

KONSEP KOMUNIKASI PARALEL

4.1. PENDAHULUAN

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

Port paralel yang baru, distandarisasi dengan IEEE. 1284 yang dikeluarkan pada tahun 1984. Standar ini mendefinisikan 5 macam mode operasi sebagai berikut:

1. Mode Kompatibilitas;
2. Mode Nibel, tidak akan dibahas pada buku ini;
3. Mode Byte, tidak akan dibahas pada buku ini;
4. Mode EPP (*Enhanced Parallel Port*), dibahas sebagian;
5. Mode ECP (*Extended Capabilities Port*), dibahas sebagian.

Tujuan standarisasi ini untuk membantu merancang penggerak (*driver*) dan piranti yang baru yang kompatibel antara satu dengan lainnya serta kompatibel mundur (*backwards*) dengan SPP (*Standard Printer Port*). Mode Kompatibilitas, Nibel dan Byte menggunakan perangkat keras standar yang tersedia pada kartu port paralel asli, sedangkan Mode ECP dan EPP membutuhkan perangkat keras tambahan yang mampu bekerja secara cepat, namun masih kompatibel dengan SPP.

Sebagaimana diketahui, mode kompatibel atau "Mode Centronics", hanya mampu mengirim data searah saja pada kecepatan normal 50 kbyte per detik namun dapat lebih dipercepat hingga 150 kbyte/detik. Untuk dapat menerima data, Anda harus merubahnya menjadi Mode Nibel atau Byte. Mode Nibel mampu memasukkan data nibel (4 bit). Sedangkan Mode Byte menggunakan sifat dwi-arah dari port paralel (hanya Anda dapatkan pada beberapa komputer lama) untuk memasukkan data byte (8 bit).

4.2. PENJELASAN UMUM PORT PARALEL

Port ECP dan EPP menggunakan tambahan perangkat keras untuk menghasilkan dan mengatur *handshaking* (sinyal-sinyal tanda *acknowledge*). Untuk mengeluarkan sebuah byte ke pencetak (atau apa saja) menggunakan Mode Kompatibel, maka langkah-langkah operasinya dalam perangkat lunaknya sebagai berikut:

1. Tuliskan data byte ke Port Data;
2. Periksa apakah pencetak sedang sibuk; Jika pencetaknya sibuk, maka tidak akan menerima data apapun, sehingga data yang dituliskan akan hilang begitu saja;
3. Jadikan sinyal *strobe* (pin 1) menjadi rendah (=0), ini digunakan untuk memberitahukan pencetak bahwa ada data yang siap dikirimkan. pada jalur data (pin 2 s/d 9);
4. Jadikan sinyal *strobe*-nya kembali tinggi (=1) setelah menunggu kira-kira 5 mikrodetik setelah menjadikan *strobe*-nya = 0.

Langkah-langkah tersebut mengakibatkan kecepatan port menjadi menurun. Port ECP dan EPP melakukan hal yang serupa dengan bantuan perangkat keras agar memeriksa apakah pencetaknya sibuk atau tidak dan menghasilkan sinyal *strobe* dan/atau sinyal *handshaking* lain yang sesuai. Hal ini berarti hanya ada satu instruksi saja yang dilaksanakan, sehingga bisa meningkatkan kecepatan. Port-port ini dapat mengeluarkan data dengan kecepatan 1 s/d 2 megabyte/detik. Port ECP juga memiliki kelebihan dalam penggunaan port DMA dan penyangga FIFO, sehingga data bisa digeser-geser tanpa perintah I/O.

Pada tabel 4.1 ditunjukkan daftar pin pada konektor DB-25 dan Centronics (34 pin). Konektor DB-25 adalah yang paling banyak ditemukan pada port paralel komputer, sedangkan konektor Centronics banyak dijumpai pada pencetak. Standar IEEE. 1284 menetapkan 3 macam jenis konektor yang berbeda, yaitu:

1. IEEE 1284 Tipe A adalah konektor DB-25 yang banyak kita jumpai pada komputer-komputer saat ini;

2. 1284 Tipe B adalah konektor Centronics 34 pin yang banyak dijumpai pada pencetak; dan
3. 1284 Tipe C adalah konektor 36 pin yang mirip dengan Centronics, namun lebih kecil. Konektor ini diklaim memiliki pengunci (*latch*) jenis klip (*clip*), sifat elektrik yang lebih baik serta mudah dirakit. Juga mengandung dua pin tambahan yang dapat digunakan untuk mendeteksi apakah piranti yang terpasang memiliki daya atau tidak. Konektor tipe ini disarankan untuk rancangan-rancangan baru. *STOBE*

Tabel 4.1 Daftar pin pada DB_25 dan Centronics (PS = Printer Status, PC = Printer Control)

Pin (DB-25)	Pin (Centronics)	SPP Signal	Arah In/out	Register
1	1	<i>STOBE</i>	Out	<i>PCO</i>
2	2	Data 0	Out	Data
3	3	Data 1	Out	Data
4	4	Data 2	Out	Data
5	5	Data 3	Out	Data
6	6	Data 4	Out	Data
7	7	Data 5	Out	Data
8	8	Data 6	Out	Data
9	9	Data 7	Out	Data
10	10	ACK	In	PS6
11	11	<i>BUSY</i>	In	<i>PC7</i>
12	12	Paper-Out/ Paper-End	In	PS5
13	13	Select	In	PS4
14	14	<i>Autofeed</i>	In/Out	<i>PC</i>
15	32	Error	In	PC3
16	31	Initialize	In/Out	PC2
17	36	<i>Select-in</i>	In/Out	<i>PC3</i>
18-25	19-30	Ground	Gnd	

