

14. ANTAR MUKA I/O STANDAR

Bagian sebelumnya menekankan adanya beberapa desain alternatif untuk bus komputer. Keanekaragaman ini berarti bahwa perangkat I/O yang sesuai dengan suatu sirkuit antar muka yang cocok untuk satu komputer mungkin tidak dapat digunakan untuk komputer lain. Antar muka yang berbeda mungkin harus didesain untuk setiap kombinasi perangkat I/O dan komputer, sehingga menghasilkan banyak antar muka yang berbeda. Solusi yang paling praktis adalah dengan mengembangkan sinyal dan protokol antar muka standar.

Pada titik ini akan sangat menolong untuk memahami bagaimana sistem komputer dibangun. Misalnya personal computer biasa, menyertakan printed circuit board yang disebut motherboard. Board ini merupakan rumah bagi chip prosesor, memori utama, dan beberapa antar muka I/O. Board ini juga memiliki beberapa konektor yang dapat digunakan untuk mencolokkan antar muka tambahan.

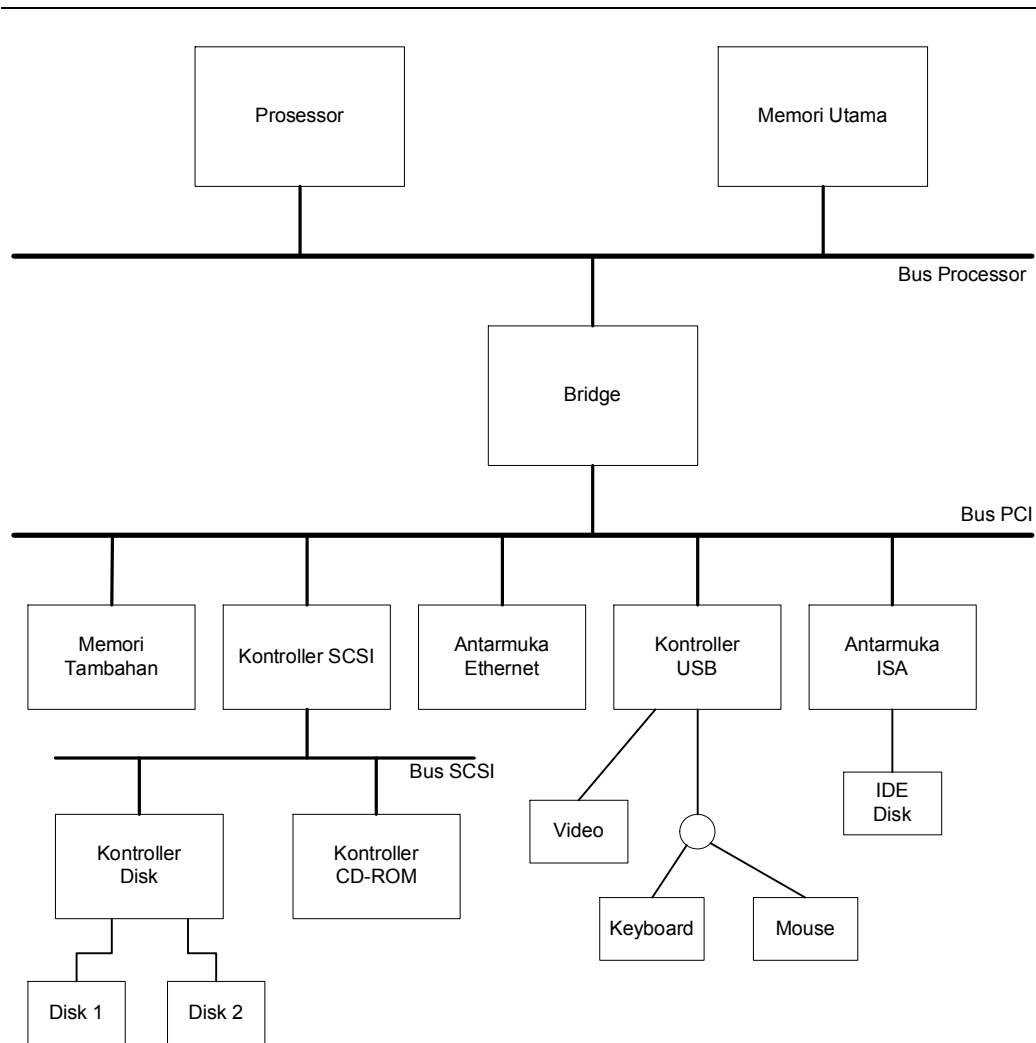
Bus prosesor adalah bus yang didefinisikan oleh sinyal pada chip prosesor tersebut. Perangkat yang memerlukan koneksi dengan kecepatan sangat tinggi ke prosesor, seperti memori utama, dapat dihubungkan langsung ke bus ini. Untuk alasan elektrik, hanya beberapa perangkat yang dapat dihubungkan dengan cara ini. Motherboard biasanya menyediakan bus lain yang dapat mendukung lebih banyak perangkat. Dua bus tersebut diinterkoneksi oleh suatu sirkuit, yang kita sebut bridge, yang mentranslasikan sinyal dan protokol satu bus menjadi yang lainnya. Perangkat yang dihubungkan ke bus ekspansi tampak pada prosesor seakan perangkat tersebut terkoneksi langsung ke bus prosesor. Satu-satunya perbedaan adalah sirkuit bridge menimbulkan jeda singkat dalam transfer data antara prosesor dan perangkat tersebut.

Tidak mungkin menetapkan standar seragam untuk bus prosesor. Struktur bus ini terikat erat dengan arsitektur prosesor. Juga tergantung pada karakteristik elektrik chip prosesor, seperti kecepatan clock. Bus ekspansi tidak termasuk dalam batasan ini, dan oleh karena itu dapat menggunakan skema signaling terstandarisasi. Sejumlah standar telah dikembangkan. Beberapa telah berkembang secara default, pada saat desain tertentu sukses secara komersial. Misalnya, IBM mengembangkan suatu bus yang disebut ISA (Industry Standard

Architecture) untuk personal computer mereka yang saat itu dikenal sebagai PC AT. Popularitas komputer tersebut mendorong produsen lain membuat antar muka ISA-compatible untuk perangkat UOnya, sehingga menjadikan ISA standar de facto.

Beberapa standar telah berkembang melalui usaha kerja sama industrial, bahkan diantara perusahaan pesaing dikarenakan keinginan bersama dalam memiliki produk yang kompatibel. Pada beberapa kasus, organisasi seperti IEEE (Institute of Electrical and Electronic Engineers), ANSI (American National Standards Institute), atau badan internasional seperti ISO (International Standards Organization) telah menyetujui standar tersebut dan memberinya status resmi.

Pada bagian ini kita menyajikan tiga standar bus yang digunakan secara luas, PCI (Peripheral Computer Interconnect), SCSI (Small Computer System Interface), dan USB (Universal Serial Bus). Cara standar ini digunakan dalam sistem komputer biasa diilustrasikan pada Gambar 14.1. Standar PCI mendefinisikan bus ekspansi pada motherboard. SCSI dan USB digunakan untuk menghubungkan perangkat tambahan, di dalam dan di luar kotak komputer. Bus SCSI adalah bus paralel high-speed yang ditujukan untuk perangkat, seperti disk dan video display. Bus USB menggunakan transmisi serial untuk menyesuaikan dengan kebutuhan peralatan, dari keyboard hingga kontrol game hingga koneksi internet. Gambar tersebut menunjukkan sirkuit antar muka yang memungkinkan dihubungkannya perangkat yang kompatibel dengan standar ISA awal, seperti disk IDE (Integrated Device Electronics) populer. Juga menunjukkan koneksi ke Ethernet. Ethernet digunakan secara luas dalam local area network, menyediakan koneksi high-speed diantara komputer dalam bangunan atau kampus universitas.



