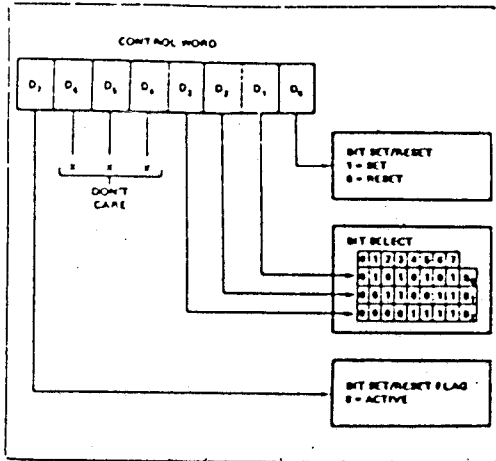


Figure 7. Bit Set/Reset Format

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 data output ports.

Interrupt Control Functions

When the 8255A 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 disallow or allow a specific I/O device to interrupt the CPU 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

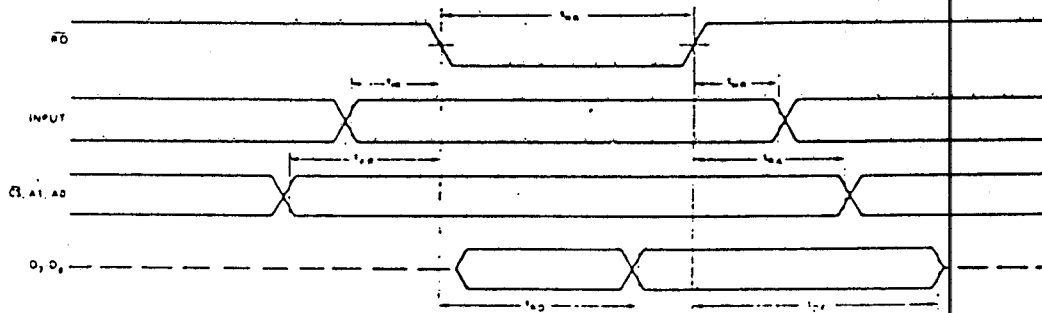
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

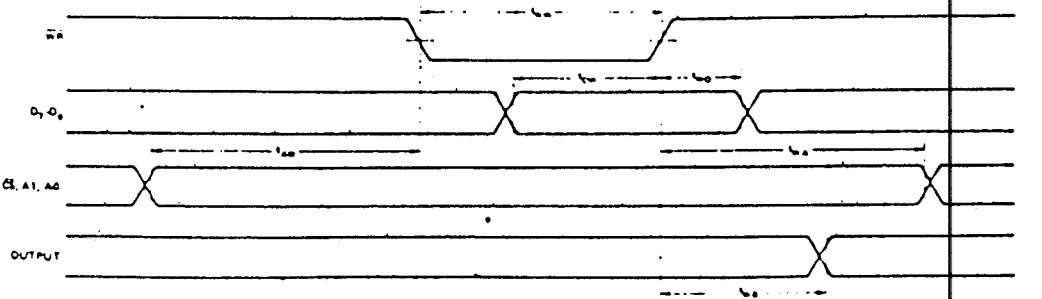
MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.



MODE 0 (Basic Input)



MODE 0 (Basic Output)



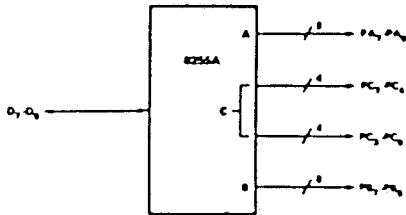
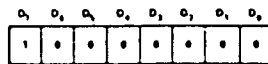
8255A/8255A-5

MODE 0 Port Definition

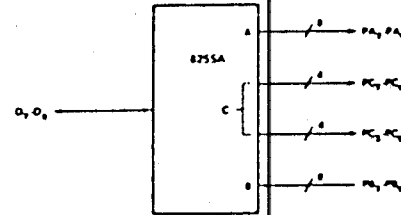
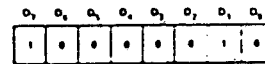
A		B		GROUP A			GROUP B		
D ₄	D ₃	D ₁	D ₀	PORT A	PORT C (UPPER)	n	PORT B	PORT C (LOWER)	
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT	
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT	
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT	
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT	
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT	
0	1	0	1	OUTPUT	INPUT	5	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	

MODE 0 Configurations

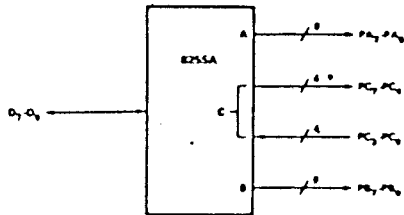
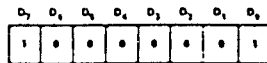
CONTROL WORD #0



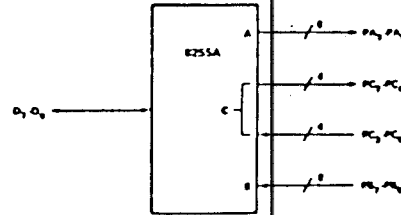
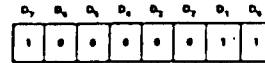
CONTROL WORD #2