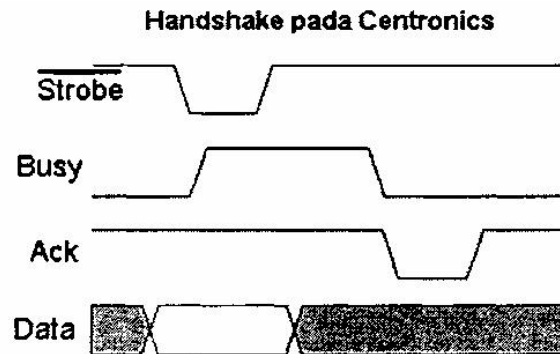
Terjadinya transisi dari logika 1 ke 0 pada ACK akan membangkitkan interupsi perangkat keras port paralel IRQ7. Pada jalur Busy (dan yang sejenis, dengan tanda negasi), jika sinyal logika 1 diterapkan pada pin ini dan kemudian

register status dibaca, maka akan terbaca (pada bit-7) sebagai 0 (nol) bukan 1 (satu).

Keluaran dari port paralel normalnya dalam tingkat logika TTL. Arus yang dapat disedot (*sink*) atau diberikan (*source*) ber-variabel dari port satu ke port yang lain. Hampir semua port paralel diimplementasikan dalam ASIC, yang mampu menyedot dan memberikan arus sekitar 12 mA. Berdasarkan *data sheet*, kemampuannya beragam, Sink/Source 6 mA, Source 12 mA / Sink 20 mA, Sink 16 mA / Source 4 mA, Sink/Source 12 mA dan lain sebagainya (hanya terdapat perbedaan-perbedaan kecil). Hal yang terbaik yang dapat Anda lakukan adalah dengan menggunakan penyangga (*buffer*), sehingga cukup hanya arus kecil saja yang dapat digunakan melalui port paralel.

4.3. SPESIFIKASI PORT PARALEL CENTRONICS

Centronics merupakan standar pengiriman data komputer ke pencetak generasi awal. Hampir semua pencetak menggunakan teknik *handshake* ini dan biasanya diimplementasikan menggunakan Port Paralel Standar (KPS atau *Standar Parallel Port - SPP*) melalui kontrol perangkat lunak.



Gambar 4.1 Diagram pewaktuan handshake pada Centronics

Perhatikan gambar 4.1, data pertama kali dikirim pada jalur data (pin 2 - 7 Port Paralel), kemudian komputer akan memeriksa apakah pencetak dalam kondisi sibuk (*busy*) atau tidak, dalam hal ini sedang dalam kondisi rendah (logika 0). Program kemudian meng-aktifkan *strobe*, tunggu selama minimum $1 \mu d$ kemudian dimatikan kembali. Data kemudian dibaca oleh pencetak (atau alat lain) saat

transisi naik dari sinyal *strobe*. Pencetak akan memberikan indikasi sibuk karena sedang memproses data melalui jalur bus data. Sekali pencetak menerima data, maka dia akan mengirimkan sinyal **Ack** (*acknowledge*) sebagai pulsa rendah selama 5 μ d.

Seringkali komputer akan mengabaikan sinyal **Ack** untuk menghemat waktu. Pada Port ECP, Anda akan lihat mode Centronics Cepat, yang membolehkan perangkat keras mengerjakan *handshaking* untuk Anda. Seorang pemrogram hanya cukup menuliskan data pada jalur I/O. Perangkat keras akan memeriksa apakah pencetak dalam kondisi sibuk atau tidak, kemudian membangkitkan sinyal *strobe*. Catatan: dalam mode ini juga tidak akan diperiksa sinyal Ack!

4.4. ALAMAT-ALAMAT PORT PARALEL

Port paralel umumnya memiliki tiga alamat dasar yang bisa digunakan, sebagaimana ditunjukkan pada tabel 4.2. Alamat dasar 3BCh pertama kali diperkenalkan sebagai alamat port paralel pada kartu-kartu video lama. Alamat ini kemudian sempat menghilang, saat port paralel dicabut dari kartu-kartu video. Sekarang muncul kembali sebagai pilihan untuk port paralel yang terpadu dengan *motherboard*, yang konfigurasinya dapat diubah melalui BIOS.

LPT1 biasanya memiliki alamat dasar \$378, sedangkan LPT2 adalah 278h. Ini adalah alamat umum yang bisa dijumpai, namun alamat-alamat dasar ini bisa berlainan antara satu komputer dengan komputer lainnya.

Tabel 4.2. Alamat-alamat dasar port paralel

Alamat (Heks)	Keterangan
3BC-3BF	Digunakan untuk Port Paralel Yang terpadu dengan kartu-kartu video, tidak mendukung alamat-alamat ECP
378-37F	Bisa digunakan untuk LPT1
278-27F	Bisa digunakan untuk LPT2

Saat pertama kali komputer dihidupkan, BIOS (*Basic Input/ Output System*) akan menentukan jumlah port yang dimiliki kemudian diberi label LPT1, LPT2 dan LPT3. Pertama kali BIOS akan memeriksa alamat \$3BC, jika ditemukan port paralel pada alamat tersebut, maka akan diberi label LPT1, kemudian dicari pada lokasi berikutnya

\$378, jika ditemukan akan diberi label selanjutnya yang sesuai. Bisa jadi LPT1 jika tidak ditemukan port paralel di \$3BC atau mungkin LPT2, jika ditemukan port paralel pada alamat tersebut. Alamat port terakhir yang diperiksa adalah \$278 dan mengikuti langkah-langkah yang telah dijelaskan tadi. Sehingga dimungkinkan kita memiliki LPT2 dengan alamat \$378 bukan \$278 sebagaimana yang diharapkan.