Gambar 14.1. Contoh sistem computer yang menggunakan standar antarmuka yang berbeda

Suatu komputer mungkin menggunakan lebih dari satu standar bus. Komputer Pentium biasa memiliki bus PCI dan bus ISA, sehingga menyediakan rentang pilihan perangkat yang luas bagi user.

14.1. BUS PERIPHERAL COMPONENT INTERCONNECT (PCI)

Bus PCI adalah contoh yang baik dari sistem bus yang muncul dari kebutuhan standarisasi. Bus tersebut mendukung fungsi yang terdapat dalam bus prosesor tetapi dalam format terstandarisasi yang lepas dari prosesor tertentu. Perangkat yang terkoneksi ke bus PCI tampak bagi prosesor seakan dihubungkan secara langsung ke bus prosesor. Perangkat tersebut diberi alamat dalam ruang alamat memori pada prosesor.

PCI mengikuti suatu rangkaian standar bus yang sebelumnya digunakan terutama pada IBM PC. PC awal menggunakan bus 8-bit XT, yang sinyalnya sangat mirip dengan prosesor 80x86 Intel. Setelahnya, bus 16-bit yang digunakan pada komputer PC AT dikenal sebagai bus ISA. Versi extended 32-bit-nya dikenal sebagai bus EISA. Bus lain yang dikembangkan pada tahun delapan puluhan dengan kemampuan serupa adalah Microchannel yang digunakan dalam IBM PC dan NuBus yang digunakan dalam komputer Macintosh.

PCI dikembangkan sebagai bus low-cost yang sangat processor dependent. Desainnya mengantisipasi tuntutan bandwidth bus yang berkembang sangat cepat untuk mendukung disk high-speed dan perangkat grafik dan video, dan juga kebutuhan khusus terhadap sistem multiprosesor. Akibatnya, PCI masih populer sebagai standar industri hampir satu dekade setelah diperkenalkan pertama kali pada tahun 1992.

Fitur penting yang dirintis oleh PCI adalah kemampuan plug-and-play untuk menghubungkan perangkat I/O. Untuk menghubungkan perangkat baru, user cukup menghubungkan board antar muka perangkat ke bus tersebut. Software menangani bagian selanjutnya.

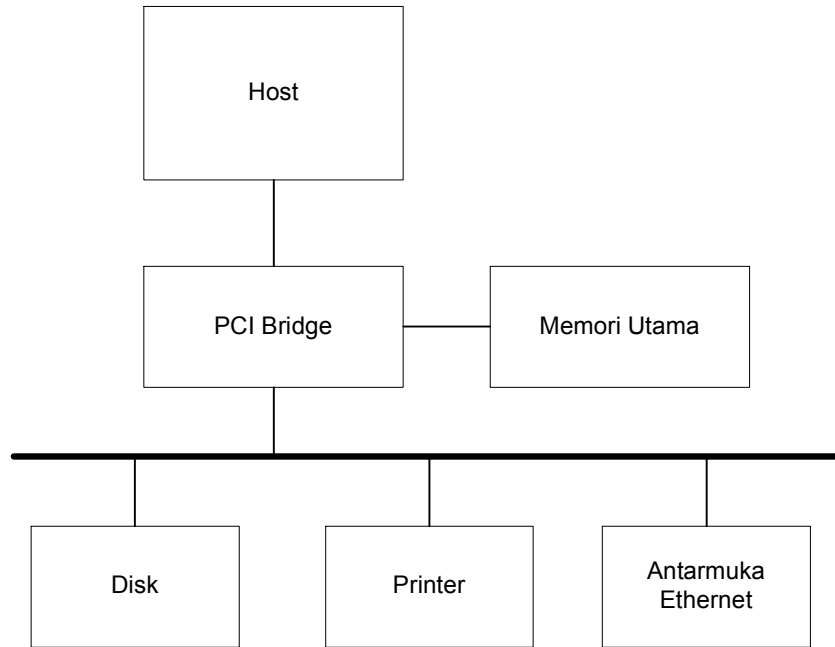
14.1.1. TRANSFER DATA

Dalam komputer saat ini, kebanyakan transfer memori melibatkan burst data bukannya hanya satu word. Alasannya adalah prosesor modern menyertakan memori cache. Data ditransfer antara cache dan memori utama tiap kali dalam burst beberapa word. Word yang terlibat dalam transfer semacam itu disimpan pada lokasi memori yang berurutan. Pada saat prosesor (sebenarnya kontroler cache) menetapkan suatu alamat dan me-request operasi baca dari memori utama, memori merespon dengan mengirim rangkaian word data mulai dari alamat tersebut. Serupa pula, selama operasi tulis, prosesor mengirim alamat memori diikuti oleh suatu rangkaian word data, yang akan ditulis dalam lokasi memori yang berurutan mulai dari alamat itu. PCI didesain terutama untuk mendukung mode operasi ini. Operasi baca atau tulis yang melibatkan word tunggal hanya diperlakukan sebagai burst suatu panjang.

Bus mendukung tiga ruang alamat mandiri: memori, I/O, dan konfigurasi. Dua yang pertama adalah self-explanatory. Ruang alamat I/Q dimaksudkan untuk penggunaan dengan prosesor, seperti Pentium, yang memiliki ruang alamat I/O terpisah. Akan tetapi, sebagaimana yang dinyatakan dalam Bab 3, desainer sistem dapat memilih untuk menggunakan memory-mapped I/O sekalipun tersedia ruang alamat I/O terpisah. Sebenarnya, ini adalah pendekatan yang direkomendasikan oleh standar PCI untuk kompatibilitas yang lebih luas. Ruang konfigurasi dimaksudkan untuk memberi PCI kemampuan plug-and-play-nya, sebagaimana yang akan kita jelaskan sebentar lagi. Perintah 4-bit bersama dengan alamat tersebut mengidentifikasi mana dari ketiga ruang tersebut yang digunakan dalam suatu operasi transfer data.

Gambar 14.1 menunjukkan memori utama komputer yang dihubungkan langsung ke bus prosesor. Pengaturan alternatif yang sering digunakan dengan bus PCI ditampilkan pada Gambar 13.3. Bridge PCI menyediakan koneksi fisik terpisah untuk memori utama. Untuk alasan elektrik, bus tersebut mungkin dibagi lebih lanjut menjadi segmen-segmen yang dihubungkan melalui bridge. Akan tetapi, tanpa memperhatikan segmen bus mana yang dihubungkan dengan suatu perangkat, perangkat tersebut masih dapat dipetakan ke dalam ruang alamat memori prosesor.

Konvensi signaling pada bus PCI mirip dengan yang digunakan pada Gambar 13.3. Pada gambar tersebut, kita mengasumsikan bahwa master mempertahankan informasi alamat pada bus hingga transfer data selesai. Tetapi, hal ini tidak perlu dilakukan. Alamat tersebut diperlukan hanya hingga dipilih oleh slave. Slave dapat menyimpan alamat tersebut dalam buffer internal-nya. Jadi, alamat tersebut diperlukan dalam bus hanya selama satu clock cycle, kemudian membebaskan jalur alamat untuk digunakan mengirim data pada clock cycle selanjutnya. Hasilnya adalah reduksi biaya yang cukup signifikan karena jumlah kabel pada bus adalah faktor biaya yang penting. Pendekatan ini digunakan dalam bus PCI.



Gambar 14.2 Penggunaan bus PCI dalam sistem komputer.