CONTROL WORD #1



CONTROL WORD #3



intel

8255A/8255A-5

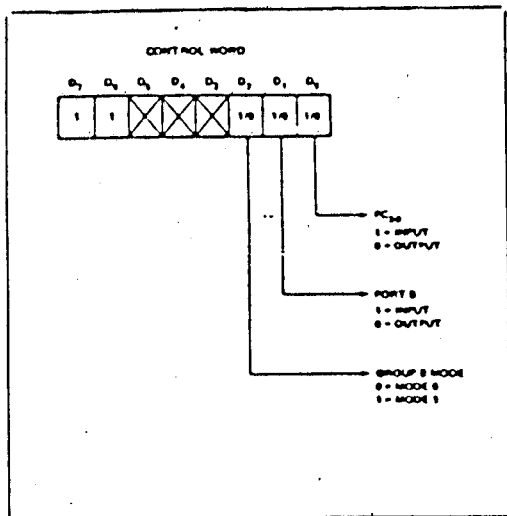


Figure 13. MODE Control Word

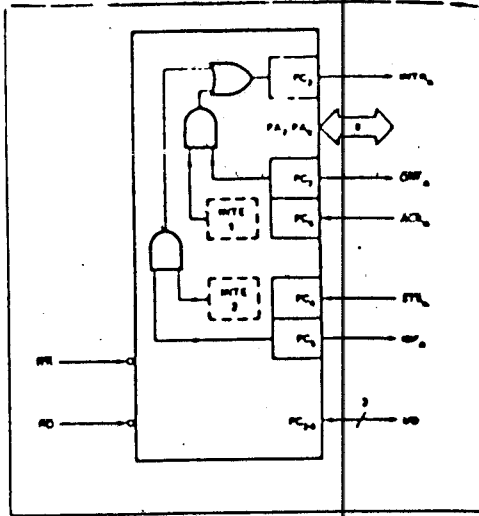


Figure 14. MODE 2

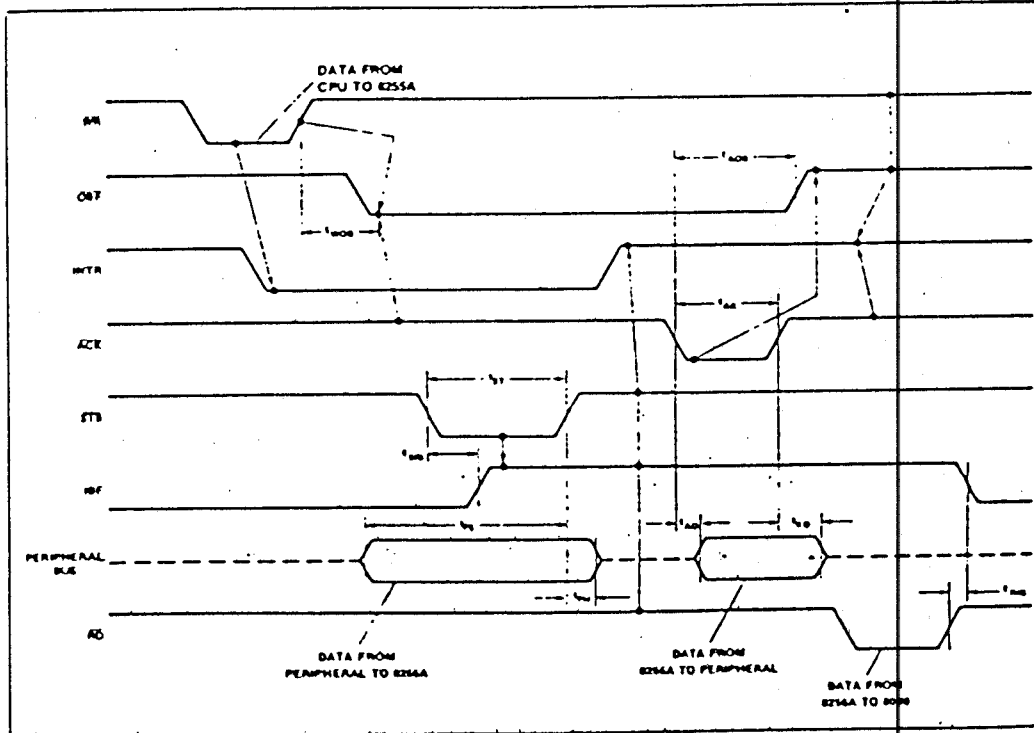


Figure 15. MODE 2 (Bidirectional)

NOTE: Any sequence where \overline{WR} occurs before \overline{ACK} and \overline{STB} occurs before \overline{RD} is permissible.
 $(\overline{INTR} = \overline{IBF} \cdot \overline{MASK} \cdot \overline{STB} \cdot \overline{RD} \cdot \overline{OBF} \cdot \overline{MASK} \cdot \overline{ACK} \cdot \overline{WR})$



MOKHAMMAD SOLEHUDDIN adalah putra ke lima dari tujuh bersaudara, dari putra bapak H. Abdul Qodir dengan ibu Kasyiatun, dilahirkan di Sidoarjo pada tanggal 20 April 1966.

Pendidikan formal yang telah ditempuhnya adalah :

1. SD Negeri Tanjungsari Sepanjang, pada tahun 1972-1978.
2. SMP Negeri Taman Sidoarjo, pada tahun 1978-1982.
3. SMA Negeri Krian Sidoarjo, pada tahun 1982-1985.
4. Perguruan Tinggi ITS Surabaya Jurusan Teknik Elektro pada tahun 1984 - sekarang.

Pada bulan Oktober 1992 mengikuti seminar Tugas Akhir di Jurusan Teknik Elektro bidang studi Teknik Sistem Pengaturan sebagai salah satu syarat untuk mendapat gelar sarjana teknik elektro.