Apa yang membuat hal seperti ini menjadi membingungkan adalah, karena beberapa perusahaan memasang *jumper* yang membolehkan Anda untuk mengatur port Anda ke LPT1, LPT2 dan LPT3. Nah sekarang berapa alamat dari LPT1? Hampir semua kartu, untuk LPT1 dialamatkan pada \$378 dan LPT2 pada \$278 namun beberapa menggunakan \$3BC sebagai LPT1, \$378 sebagai LPT2 dan \$278 sebagai LPT3.

Label-label LPT1, LPT2 dan LPT3 seharusnya tidak perlu di-kawatirkan bagi mereka yang hanya menginginkan pengantar-mukaan piranti dengan komputer. Biasanya alamat dasar-lah yang digunakan dalam program antarmuka dari label LPT1 dan seterusnya. Namun jika Anda tetap ingin tahu alamat LPT1 atau LPT yang lain, Anda dapat menggunakan tabel tengok (*lookup table*) yang disediakan BIOS. Saat BIOS mengarahkan alamat-alamat pada piranti pencetak Anda, maka dia akan menyimpan alamat pada lokasi khusus dalam memori, sehingga Anda bisa menemukannya, hal ini sebagaimana ditunjukkan pada tabel 4.3.

Pada tabel 4.3 tersebut, ditunjukkan alamat yang dapat kita gunakan untuk menemukan alamat-alamat port pencetak (paralel!) dalam Area Data BIOS. Masing-masing alamat membutuhkan ruang 2 byte. Program berikut (dalam bahasa C) digunakan untuk membaca lokasi alamat-alamat tersebut untuk menemukan alamat-alamat port paralel (pencetak):

Tabel 4.3 Daftar Alamat port paralel

Alamat Awal	Fungsi
0000 :0408	Alamat Dasar LPT1
0000:040A	Alamat Dasar LPT2
0000 :040C	Alamat Dasar LPT3
0000 :040E	Alamat Dasar LPT4

4.5. REGISTER-REGISTER PERANGKAT LUNAK-SPP (STANDARD PARALLEL PORT)

Tabel 4.4. Register Data port paralel

Offset	Nama	Baca/Tulis	Bit 7	Keterangan
Base + 0	Data Port	Baca	Bit 7	Data 7
		Dan/atau	Bit 6	Data 6
		Tulis	Bit 5	Data 5
			Bit 4	Data 4
			Bit 3	Data 3
			Bit 2	Data 2
			Bit 1	Data 1
			Bit 0	Data 0

Alamat dasar, biasa dinamakan Port Data atau Register Data digunakan untuk mengeluarkan data pada jalur data Port Paralel (pin 2 s/d 9). Register ini normalnya sebagai port hanya-tulis. Jika Anda mencoba membaca dari port ini, Anda akan mendapatkan byte terakhir yang terkirim. Bagaimanapun juga jika port Anda dwi-arah, Anda juga bisa menerima data melalui alamat ini.

Port status (alamat dasar+1), perhatikan tabel 4.5, sebagai port hanya-baca saja. Data apa saja yang dituliskan ke port ini akan diabaikan. Port status berasal dari lima masukan port paralel (pin 10, 11, 12, 13 dan 15), sebuah register status IRQ dan dua bit tercadang. Harap dicatat bahwa bit-7 (Busy) sebagai masukan aktif rendah. Jika bit-7 terbaca sebagai logika 0 artinya pada pin tersebut terpasang tegangan +5V. Juga pada bit-2 (IRQ), jika bit ini terbaca '1', artinya interupsi (sela) tidak muncul.

Register kontrol (alamat dasar+2), perhatikan tabel 4.6, sebagai register tulis saja. Saat sebuah pencetak disambungkan pada port paralel, maka ia membutuhkan 4 kontrol, yaitu *Strobe*, *Auto Linefeed*, *Initialize* dan *Select Printer*, yang semua sifatnya sungsang (*inverted*) kecuali *Initiliaze*.

Tabel 4.5. Register Status port paralel

Offset	Nama	Baca/Tulis	Bit	Keterangan
Base+1	Port Status	Hanya Baca	Bit 7	Busy
			Bit 6	Ack
			Bit 5	Paper Out/End
			Bit 4	Select
			Bit 3	Error
			Bit 2	IRQ
			Bit 1	Tercadang
			Bit 0	Tercadang

Tabel 4.6. Register Kontrol port paralel

Offset	Nama	Baca/Tulis	Bit	Keterangan
Base+2	Port Kontrol	Tulis	Bit 7	Tidak Digunakan
			Bit 6	Tidak Digunakan
			Bit 5	Aktifasi Port Dwi-arah
			Bit 4	Aktivasi IRQ Melalui jalur Ack
			Bit 3	Select-In
			Bit 2	Inisialisasi (Reset)
			Bit 1	Autofeet
			Bit 0	STORE

Dalam hal ini pencetak tidak akan mengirimkan sinyal untuk menginisialisasi komputer dan juga tidak akan memberitahukan komputer untuk menggunakan *auto linefeed*. Bagaimanapun juga, keempat keluaran ini dapat dimanfaatkan sebagai masukan. Namun perlu diketahui bahwa sifatnya adalah "kolektor terbuka" (*open collector* atau *open drain* pada piranti CMOS). Ini artinya memiliki dua kondisi, kondisi rendah (0V) dan kondisi impedansi tinggi (rangkaian terbuka). Sehingga jika komputer mengkondisikan pin-pin sebagai tinggi (misalnya +5V) dan piranti Anda yang terhubung membutuhkan kondisi rendah (0V), maka akan terjadi hubung singkat pada port yang menyebabkan terjadinya konflik pada pin tersebut

Normalnya port paralel memiliki resistor internal *pull-up* (kira-kira sekitar 4k7 ohm), tetapi seperti yang Anda harapkan, tidak semuanya seperti itu. Beberapa

memiliki keluaran kolektor terbuka, sedangkan lainnya hanya memiliki keluaran *totem pole* normal. Agar piranti Anda bisa bekerja dengan benar pada berbagai macam port paralel, maka Anda dapat menggunakan resistor eksternal. Jika digunakan pada port paralel dengan resistor *pull-up* internal, maka seakan-akan resistor ini akan paralel dengan resistor internal tersebut, sedangkan jika digunakan pada port paralel dengan keluaran *totem pole*, resistor akan bertindak seolah-olah sebagai beban.