Nama	Fungsi
CLK	Clock 33-MHz atau 66-MHz.
FRAME#	Dikirim oleh initiator untuk mengindikasikan durasi suatu transaksi.
AD	32 jalur alamat/ data, yang dapat secara opsional ditingkatkan ke 64.
C/BE#	4 jalur perintah/ byte-enable (8 untuk bus 64-bit).
IRDY#, TRDY#	Sinyal initiator-ready dan Target-ready.
DEVSEL#	Respon dari perangkat yang mengindikasikan bahwa telah mengenali alamatnya dan siap untuk transaksi transfer data.
IDSEL#	Initialization Device Select

Tabel 14.1 Sinyal transfer data pada bus PCI.

Pada setiap waktu, satu perangkat menjadi bus master. Perangkat tersebut memiliki hak untuk menginisiasi transfer data dengan mengeluarkan perintah baca dan tulis. Master disebut initiator dalam istilah PCI. Master adalah

prosesor atau kontroler DMA. Perangkat yang dituju yang merespon perintah baca atau tulis disebut target.

14.1.2. KONFIGURASI PERANGKAT

Pada saat perangkat I/O dihubungkan dengan suatu komputer, beberapa aksi diperlukan untuk mengkonfigurasi perangkat dan software yang berkomunikasi dengannya. Misalnya, kartu antar muka perangkat biasa untuk bus ISA memiliki sejumlah jumper atau switch yang harus di-set oleh user untuk memilih opsi tertentu. Pada saat perangkat telah terhubung, software perlu mengetahui alamat perangkat tersebut. Mungkin juga perlu mengetahui karakteristik perangkat yang relevan, seperti kecepatan link transmisi, apakah menggunakan bit paritas, dan lain sebagainya.

PCI menyederhanakan proses ini dengan menggabungkan dalam tiap antar muka perangkat suatu memori ROM konfigurasi sederhana yang menyimpan informasi tentang perangkat tersebut. ROM konfigurasi semua perangkat dapat diakses dalam ruang alamat konfigurasi.

Software inisialisasi PCI membaca ROM ini kapanpun sistem dinyalakan atau di-reset. Pada tiap kasus, software menetapkan apakah perangkat tersebut adalah printer, keyboard, antar muka Ethernet, atau kontroler disk. Software dapat mempelajari lebih jauh tentang berbagai pilihan dan karakteristik perangkat.

Perangkat tersebut diberi alamat selama proses inisialisasi. Hal ini berarti selama operasi konfigurasi bus, perangkat tidak dapat diakses berdasarkan alamatnya, seakan perangkat tersebut belum memiliki alamat. Karenanya, ruang alamat konfigurasi menggunakan mekanisme yang berbeda. Tiap perangkat memiliki sinyal input yang disebut Initialization Device Select, IDSEL#. Selama operasi konfigurasi, sinyal inilah yang menyebabkan perangkat tersebut dipilih, bukannya alamat yang diterapkan pada input AD perangkat. Motherboard dimana konektor perangkat dicolokkan, biasanya mengharuskan pin IDSEL# tiap perangkat dihubungkan ke salah satu dari 21 jalur alamat atas, AD 11 hingga AD31. Karenanya, suatu perangkat dapat dipilih untuk operasi konfigurasi dengan mengeluarkan perintah konfigurasi

dan alamat dimana jalur AD yang sesuai di-set ke 1 dan 20 jalur lainnya di-set ke 0. Jalur alamat yang lebih rendah, AD 10 hingga AD00, digunakan untuk menentukan tipe operasi dan untuk mengakses isi ROM konfigurasi perangkat. Pengaturan ini membatasi jumlah perangkat I/O sampai dengan 21.

Software konfigurasi memindai ke-21 lokasi dalam ruang alamat konfigurasi atau ruang memori. Perangkat tersebut kemudian mendapat suatu alamat dengan menuliskan alamat tersebut ke register perangkat yang sesuai. Software konfigurasi juga men-set parameter semacam prioritas interrupt perangkat. Bus PCI memiliki empat jalur interrupt-request. Dengan menulis ke dalam register konfigurasi perangkat, software menginstruksikan pada perangkat jalur mana yang dapat digunakannya untuk me-request suatu interrupt. Jika suatu perangkat meminta insialisasi, maka kode inialisasi disimpan dalam ROM pada antar muka perangkat. (Ini adalah ROM yang berbeda dari yang digunakan dalam proses konfigurasi.) Software PCI membaca kode ini dan mengeksekusinya untuk melakukan inialisasi yang diperlukan.

Proses tersebut membebaskan user dari keharusan untuk terlibat dalam proses konfigurasi. User cukup mencolokkan board antar muka dan menyalakan daya. Software menyelesaikan bagian lainnya. Perangkat tersebut siap digunakan.

Bus PCI telah mendapatkan popularitas yang luar biasa dalam dunia PC. Bus tersebut juga digunakan dalam banyak komputer lain, seperti SUN, untuk memanfaatkan perangkat I/O skala luas yang menggunakan antar muka PCI. Dalam kasus beberapa prosesor, seperti Compaq Alpha, sirkuit bridge PCI-processor dibangun pada chip prosesor tersebut, sehingga lebih menyederhanakan desain sistem dan pengepakan.

14.1.3. KARAKTERISTIK ELEKTRIK

Bus PCI telah didefinisikan untuk beroperasi dengan suplai daya 5- atau 3,3-V Motherboard mungkin didesain untuk beroperasi dengan kedua sistem signaling tersebut. Konektor pada board ekspansi didesain untuk memastikan bahwa konektor tersebut dapat dicolokkan hanya pada motherboard yang kompatibel.

14.2. BUS SCSI

Akronim SCSI merupakan singkatan dari Small Computer System Interface. Akronim tersebut mengacu pada bus standar yang didefinisikan oleh American National Standards Institute (ANSI) dengan nomor X3.131. Dalam spesifikasi asli standar tersebut, perangkat seperti disk dihubungkan ke komputer melalui kabel 50-wire, yang dapat mencapai panjang 25 meter dan dapat mentransfer data hingga kecepatan 5 megabyte/detik.

Standar bus SCSI telah mengalami banyak revisi, dan kemampuan transfer datanya telah meningkat sangat besar, hampir dua kali setiap dua tahun. SCSI-2 dan SCSI-3 telah didefinisikan, dan masing-masing memiliki beberapa opsi. Bus SCSI memiliki delapan jalur data, yang disebut narrow bus dan mentransfer data satu byte pada satu waktu. Sebagai alternatif, bus wide SCSI memiliki 16 jalur data dan mentransfer data 16 bit pada satu waktu. Terdapat pula beberapa opsi untuk skema signaling elektrik yang digunakan. Bus dapat menggunakan transmisi single-ended (SE), dimana tiap sinyal menggunakan satu wire, dengan common ground return untuk semua sinyal. Dalam opsi lain, digunakan signaling diferensial, dimana disediakan return wire terpisah untuk tiap sinyal. Dalam hal ini, dimungkinkan dua tingkat tegangan. Versi yang lebih awal menggunakan 5 V (tingkat TTL) dan dikenal sebagai High Voltage Differential (HVD). Yang lebih baru, versi 3,3V telah diperkenalkan dan dikenal sebagai Low Voltage Differential (LVD).

Karena berbagai pilihan ini, maka konektor SCSI dapat memiliki 50, 68, atau 80 pin. Kecepatan transfer maksimum dalam perangkat komersial yang saat ini tersedia bervariasi dari 5 megabyte/det hingga 160 megabyte/det. Versi terbaru dari standar tersebut diraksudkan untuk mendukung kecepatan transfer hingga 320 megabyte/det, dan 640 megabyte/det diantisipasi kemudian. Kecepatan transfer maksimum pada bus tertentu sering merupakan fungsi panjang kabel dan jumlah perangkat yang dihubungkan, dengan kecepatan lebih tinggi untuk kabel yang lebih pendek dan perangkat yang lebih sedikit. Untuk mencapai kecepatan transfer data puncak, panjang bus biasanya dibatasi hingga 1,6 m untuk signaling SE dan 12 m untuk signaling LVD. Akan tetapi produsen sering menyediakan bus expander khusus untuk menghubungkan perangkat yang jauh letaknya. Kapasitas

maksimum bus adalah 8 perangkat untuk narrow bus dan 16 perangkat untuk wide bus.