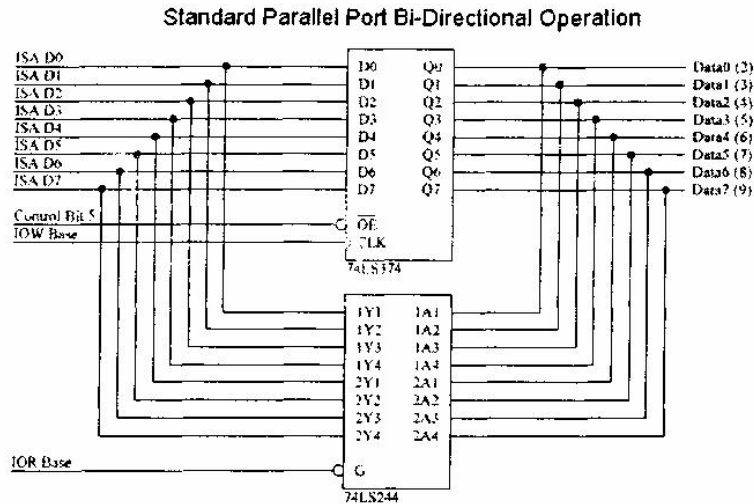
Resistor 4K7 ohm (usahakan jangan yang lebih rendah) dapat digunakan untuk mengkondisikan pin menjadi HIGH (*pull high*). Ini sebagai antisipasi, jika yang digunakan adalah port paralel dengan resistor internal, sehingga penggunaan resistor eksternal yang kemudian diparalelkan dengan resistor internal, akan menghasilkan resistor *pull-up* yang rendah nilainya. Dalam kondisi impedansi tinggi, pin pada port paralel dalam tegangan HIGH (+5V), pada kondisi inilah piranti eksternal Anda dapat menyebabkan pin menjadi LOW (*pull low*) dan mengakibatkan hasil pembacaan port kontrol menjadi berbeda. Dengan dasar ini, maka port kontrol yang 4 bit dapat digunakan sebagai jalur data dua arah, hanya saja port kontrol harus diset xxxx 0100 terlebih dahulu agar dapat digunakan untuk membaca data (yang bernilai LOW tentunya), karena semua pin kontrol dalam kondisi +5V sehingga Anda bisa mengubahnya menjadi LOW (*GND*) atau *pull-low*.

Bit 4 dan 5 merupakan kontrol internal. Bit-4 untuk mengaktifkan IRQ (lihat daftar IRQ pada pembahasan port serial) dan bit-5 digunakan untuk mengaktifkan port data dwi-arah, artinya port data dapat digunakan sebagai masukan 8-bit (DATA0 s/d DATA7). Hal ini bisa dilakukan jika port paralelnya mendukung. Bit-6 dan 7 sebagai cadangan, jika dilakukan penulisan data pada bit-6 dan 7 ini, maka akan diabaikan.

4.6. PORT DWI-ARAH (*BI-DIRECTIONAL PORT*)

Pada gambar 4.2 ditunjukkan Register data Port Paralel yang disederhanakan. Kartu port Paralel yang asli biasanya mengimplementasikan IC-IC seri 74LS. Saat ini semuanya sudah dikemas dalam satu IC ASIC

(Application Specific Integrated Circuit), namun dasar kerjanya tetap sama.



Gambar 4.2. Operasi Port Paralel dwi-arah Standar

Port non dwi-arah biasanya menggunakan IC 74LS374 dengan keluaran yang dibuat permanen rendah (LOW), dengan demikian port data selalu bertindak sebagai keluaran saja. Saat Anda membaca register data Port Paralel, datanya berasal dari 74LS374 yang juga terhubung dengan pin-pin data. Sekarang jika Anda mampu *meng-overdrive* 374, akan Anda dapatkan Port Dwi-arah (atau sebagai port masukan saja, jika Anda merusakkan keluaran-keluaran terkunci atau *output latches*!)

Banyak yang melakukan hal tersebut. Pernah ada suatu rangkaian (di Internet) yang menghubungkan antara sebuah osiloskop dengan port paralel. Pengarangnya menggunakan suatu ADC, namun ternyata ADC - nya menggunakan transistor pada masing-masing jalurnya, agar dapat bekerja! Tidak tahu mengapa? Ada juga yang menggunakan 68HC11 yang tidak cukup mampu menyedot (*sink*) arus (30 hingga 40 mA)!

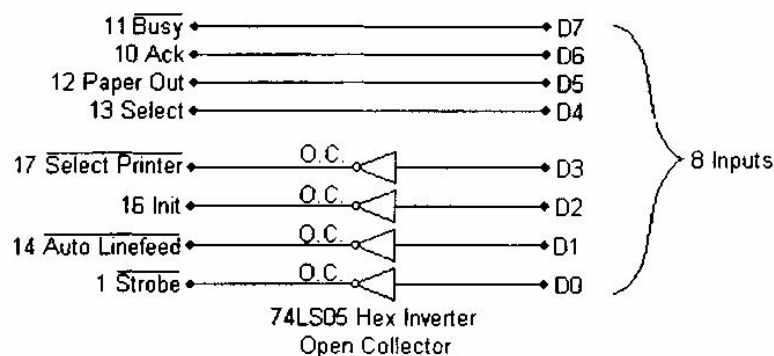
Port paralel dwi-arah menggunakan bit-5 dari register kontrol yang dihubungkan ke OE (*Output Enable*)-nya. 374 sehingga peng-gerak keluaran dapat dimatikan. Dengan cara ini, Anda dapat mem-baca data yang tersedia pada pin-pin data Port Paralel, tanpa meng-alami konflik bus dan kebocoran arus yang berlebihan.

Bit-5 Register Kontrol dapat digunakan untuk mengaktifkan dan mematikan fungsi dwi-arah pada Port Paralel. Hal ini hanya di-benarkan pada port dwi-arah yang sesungguhnya. Jika bit ini sama dengan 1, pin 2 hingga 9 dalam kondisi impedansi tinggi. Dalam kon-disi ini, Anda boleh memasukkan data pada jalur-jalur tersebut dan membacanya dari Port Data. Sembarang data yang dituliskan ke port data akan disimpan tetapi tidak akan tersedia pada pin-pin data. Untuk mematikan fungsi dwi-arah ini, set bit-5=0.

Bagaimanapun juga tidak semua port sifatnya sama. Port lain boleh jadi membutuhkan bit- 6 pada Register Kontrol untuk meng aktifkan mode dwi- arah dan menset bit-5-nya untuk mematikan mode tersebut. Perusahaan lain membuat port dwi-arah mereka dengan cara yang berbeda. Jika Anda ingin menggunakan port dwi-arah untuk pemasukan data, maka ujilah terlebih dahulu dengan *logic probe* atau multimeter untuk memastikan port paralel dalam mode dwi-arah.

4.7. MENGGUNAKAN PORT PARALEL UNTUKMASUKAN 8 BIT

Jika port paralel Anda tidak mendukung mode dwi-arah, janganlah kecewa! Anda dapat memasukkan maksimum 9-bit kapan saja. Untuk melakukan hal ini, Anda dapat menggunakan 5 jalur masukan pada Port Status dan 4 jalur masukan (kolektor terbuka) pada Port Kontrol.



Gambar 4.3. Inverter Heks 74LS05 Kolektor terbuka

Masukan ke Port Paralel telah dipilih sedemikian rupa sehingga akan memudahkan kita. Sinyal Busy dijadikan sebagai MSB (Bit - 7) dari Port Status, kemudian diikuti, sesuai urutan gambar 4.3, sinyal-sinyal Ack, Paper Out dan Select melalui nibel (4-bit) Tinggi atau MS (*Most Significant*) dari Port Kontrol. Tanda garis di atas sinyal digunakan sebagai tanda inversi perangkat keras, yaitu +5V akan terbaca 0 dari register, sedangkan GND akan terbaca 1. Sedangkan Port Status hanya memiliki sebuah masukan terinversi.

Port Kontrol digunakan untuk membaca nibel Rendah atau LS (*Least Significant*). Sebagaimana telah dijelaskan sebelumnya, Port Kontrol merupakan keluaran kolektor terbuka, yaitu dua kemungkinan kondisi, impedansi tinggi dan GND. Jika kita menghubungkan masukan-masukan kita langsung ke port (misalnya dengan sebuah ADC0804 yang keluarannya totem pole), maka akan terjadi konflik jika masukannya tinggi dan port mencoba untuk membuat rendah (*pull it down*), dengan demikian kita gunakan inverter kolektor terbuka.

Bagaimanapun juga hal ini tidak perlu harus selalu dilakukan. Jika kita menghubungkan saklar-saklar kutub tunggal (*single pole switches*) dengan port menggunakan resistor *pull up*, maka tidak perlu dkuatirkan masalah proteksi di atas. Demikian juga jika perangkat lunak Anda menginisialisasi Port Kontrol dengan xxxxOIOO sehingga semua pin pada port kontrol dalam kondisi tinggi, maka mungkin tidak perlu juga. Jika Anda tidak mempedulikan dan alat terlanjur terpasang pada Port Paralel dan program telah berjalan, maka Anda bisa mendapatkan masalah.

Permasalahan lain yang perlu diperhatikan adalah resistor *pull-up* yang terpasang pada Port Kontrol dan rata-rata resistor tersebut bernilai 4K7 ohm. Agar jalur tersebut dapat menyedot arus ke kondisi rendah (LOW), maka alat Anda harus mampu menyedot arus sekitar 1 mA, yang mana beberapa piranti-piranti daya rendah kesulitan melakukan hal tersebut. Sekarang apa yang akan terjadi jika dianggap beberapa port memiliki resistor *pull-up* sebesar 1K ohm? Maka alat Anda harus mampu menyedot arus (*sink*) sebesar 5 mA. Hal ini semakin menguatkan alasan penggunaan inverter kolektor terbuka.

Walaupun penyangga kolektor terbuka bisa juga digunakan, namun inverter kolektor terbuka yang dipiih, karena inverter tersebut banyak dijumpai di pasaran. Kemungkinan lain adalah penggunaan transistor.