Perangkat yang dihubungkan ke SCSI bukanlah bagian dari ruang alamat prosesor seperti perangkat yang terhubung ke bus prosesor. Bus SCSI dihubungkan ke bus prosesor melalui kontroler SCSI, sebagaimana ditunjukkan pada Gambar 14.1. Kontroler ini menggunakan DMA untuk mentransfer paket data dari memori utama ke perangkat tersebut, atau sebaliknya. Suatu paket dapat berisi blok data, perintah dari prosesor ke perangkat, atau informasi status tentang perangkat tersebut.

Untuk mengilustrasikan operasi bus SCSI, marilah kita memperhatikan bagaimana bus tersebut digunakan bersama disk drive. Komunikasi dengan disk drive berbeda secara substansial dari komunikasi dengan memori utama. Data disimpan pada disk dalam blok yang disebut sector, dimana tiap sector dapat berisi beberapa ratus byte. Data tersebut mungkin tidak perlu disimpan dalam sector yang berbatasan. Beberapa sector dapat berisi data yang disimpan sebelumnya; yang lain mungkin rusak dan harus dilompati. Karenanya, request baca atau tulis dapat menghasilkan pengaksesan beberapa sector disk yang tidak harus berdekatan. Karena batasan gerakan mekanik disk, maka terdapat jeda yang panjang, dalam order beberapa milidetik, sebelum mencapai sector pertama ke atau dari mana data ditransfer. Kemudian, burst data ditransfer pada kecepatan tinggi. Jeda lain mungkin terjadi, diikuti oleh burst data. Request baca atau tulis tunggal dapat melibatkan beberapa burst semacam itu. Protokol SCSI didesain untuk memfasilitasi mode operasi ini.

Kontroler yang dihubungkan ke bus SCSI adalah salah satu dari dua tipe - initiator atau target. Initiator memiliki kemampuan untuk memilih target tertentu dan mengirim perintah yang menetapkan operasi yang akan dilakukan. Kontroler pada sisi prosesor, seperti kontroler SCSI pada Gambar 14.1, harus mampu beroperasi sebagai initiator. Kontroler disk beroperasi sebagai target. Kontroler tersebut menetapkan koneksi logika dengan target yang dimaksud. Setelah koneksi terbentuk, koneksi tersebut dapat ditangguhkan dan dipulihkan seperlunya untuk mentransfer perintah dan burst data. Pada saat koneksi tertentu ditangguhkan, perangkat lain dapat menggunakan bus tersebut untuk

mentransfer informasi. Kemampuan untuk menoverlap request transfer data ini adalah salah satu fitur utama dalam bus SCSI yang menghasilkan performa tinggi.

Transfer data pada bus SCSI selalu dikontrol oleh kontroler target. Untuk mengirim perintah ke target, suatu initiator me-request kontrol bus dan, setelah memenangkan arbitration, memilih kontroler yang akan diajak berkomunikasi dan menyerahkan kontrol bus padanya. Kemudian kontroler memulai operasi transfer data untuk menerima perintah dari initiator.

Marilah kita menganalisa operasi baca disk lengkap sebagai contoh. Dalam pembahasan ini, sekalipun kita menyatakan kontroler initiator sebagai melakukan suatu tindakan, sebaiknya diperjelas bahwa kontroler melakukan tindakan ini setelah menerima perintah yang tepat dari prosesor. Asumsikan bahwa prosesor ingin membaca blok data dari disk drive dan data tersebut disimpan dalam dua sektor disk yang tidak berbatasan. Prosesor mengirim perintah ke kontroler SCSI, yang menghasilkan rangkaian event berikut:

1. Kontroler SCSI, yang bertindak sebagai initiator, berjuang untuk mendapatkan kontrol bus.
2. Pada saat initiator memenangkan proses arbitration, initiator memilih kontroler target dan menyerahkan kontrol bus padanya.
3. Target memulai operasi output (dari initiator ke target); sebagai respon terhadap hal ini, initiator mengirim perintah yang menentukan operasi baca yang diminta.
4. Target, yang mengerti bahwa harus melakukan operasi disk seek terlebih dahulu, mengirim pesan ke initiator yang mengindikasikan akan menanggihkan sementara koneksi antara initiator dan target. Kemudian target membebaskan bus tersebut.
5. Kontroler target mengirim perintah ke disk drive untuk memindahkan head baca ke sektor pertama yang terlibat dalam operasi baca yang dimaksud. Kemudian membaca data yang disimpan dalam sektor tersebut dan menyimpannya dalam buffer data. Pada saat target siap mentransfer data ke initiator, target me-request kontrol bus. Setelah memenangkan arbitration,

target me-reselect kontroler initiator, sehingga memulihkan koneksi yang ditanggihkan.

6. Target mentransfer isi buffer data ke initiator dan kemudian menanggihkan lagi koneksi tersebut. Data ditransfer 8 atau 16 bit secara paralel, tergantung pada lebar bus.
7. Kontroler target mengirim perintah ke disk drive untuk melakukan operasi seek lainnya. Kemudian mentransfer isi sektor disk kedua ke initiator, seperti sebelumnya. Pada akhir transfer ini, koneksi logika antara dua kontroler tersebut diterminasi.
8. Pada saat kontroler initiator menerima data tersebut, maka kontroler menyimpannya dalam memori utama menggunakan pendekatan DMA.
9. Kontroler SCSI mengirim interrupt ke prosesor untuk memberitahu bahwa operasi yang diminta telah selesai.

Skenario ini menunjukkan bahwa pesan yang dipertukarkan melalui bus SCSI berada pada tingkat yang lebih tinggi daripada yang dipertukarkan melalui prosesor bus. Dalam konteks ini, "tingkat yang lebih tinggi" berarti bahwa pesan yang mengacu pada operasi yang meminta beberapa langkah penyelesaian, tergantung pada perangkat tersebut. Prosesor maupun kontroler SCSI harus waspada terhadap detail operasi perangkat tertentu yang terlibat dalam transfer data. Pada contoh sebelumnya, prosesor tidak perlu dilibatkan dalam operasi disk seek.

Standar bus SCSI mendefinisikan rentang pesan kontrol yang luas, yang dapat dipertukarkan antar-kontroler untuk menangani tipe perangkat I/O yang berbeda. Pesan juga didefinisikan untuk menangani berbagai error atau kondisi kegagalan yang mungkin muncul selama operasi perangkat atau transfer data.

14.2.1. SINYAL BUS

Sekarang kita mendeskripsikan operasi bus SCSI dari sudut pandang hardware. Sinyal bus diringkas pada Tabel 14.2. Untuk menyederhanakan kita menampilkan sinyal-sinyal untuk narrow bus (jalur 8 data). Perhatikanlah bahwa semua nama sinyal didahului dengan tanda minus. Tanda ini mengindikasikan bahwa sinyal tersebut aktif, atau jalur data setara dengan satu,

pada saat berada dalam keadaan low-voltage. Bus tersebut tidak memiliki jalur alamat. Sebaliknya, jalur data digunakan untuk mengidentifikasi kontroler bus yang terlibat selama proses selection atau reselection dan selama bus arbitration. Untuk narrow bus, terdapat delapan kontroler, bernomor 0 hingga 7, dan masing-masing dihubungkan dengan jalur data yang memiliki nomor sama. Suatu wide bus memuat hingga 16 kontroler. Kontroler meletakkan alamatnya sendiri atau alamat kontroler lain pada bus dengan mengaktifkan jalur data yang sesuai. Sehingga, dimungkinkan untuk memiliki lebih dari satu alamat pada bus pada satu waktu, seperti dalam proses arbitration yang akan kita deskripsikan selanjutnya. Pada saat koneksi antara dua kontroler telah terbentuk, maka tidak ada kebutuhan pengalamatan lagi, dan jalur data digunakan untuk membawa data.