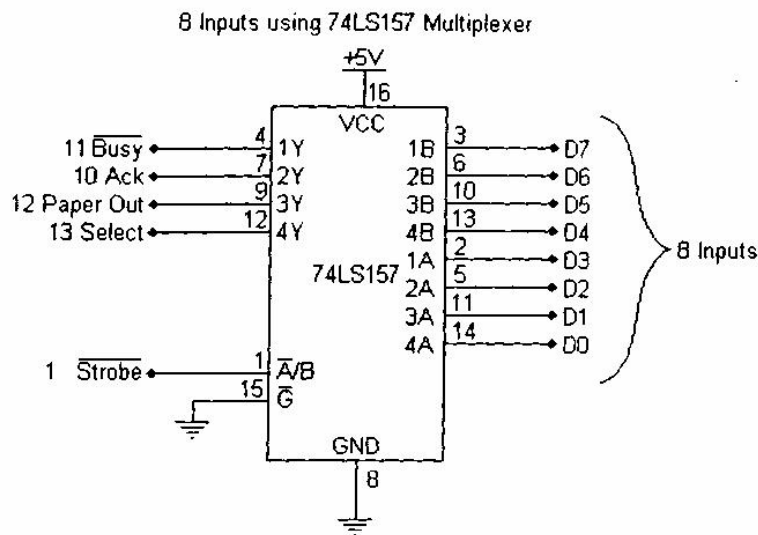
Masukan D3 dihubungkan melalui inverter ke Select **Printer**. Sinyal ini merupakan bit-3 dari Port, Kontrol, D2, D1 dan DO di-hubungkan masing-masing ke **Init**, Auto linefeed dan Strobe. Jika hal ini suciuh selesai kita rangkai, sekarang ditulis programnya. Pertama kali, kita harus menuliskan xxxxOIOO ke Port Kontrol agar semua jalur kontrol dalam kondisi tinggi (HIGH), sehingga dapat digunakan untuk menarik rendah (*pull down*) masukan data. Kemudian kita dapat membaca nibel MS dari port Status. Dengan demikian akan kita AND-kan hasilnya dengan FOh (agar nibel LS terhapus = 0). Sinyal **Busy** di-inversikan secara perangkat keras, namun sementara ini belum kita khawatirkan. Sekali dua byte direkonstruksikan, kita dapat membunuh dua burung dengan hanya sebuah batu, dengan cara mentogel **Busy dan Init** pada saat yang bersamaan.

Sekarang kita bisa membaca nibel LS pada Port Kontrol begitu mudahnya! Sekarang kita tidak tertarik pada nibel MS pada port, sehingga kita AND-kan hasilnya dengan OFh untuk menghapus nibel MS tersebut. Setelah itu, kita kombinasikan keduanya, dengan cara OR-kan kedua byte tersebut! Sekarang diperoleh sebuah byte, namun perlu diketahui bahwa bit-2 dan 7 diinversi, sehingga perlu di-XOR-kan dengan 84h atau 1000 OIOOb untuk mentogel kedua bit tersebut.

Catatan: Beberapa port kontrol bukan merupakan kolektor terbuka, namun berupa *totem pole*. Hal ini berlaku juga untuk port EPP dan ECP. Normalnya jika anda menggunakan mode EPP atau ECP pada Port Paralel, port kontrol menjadi keluaran *totem pole* saja. Sekarang apa yang terjadi jika Anda menghubungkan peralatan ke Port Paralel dalam mode ini? Untuk portabilitas, disarankan untuk menggunakan rangkaian yang akan dibahas berikut, membaca sebuah nibel.

4.8. MODE NIBEL

Mode nibel merupakan cara untuk membaca 8bit data tanpa menempatkan port pada mode terbalik (*reverse mode*) dan tanpa menggunakan jalur data. Mode nibel menggunakan sebuah IC *Quad 2 to 1 multiplexer* (74LS157) untuk membaca sebuah nibel data, ke- mudian pindah ke nibel satunya dan membacanya. Melalui perangkat lunak kedua nibel tersebut disatukan menjadi sebuah byte. Kelemahan mode ini hanya masalah lambatnya proses. Dan dibutuhkan beberapa perintah I/O untuk membaca sebuah byte dan membutuhkan IC eksternal



Gambar 4.5. Masukan 8-bit menggunakan 74LS157

Cara kerja rangkaian pada gambar 4.5 cukup mudah. 74LS157 Bertindak seakan-akan sebagai empat saklar. Jika masukan A/B dalam kondisi LOW, maka masukan A dipilih, misalnya 1A ke 1Y, 2A ke 2Y dan seterusnya. Jika sekarang masukan A/B = 1, maka masukan B dipilih. Keluaran Y dihubungkan ke port status dari Port Paralel sedemikian hingga menyatakan nibel MS dari register status, hal ini untuk memudahkan dalam penulisan program.

Untuk menggunakan rangkaian ini, pertama kali harus dilakukan inisialisasi multi-plekser untuk memilih masukan A atau B. Kita akan membaca nibel LS terlebih dahulu, sehingga masukan A/B dibuat = 0 (dari B -> Y), karena sinyal **strobe** terinversi secara perangkat keras, maka bit-0 pada port kontrol agar dalam

kondisi LOW (=0) diberi nilai 1. Setelah nibel rendah terpilih, kemudian dilakukan pembacaan dari port status. Perhatikan, jalur **busy** diinversikan, namun belum akan kita tangani saat ini. Kita hanya tertarik pada nibel LS (rendah) saja, sehingga hasilnya di-AND-kan dengan FOH untuk menghapus nibel MS (tinggi).

4.9. PENGGUNAAN INTERUPSI PADA PORT PARALEL

IRQ pada Port Paralel tidak digunakan untuk pencetakan dalam lingkungan DOS atau Windows. Versi awal dari O/S 2 yang menggunakannya, namun tidak dipakai pada versi-versi berikutnya. Interupsi sangat baik digunakan saat melakukan pengantarmukaan alat pengawasan seperti alarm suhu tinggi dan lain - lain, yang mana Anda tidak tahu kapan alarm tersebut aktif. Akan lebih enak menggunakan interupsi dibandingkan menggunakan teknik *polling* dan hal ini akan terasa manfaatnya jika komputer juga harus melakukan proses-proses yang lain.