Kategori	Nama	Fungsi
Data	-DB(0) hingga -DB(7)	Jalur data: membawa satu byte informasi selama transfer informasi dan mengidentifikasi perangkat selama fase arbitration, selection, dan reselection.
	-DB(P)	Bit paritas untuk bus data
	Fase	-BSY
-SEL		Selection: dinyatakan selama selection dan reselection
Tipe informasi	-C/D	Control/Data: dinyatakan selama transfer informasi kontrol (perintah, status, atau pesan)
	-MSG	Message: mengindikasikan bahwa informasi yang sedang ditransfer adalah suatu pesan (message)
Handshake	-REQ	Request: dinyatakan oleh target untuk me-request siklus transfer data
	-ACK	Acknowledge: dinyatakan oleh initiator pada saat telah menyelesaikan suatu operasi transfer data
Arah transfer	-I/O	Input/Output: dinyatakan untuk mengindikasikan operasi input (relatif terhadap initiator)

Lainnya	-ATN	Attention: dinyatakan oleh initiator pada saat initiator
	-RST	Reset: menyebabkan semua kontrol perangkat disconnect dari bus dan mengasumsikan keadaan start-up-nya

Tabel 14.2 Sinyal bus SCSI.

Fase utama yang terlibat dalam operasi bus SCSI adalah arbitration, selection, transfer informasi, dan reselection. Sekarang kita menganalisa fase-fase tersebut.

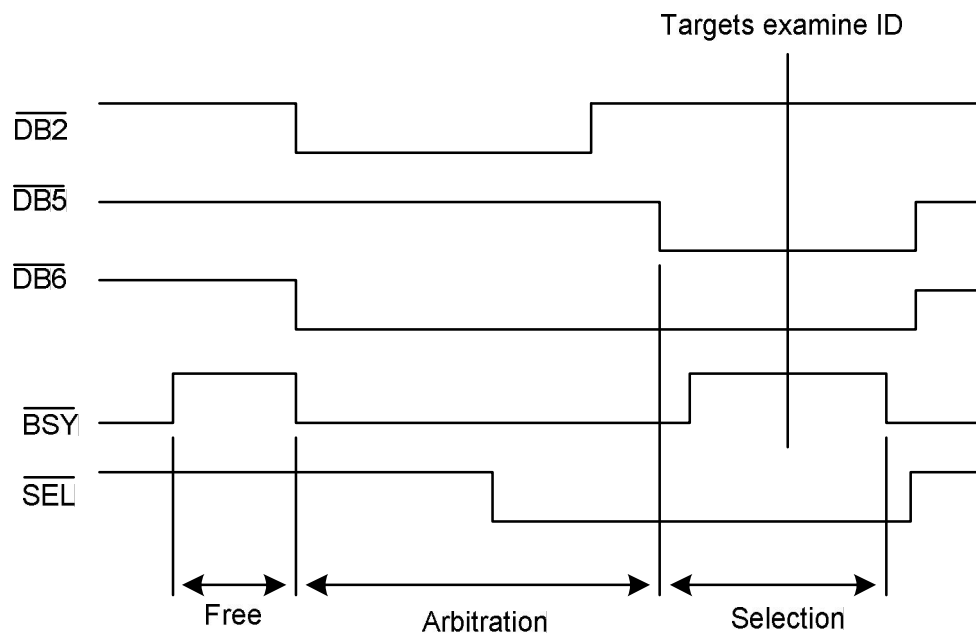
14.2.2. ARBITRATION

Bus bebas pada saat sinyal -BSY berada pada keadaan inactive (high-voltage). Kontroler apapun dapat me-request penggunaan bus tersebut pada saat bus berada dalam keadaan ini. Karena dua atau lebih kontroler dapat menghasilkan request pada saat yang sama, maka harus diterapkan skema arbitration. Kontroler me-request bus tersebut dengan menyatakan sinyal -BSY dan dengan menyatakan jalur data yang berhubungan dengannya untuk mengidentifikasi dirinya. Bus SCSI menggunakan skema arbitration terdistribusi sederhana. Skema ini diilustrasikan dengan contoh pada Gambar 14.3, dimana kontroler 2 dan 6 me-request penggunaan bus tersebut secara bersamaan.

Tiap kontroler pada bus tersebut ditempatkan pada prioritas tetap, dengan kontroler 7 mendapatkan prioritas tertinggi. Pada saat -BSY aktif, semua kontroler yang me-request bus tersebut menganalisa jalur data dan menetapkan apakah perangkat dengan prioritas yang lebih tinggi me-request bus tersebut pada saat yang bersamaan. Kontroler yang menggunakan jalur bernomor paling tinggi menyadari telah memenangkan proses arbitration. Semua kontroler lain disconnect dari bus tersebut dan menunggu sinyal -BSY menjadi tidak aktif lagi.

Dalam Gambar 14.3, kita telah mengasumsikan bahwa kontroler 6 adalah suatu initiator yang ingin membentuk koneksi dengan kontroler 5. Setelah

memenangkan arbitration, kontroler 6 melanjutkan ke fase selection, dimana kontroler tersebut mengidentifikasi target.



Gambar 14.3. Arbitration dan selection pada bus scsi. Perangkat 6 memenangkan arbitration dan memilih perangkat 2

14.2.3. SELECTION

Setelah memenangkan arbitration, kontroler 6 melanjutkan menyatakan -BSY dan -DB6 (alamatnya). Kontroler tersebut mengindikasikan ingin memilih kontroler 5 dengan menyatakan -SEL dan kemudian jalur-DBS. Kontroler lain yang telah terlibat dalam fase arbitration, seperti kontroler 2 pada gambar tersebut, harus berhenti menggunakan jalur data pada saat jalur -SEL aktif, jika kontroler 2 belum melakukannya. Setelah meletakkan alamat kontroler target pada bus tersebut, maka initiator melepaskan jalur-BSY

Kontroler target yang dipilih merespon dengan menyatakan -BSY Hal ini menunjukkan pada initiator bahwa koneksi yang di-request telah terbentuk, sehingga initiator dapat menghapus informasi alamat dari jalur data. Proses selection telah selesai, dan kontroler target (kontroler 5) menyatakan -BSY Dari titik ini, kontroler 5 memiliki bus kontrol, sebagaimana diperlukan untuk fase transfer informasi.

14.2.4. TRANSFER INFORMASI

Informasi yang ditransfer antara dua kontroler dapat berisi perintah dari initiator ke target, respon status dari target ke initiator, atau data yang sedang ditransfer ke atau dari perangkat I/O. Signaling handshake digunakan untuk mengontrol transfer informasi, dengan kontroler target berperan sebagai bus master. Sinyal -REQ dan -ACK menggantikan sinyal Master-ready dan Slave-ready pada Gambar 13.4 dan 13.5. Target menyatakan -I/O selama operasi input (target ke initiator), dan menyatakan -C/D untuk mengindikasikan bahwa informasi yang sedang ditransfer adalah perintah atau respon status bukan data.