Interupsi yang digunakan pada Port Paralel adalah IRQ5 atau IRQ7, atau yang lainnya jika kedua interupsi ini telah dipakai. Serta dimungkinkan juga interupsi tersebut dimatikan pada kartunya, jika kartu tersebut hanya digunakan untuk tujuan pencetakan. Namun dengan menggunakan bit -4 pada register kontrol, Anda dapat mengaktifkan dan mematikan interupsi, mengaktifkan IRQ - nya, melalui jalur Ack. Interupsi umumnya akan muncul pada saat terjadi transisi dari rendah ke tinggi (*rising edge*) pada jalur Ack. Dan ada pula yang terjadi pada saat transisi dari tinggi ke rendah.

Kode-kode program berikut digunakan untuk menguji polaritas suatu interupsi sekaligus contoh program yang menggunakan interupsi. Program ini akan memeriksa apakah interupsi terjadi pada saat naik atau jatuhnya sisi pulsa pada jalur Ack. Untuk menggunakan program, cukup Anda sambung salah satu dari kaki 2 hingga 9 (pin-pin data) ke pin Ack (atau kaki 10), misalnya kaki 9 dan 10 (agar tidak memerlukan kabel penghubung atau langsung disolder).

Pada saat kompilasi mungkin akan dihasilkan beberapa peringatan atau warning seperti "condition always true", "condition always false", "unreachable codew" dan lain-lain. Hal ini tidak perlu dicemaskan, karena peringatan-peringatan tersebut berasal dari beberapa

pengujian struktur kondisi dimana IRQ -nya diuji dan karena IRQ didefinisikan sebagai konstanta beberapa hasil tidak berubah. Sebenarnya hal-hal demikian bisa juga diimplementasikan sebagai pengarah praprosesor (preprocessor directive).

Bagian awal rutin program menghitung Vektor Interupsi, alamat PIC dan Mask dalam rangka penggunaan fasilitas interupsi Port Paralel. Setelah Rutin Layanan Interupsi (RLI) disiapkan demikian juga dengan PIC-nya, maka kita aktifkan interupsi pada Port Paralel, yang bisa dilakukan dengan membuat bit-4 pada register kontrol menjadi 1:

Sebelum mengaktifkan interupsi, kita tuliskan \$FF pada port paralel terlebih dahulu agar kondisi 8 jaiur data menjadi jelas (=1). Karena semua jaiur data sudah sama dengan 1, maka kita sekarang jalam posisi siap untuk menuliskan OOh pada port data, yang akan mengakibatkan transisi tinggi ke rendah pada jaiur Ack di port paralel (karena terhubung pada salah satu dari 8 jaiur data).

Jika interupsi muncul pada transisi tinggi ke rendah tersebut maka IF (Interrupt Flag) akan diset dan otomatis jenis polaritas interupsinya adalah transisi tinggi ke rendah. Jika dengan transisi tersebut interupsi belum muncul maka kita tuliskan port data (paralel,) agar terjadi transit rendah ke tinggi. Jika tetap tidak muncul interupsi, kemungkinan tidak bekerja, pastikan IRQ dan alamatdasarnya benar dan koneksi (jalur data dan Ack) pada konektor DB25 male.

4.10. MODE-MODE PORT PAEALEL DALAM BIOS

Saat ini. kebanyakan Part Paralel msrupakan port-port beragam mode dan normalnya dapat dikonfigurasi mnlalui perangkat lunak. Mode-modenyaa meliputi:

- Mode Printer (kadang kala disebut Default atau Normal Mode)
- Mode Standard and Bi-directional (SPP)
- Mode EPP1.7 and SPP
- Mode EPP1.9 and SPP
- Mode ECP
- Mode ECP and EPP1.7
- Mode ECP and EPP1.9

Mode Printer merupakan mode yang paling dasar dan merupakan Port Paralel standar satu arah, tidak ada fitur dwi-arah, sehingga bit-5 pada port kontrol tidak digunakan. Mode Standard and Bi-directional merupakan mode dwi-arah. Menggunakan mode ini, bit-5 pada port kontrol akan membalikkan arah port sehingga Anda dapat membaca kembali nilai yang diberikan pada jalur data.

Mode **EPP1.7 and SPP** merupakan suatu kombinasi dari Mode EPP 1.7 (*Enhanced Parallel Port*) dan SPP. Pada mode ini Anda diperbolehkan mengakses register-register SPP (data, status dan kontrol) serta register-register EPP. Pada mode ini juga, Anda bisa membalikkan arah port menggunakan bit-5 dari register kontrol. EPP 1.7 merupakan versi awal dari EPP dimana tidak terdapat bit untuk *time-out*.

Mode **EPP1.9 and SPP** seperti mode sebelumnya, hanya saja menggunakan versi 1.9 dari EPP dan membolehkan pengaksesan bit *time-out* pada register EPP.

Mode *ECP* akan memberikan Port dengan kemampuan tambahan (*Extended Capabilities Port*). Mode ini dapat diset melalui register kontrol tambahan (*extended control register*) dari ECP. Hanya saja mode ini EPP tidak tersedia. Kita akan bahas register kontrol tambahan ECP lebih lanjut pada pasal-pasal berikutnya.

Mode **ECP and EPP 1.7 dan ECP and EPP1.9** akan memberikan *ECP*, seperti mode sebelumnya. Namun mode EPP dalam ECR-nya mode EPP disediakan dan versi mode EPP-nya mengikmi apakah versi 1.7 atau 1.9.