Kita menekankan bahwa versi high-speed bus SCSI menggunakan teknik yang dikenal sebagai double-edge clocking atau Double Transitions (DT). Pada Gambar 13.4 dan 13.5, tiap transfer data memerlukan transisi high-to-low yang diikuti dengan transisi low-to-high pada dua sinyal handshake. Double-edge clocking berarti data ditransfer pada sisi menanjak dan menurun sinyal-sinyal tersebut, sehingga menggandakan kecepatan transfer.

Pada akhir transfer tersebut, kontroler target melepaskan sinyal -BSY, sehingga membebaskan bus untuk dapat digunakan oleh perangkat lain. Selanjutnya, kontroler target membentuk ulang koneksi ke kontroler initiator pada saat kontroler target siap mentransfer lebih banyak data. Hal ini dilakukan dalam operasi reselection yang dideskripsikan selanjutnya.

14.2.5. RESELECTION

Pada saat koneksi logika ditangguhkan dan target siap untuk memulihkannya, maka target harus terlebih dahulu memperoleh kontrol bus. Target memulai siklus arbitration, dan setelah memenangkan arbitration, target memilih kontroler initiator dengan cara yang tepat sama dengan yang dideskripsikan di atas. Tetapi dengan peranan target dan initiator dibalik, initiator sekarang menyatakan -BSY. Sebelum transfer data dapat dimulai, initiator harus menyerahkan kontrol ke target. Hal ini dilakukan dengan kontroler target menyatakan -BSY setelah memilih initiator. Sementara itu, initiator menunggu sebentar setelah dipilih untuk memastikan bahwa target telah menyatakan -BSY, dan kemudian melepaskan

jalur -BSY Koneksi antara dua kontroler telah terbentuk, dengan target mengontrol bus sebagaimana yang diperlukan untuk melakukan transfer data.

Skema signaling bus yang dideskripsikan di atas menyediakan mekanisme yang diperlukan bagi dua kontroler untuk membentuk koneksi logika dan mempertukarkan pesan. Koneksi tersebut dapat ditanggihkan dan dibentuk ulang setiap saat. Standar SCSI mendefinisikan struktur dan isi berbagai tipe paket yang dipertukarkan kontroler untuk menangani berbagai situasi yang berbeda. Initiator menggunakan paket tersebut untuk mengirim perintah yang diterimanya dari prosesor ke target. Target merespon dengan informasi status, dan operasi transfer data. Operasi transfer data tersebut dikontrol oleh target, karena target-lah yang mengetahui tersedianya data, kapan menanggihkan dan membentuk ulang koneksi, dan lain sebagainya.

14.3. Universal Serial Bus

Sinergi antara komputer dan komunikasi adalah jantung revolusi teknologi informasi saat ini. Sistem komputer modern tampaknya melibatkan berbagai variasi perangkat seperti keyboard mikrofon, kamera, speaker, dan perangkat display.

USB mendukung dua kecepatan operasi, disebut low-speed (1,5 megabit/det) dan fullspeed (12 megabit/det). Revisi terbaru pada spesifikasi bus (USB 2.0) memperkenalkan kecepatan operasi ketiga, disebut high-speed (480 megabit/det). USB dengan cepat memperoleh pengakuan di pasaran, dan dengan tambahan kemampuan high-speed menjadikannya sebagai pilihan metode interkoneksi bagi sebagian besar perangkat komputer.

USB didesain untuk memenuhi beberapa tujuan utama:

- Menyediakan sistem interkoneksi yang sederhana, low-cost, dan mudah digunakan yang dapat mengatasi kesulitan karena terbatasnya jumlah port I/O pada suatu komputer
- Mengakomodasi karakteristik transfer data skala luas untuk perangkat U0, termasuk koneksi telepon dan Internet
- Meningkatkan kenyamanan user melalui mode operasi plug-and-play

Kita akan menguraikan tujuan tersebut sebelum membahas detail teknis USB.

14.3.1. BATASAN PORT

Port paralel dan serial menyediakan titik koneksi general purpose yang dapat digunakan untuk menghubungkan berbagai perangkat dari low- hingga medium-speed ke komputer. Untuk alasan kepraktisan, hanya beberapa port disediakan pada komputer biasa. Untuk menambahkan port baru, user harus membuka kotak komputer untuk mendapatkan akses ke bus ekspansi internal dan menginstal kartu antar muka baru. User juga perlu mengetahui bagaimana untuk mengkonfigurasi perangkat dan software tersebut. Tujuan USB adalah untuk memungkinkan menambahkan sebanyak mungkin perangkat ke sistem komputer setiap saat, tanpa membuka kotak komputer.

14.3.2. KARAKTERISTIK PERANGKAT

Jenis perangkat yang dapat dihubungkan ke komputer meliputi fungsionalitas dalam skala luas. Batasan kecepatan, volume, dan timing berhubungan dengan transfer data ke dan dari perangkat semacam itu sangat bervariasi.

Dalam hal keyboard, satu byte data dihasilkan setiap kali tombol ditekan, yang dapat terjadi setiap saat. Data tersebut ditransfer ke komputer secara langsung. Karena event penekanan tombol tidak disinkronisasi ke event lain dalam sistem komputer, maka data yang dihasilkan oleh keyboard disebut asynchronous. Lagi pula, kecepatan data dihasilkan cukup rendah. Kecepatan tersebut dibatasi oleh kecepatan operator manusia hingga sekitar 100 byte per detik, yaitu kurang dari 1000 bit per detik.

Berbagai perangkat sederhana yang dapat dipasang ke komputer menghasilkan data dengan sifat yang sama - low speed dan asynchronous. Contoh yang bagus adalah mouse komputer dan kontrol dan manipulator yang digunakan dengan video game.

Marilah kita membahas dengan analogi lain. Banyak komputer memiliki mikrofon baik yang dipasang secara eksternal maupun yang built-in. Suara yang diterima mikrofon menghasilkan sinyal listrik analog, yang harus diubah

ke bentuk digital sebelum dapat ditangani oleh komputer. Hal ini dicapai dengan melakukan sampling sinyal analog secara periodik. Untuk tiap sample, analog-to-digital (A/D) converter menghasilkan n-bit bilangan yang mewakili besar sample. Jumlah bit, n, dipilih berdasarkan presisi yang diinginkan untuk mewakili tiap sample. Kemudian pada saat data tersebut dikirim ke speaker, digital-to-analog (D/A) converter digunakan untuk memulihkan sinyal analog dari format digital.

Proses sampling menghasilkan arus sample terdigitalisasi kontinyu yang tiba pada interval teratur, tersinkronisasi dengan sampling clock. Arus data semacam itu disebut isochronous, yang berarti bahwa event yang berurutan dipisahkan dengan periode waktu yang setara.

Sinyal harus di-sample secara cukup cepat untuk merekam komponen frekuensi tertingginya. Secara umum, jika sampling rate adalah s sample per detik, maka komponen frekuensi maksimum yang akan dicatat oleh proses sampling adalah $s/2$. Misalnya, suara manusia dapat direkam secara memadai dengan sampling rate 8kHz, yang akan merekam sinyal suara dengan frekuensi hingga 4kHz. Untuk suara dengan kualitas yang lebih tinggi, sebagaimana yang diperlukan dalam sistem musik, digunakan sampling rate yang lebih tinggi. Kecepatan standar untuk suara digital adalah 44,1 kHz. Tiap sample dinyatakan dengan 4 byte data untuk mengakomodasi rentang yang luas dalam volume suara (rentang dinamik) yang diperlukan untuk reproduksi suara kualitas tinggi. Hal ini menghasilkan kecepatan data sekitar 1,4 megabit/det.