Mode-mode diatas dapat dikonfigurasi melalui BIOS. Anda bisa mengkonfigurasi-ulang menggunakan perangkat lunak yang Anda tulis, tetapi ini **tidak disarankan**. Register-register perangkat lunak ini, umumnya ditemukan pada alamat-alamat 2FAh, 3F0h, 3F1h dan lain-lain yang diperuntukkan diakses oleh BIOS. Tidak ada standarisasi pada register-register konfigurasi ini, sehingga jika Anda ingin menggunakan register-register tersebut, maka perangkat lunak atau program Anda menjadi tidak portabel.

Pilihan yang baik adalah Mode **ECP and EPP 1.7** atau **ECP and EPP1.9** melalui BIOS kemudian menggunakan register control tambahan ECP untuk memilih mode Port Paralel Anda. Mode EPP1.7 memiliki beberapa masalah

berkaitan dengan Strobo untuk Data dan Alamat yang harus diberikan untuk mengawali sebuah siklus awal (*start cycle*) tidak peduli dengan adanya kondisi tunggu (*wait state*), sehingga mode ini sudah jarang digunakan sekarang.dengan demikian pilihan terbaik pada Mode **ECP and EPPI.9**.

4.10.1. Mode -mode PORT Paralel dan Register Kontrol ECP

Sebagaimana telah dibicarakan sebelumnya, pilihan terbaik adalah menggunakan Mode **ECP and EPPI.9** dan menggunakan Register kontrol tambahan ECP untuk memilih beragam mode operasi. Register ECP ditandarisasi dibawah **Microsoft's Extended Capabilities Port Protocol and ISA Interface Standard**, sehingga kita (setidaknya) tidak memiliki masalah lagi yang berkaitan dengan standarisasi.

Pada saat diset Mode ECP, sebuah register baru muncul pada alamat dasar + 400h. Pada tabel 4.6 ditunjukkan ECR (*Extended Control Register*) yang dipetakan pada alamat dasar + 402h. Catatan: register ECP tidak akan tersedia untuk port-port dengan alamat dasar Ox3BC.

Pada tabel 4.6 ditunjukkan untuk ECR, yang perlu diperhatikan adalah 3 MSB (.bit 5, 6 dan 7) dari ECR yang digunakan untuk memilih mode operasi. Ada 7 kemungkinan mode operasi, namun tidak semua port mendukung semua mode tersebut. Mode EPP misalnya, tidak tersedia untuk beberapa port.

Mode -mode Operasi

Mode Standar	Mode ini menyebabkan port ECP bertingkah-laku seperti SPP (Standard Parallel Port) tanpa fungsi dwi-arah;
Mode Byte (Mode PS/2)	Seperti pada SPP dalam mode dwi-arah, bit-5 akan membuat port dalam mode terbalik (data -> komputer);
Mode Paralel (FIFO)	Pada mode ini, sembarang data yang dituliskan ke FIFO data akan terkirim ke piranti menggunakan SPP <i>handshake</i> . Perangkat keras akan membangkitkan sinyal-sinyal <i>handshaking</i> yang dibutuhkan. Berguna untuk alat-alat non-ECP seperti Pencetak (Printer). Anda dapat memperoleh

	beberapa fitur ECP seperti penyangga FIFO dan perangkat keras pembangkit sinyal <i>handshaking</i> namun dengan <i>SPP handshake</i> (bukan <i>ECP handshake</i>);
Mode FIFO ECP	Mode standar untuk ECP. Mode ini menggunakan <i>ECP handshake</i> . Pada saat menggunakan Mode ECP dan register ECR diset ke mode FIFO ECP, maka register SPP akan hilang!
Mode EPP 'terkadang)	Untuk mengaktifkan mode EPP, jika tersedia. Jika pada BIOS diset ke Mode <i>ECP and EPPI.x</i> , maka EPPI.x akan tersedia. Di bawah standarisasi Microsoft's Extended Capabilities Port Protocol and ISA Interface Standard mode; ini digunakan untuk Para Pemasok (<i>Vendor</i>)-.
Tercadang	Di bawah standarisasi Microsoft's Extended Capabilities Port Protocol and ISA Interface Standard mode ini digunakan untuk Para Pemasok (<i>Vendor</i>),
Mode FIFO Test	Pada mode ini, sembarang data yang dituliskan ke Register test FIFO akan ditempatkan pada FIFO dan sembarang data yang terbaca dari register Test FIFO akan dibaca melalui penyangga FIFO. Bit status FIFO penuh/kosong sesuai dengan kondisi sebenarnya.
Mode Konfigurasi	Pada mode ini, dua register konfigurasi, <i>cnfgA</i> dan <i>cnfgB</i> akan tersedia pada alamatnya masing-masing

Tabel 4.6. ECR

Bit	Function
7-5	Pemilihan Mode Operasi
000	Mode Standar
001	Mode Byte
010	Mode Port Pararel FIFO
011	Mode ECP FIFO
100	Mode EPP
101	Tercadang
110	Mode FIFO test
111	Mode Konfigurasi
4	Bit ECP Interrupt
3	Bit Aktivasi DMA
2	Bit Pelayanan ECP
1	FIFO penuh
0	FIFO KOsong

Jika Anda berada dalam mode ECP (melalui BIOS) atau kartu pencetak Anda diset menggunakan ECP maka ada baiknya untuk menginisialisasi mode dari port ECP Anda ke suatu kondisi awal sebelum digunakan. Jika Anda ingin menggunakan mode SPP, maka set port ke Mode Standar untuk pertama kalinya, jangan punya anggapan bahwa port siap dalam mode SPP.

Untuk beberapa mode, register SPP dapat hilang atau tidak bekerja dengan baik. Jika Anda menggunakan SPP, maka set ECR ke Mode Standar!