Persyaratan penting dalam menangani suara atau musik ter-sample adalah mempertahankan timing yang tepat dalam proses sampling dan replay. Derajat tinggi pada jitter (variabilitas dalam sample timing) tidak dapat diterima. Karenanya, mekanisme transfer data antara sistem komputer dan musik harus mempertahankan jeda konsisten dari satu sample ke sample berikutnya. Jika tidak, maka diperlukan sirkuit complex buffering dan retiming. Sebaliknya, error sesekali dan sample yang hilang dapat ditoleransi. Error tersebut bisa tidak diketahui oleh pendengar atau dapat menyebabkan klik yang tidak kentara. Tidak ada mekanisme rumit yang diperlukan untuk memastikan pengiriman data yang benar-benar tepat.

Transfer data untuk image dan video memiliki persyaratan yang sama, tetapi pada bandwidth transfer data yang jauh lebih tinggi. Istilah bandwidth mengacu pada kapasitas transfer data total pada saluran komunikasi, yang diukur dalam unit yang sesuai seperti bit atau byte per detik. Untuk mempertahankan kualitas gambar pada televisi komersial, suatu image harus dinyatakan sekitar 160 kilobyte dan ditransmisikan 30 kali per detik, untuk bandwidth total 44 megabit/det. Image dengan kualitas yang lebih tinggi, seperti dalam HDTV (High Definition TV), memerlukan kecepatan yang lebih tinggi.

Perangkat penyimpanan besar seperti harddisk dan CD-ROM menggunakan persyaratan yang berbeda. Perangkat tersebut adalah bagian dari hierarki memori komputer, sebagaimana yang akan dibahas pada Bab 5. Koneksi perangkat tersebut ke komputer harus menyediakan bandwidth transfer data setidaknya 40 atau 50 megabit/det. Jeda pada order milidetik dimunculkan dalam mekanisme disk. Karenanya, jeda tambahan kecil yang dimunculkan pada saat mentransfer data ke atau dari komputer tidaklah penting, dan jitter tidak menjadi suatu masalah.

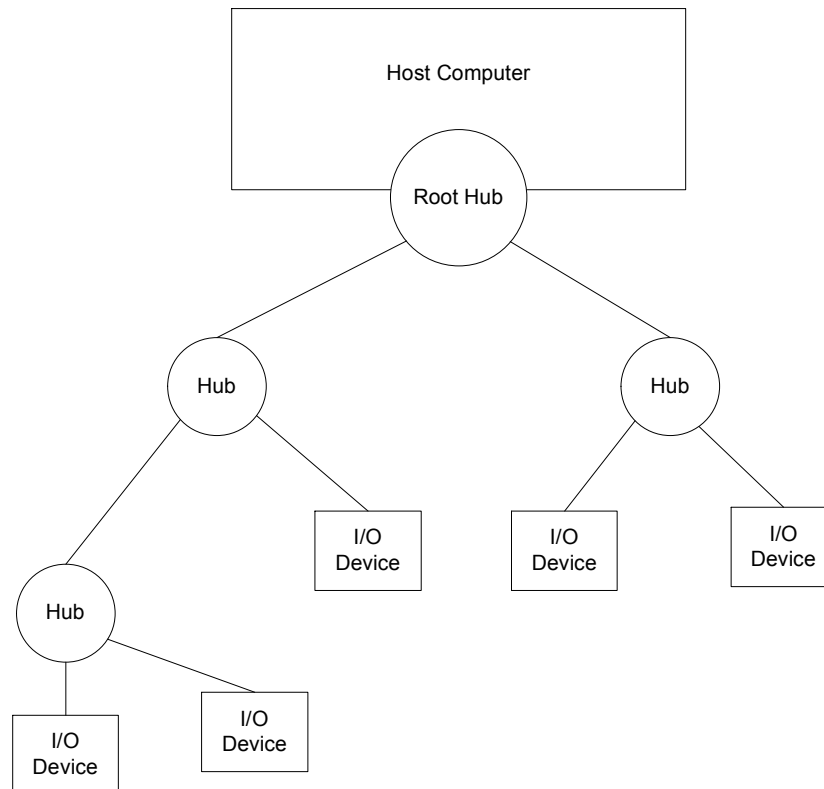
14.3.3. PLUG-AND-PLAY

Dengan komputer menjadi bagian dari kehidupan sehari-hari, keberadaannya menjadi semakin transparan. Misalnya, pada saat mengoperasikan sistem home theater, yang menyertakan setidaknya satu komputer, user tidak perlu mematikan komputer atau me-restart sistem untuk meng-connect atau disconnect suatu perangkat.

Fitur plug-and-play berarti bahwa perangkat baru, seperti speaker tambahan dapat dihubungkan kapanpun pada saat sistem beroperasi. Sistem mendeteksi keberadaan perangkat baru ini secara otomatis, mengidentifikasi device-driver software yang tepat dari perangkat baru ini dan fasilitas lain yang diperlukan untuk melayani perangkat tersebut, dan membentuk alamat dan koneksi logika untuk meng-enable komunikasinya.

Persyaratan plug-and-play memiliki banyak implikasi pada semua tingkat dalam sistem, dari hardware ke sistem operasi dan software aplikasi. Salah satu tujuan utama dari desain USB adalah menyediakan kemampuan plug-and-play.

14.3.4. ARSITEKTUR USB



Gambar 14.4. Struktur pohon Universal Serial Bus.

Pembahasan di atas menunjukkan pada kebutuhan akan sistem interkoneksi yang menggabungkan biaya rendah, fleksibilitas, dan bandwidth data-transfer yang tinggi. Juga, perangkat I/O dapat diletakkan pada jarak tertentu dari komputer yang terhubung dengannya. Persyaratan untuk bandwidth tinggi biasanya akan menyarankan wide bus yang membawa 8 bit, 16 bit, atau lebih secara paralel. Akan tetapi, sejumlah besar kabel meningkatkan biaya dan kerumitan dan menghasilkan ketidaknyaman bagi user. Juga sulit untuk mendesain wide bus yang membawa data untuk jarak jauh karena persoalan data skew. Jumlah skew

meningkat dengan bertambahnya jarak dan membatasi tingkat data yang digunakan.

Format transmisi serial telah dipilih untuk USB karena bus serial memenuhi persyaratan low-cost dan fleksibilitas. Informasi clock dan data di-encode bersama dan ditransmisikan sebagai sinyal tunggal. Karenanya, tidak ada batasan pada frekuensi clock atau jarak yang timbul dari data skew. Oleh karena itu, mungkin untuk menyediakan bandwidth transfer data tinggi dengan menggunakan frekuensi clock tinggi. Sebagaimana telah disebutkan sebelumnya, USB menawarkan tiga bit rate, mulai dari 1,5 hingga 480 megabit/det, untuk menyesuaikan dengan kebutuhan perangkat I/O yang berbeda. Untuk mengakomodasi sejumlah besar perangkat yang dapat ditambahkan atau dihilangkan setiap saat, USB memiliki struktur pohon seperti yang ditunjukkan pada Gambar 14.4. Tiap node pada pohon tersebut memiliki perangkat yang disebut hub, yang bertindak sebagai titik kontrol pertengahan antara host dan perangkat I/O. Pada akar pohon, root hub menghubungkan seluruh pohon ke komputer host. Daun pohon adalah perangkat I/O yang dilayani (misalnya keyboard, koneksi Internet, speaker, atau TV digital), yang disebut fungsi dalam terminologi USB. Untuk konsistensi dengan bagian selanjutnya dari pembahasan dalam buku ini, kita akan menyebut perangkat tersebut sebagai perangkat I/O.

Struktur pohon meng-enable banyak perangkat untuk dihubungkan pada saat hanya menggunakan link serial point-to-point sederhana. Tiap hub memiliki sejumlah port yang dapat dihubungkan dengan perangkat, termasuk dengan hub lain. Pada operasi normal, suatu hub meng-copy pesan yang diterimanya dari koneksi upstream ke semua port downstream-nya. Akibatnya, pesan yang dikirim oleh komputer host disebarkan ke semua perangkat I/O, tetapi hanya perangkat yang dituju yang akan merespon pesan tersebut. Dalam hal ini USB berfungsi dengan cara yang sama seperti bus tunggal. Akan tetapi, berbeda dengan bus tunggal, suatu pesan dari perangkat I/O dikirim oleh upstream menuju akar pohon dan tidak dilihat oleh perangkat lain. Karenanya, USB meng-enable host untuk berkomunikasi dengan perangkat I/O, tetapi

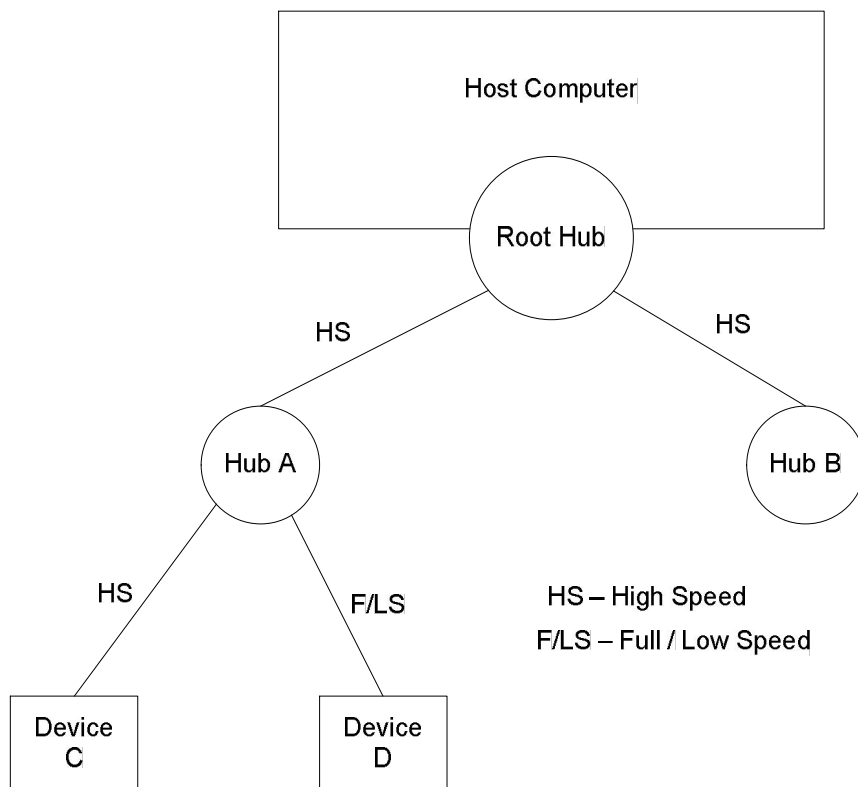
tidak meng-enable perangkat tersebut untuk berkomunikasi dengan perangkat lain.

Perhatikan bagaimana struktur pohon ini membantu memenuhi tujuan desain USB. Pohon tersebut memungkinkan USB menghubungkan sejumlah besar perangkat ke komputer melalui sedikit port (root hub). Pada saat yang sama, tiap perangkat I/O dihubungkan melalui koneksi serial point-to-point. Ini adalah pertimbangan yang penting dalam memfasilitasi fitur plug-and-play, yang akan kita perhatikan sebentar lagi. Juga karena pertimbangan transmisi listrik, transmisi data serial pada koneksi tersebut lebih mudah daripada transmisi paralel pada bus tunggal. Tingkat data yang jauh lebih tinggi dan kabel yang lebih panjang dapat digunakan.

USB beroperasi secara ketat pada basis polling. Suatu perangkat mengirim pesan hanya sebagai respon terhadap pesan poll dari host. Karenanya, pesan upstream tidak menghadapi konflik atau saling mengganggu satu dengan yang lain, sehingga tidak ada dua perangkat yang dapat mengirim pesan pada saat yang sama. Batasan ini memungkinkan hub menjadi perangkat low-cost yang sederhana.

Mode operasi yang dideskripsikan di atas digunakan untuk semua perangkat yang beroperasi baik pada kecepatan rendah ataupun kecepatan penuh. Akan tetapi, satu pengecualian dimunculkan oleh pengenalan operasi high-speed dalam USB versi 2.0. Pertimbangkanlah situasi pada Gambar 14.5. Hub A dihubungkan ke root hub oleh high-speed link. Hub ini melayani satu perangkat high-speed, C, dan satu perangkat low-speed, D. Biasanya, suatu pesan ke perangkat D akan dikirim pada kecepatan rendah dari root hub. Pada 1,5 megabit/det, bahkan pesan pendek sekalipun memerlukan beberapa puluh milidetik. Selama durasi pesan tersebut, tidak ada tranfser data lain yang bisa dilakukan, sehingga mengurangi keefektifan high-speed link dan menimbulkan jeda yang tidak dapat diterima untuk perangkat high-speed. Untuk mengurangi persoalan, protokol USB mensyaratkan agar suatu pesan yang ditransmisikan pada high-speed link selalu ditransmisikan pada kecepatan tinggi, bahkan pada saat receiver akhir adalah perangkat low-speed. Karenanya, pesan yang ditujukan untuk perangkat D dikirim pada kecepatan tinggi dari root hub ke

hub A, kemudian dilanjutkan pada kecepatan rendah ke perangkat D. Transfer yang terakhir tersebut akan memerlukan waktu lama, namun selama itu lalu lintas high-speed ke node lain dapat berlanjut. Misalnya, root hub dapat mempertukarkan beberapa pesan dengan perangkat C pada saat pesan low-speed dikirim dari hub A ke perangkat D. Selama periode ini, bus dikatakan di-split antara lalu lintas high-speed dan low-speed. Pesan ke perangkat D didahului dan diikuti oleh perintah khusus ke hub A untuk memulai dan mengakhiri masing-masing mode operasi split-traffic tersebut.



Gambar 14.5. Operasi Split bus

Standar USB menetapkan detail hardware interkoneksi USB dan juga organisasi dan persyaratan host software. Tujuan software USB adalah untuk menyediakan link komunikasi dua arah antara software aplikasi dan perangkat I/O. Link ini disebut pipe. Tiap data yang masuk pada satu ujung pipe dikirim ke ujung yang lain. Permasalahan seperti pengalamatan, timing, deteksi error, dan pemulihan ditangani oleh protokol USB.

Software yang mentransfer data ke atau dari perangkat I/O tertentu disebut driver perangkat untuk perangkat tersebut. Driver perangkat tergantung pada

karakteristik perangkat yang didukungnya. Karenanya, deskripsi yang lebih tepat tentang pipe USB adalah pipe menghubungkan perangkat I/O dengan driver perangkatnya. Pipe tersebut terbentuk pada saat suatu perangkat dihubungkan dan diberi alamat unik oleh software USB. Setelah terbentuk, data dapat mengalir melalui pipe tersebut setiap saat.

Kita sekarang akan menganalisa bagaimana perangkat diberi alamat dalam USB. Kemudian kita akan membahas berbagai cara berlangsungnya transfer data.

14.3.5. PENGALAMATAN

Pada pembahasan operasi input dan output sebelumnya, kita menjelaskan bahwa perangkat FO biasanya diidentifikasi dengan menetapkan alamat memori yang unik untuknya. Sebenarnya, suatu perangkat biasanya memiliki beberapa lokasi addressable untuk meng-enable software mengirim dan menerima informasi kontrol dan status dan untuk mentransfer data.

Pada saat USB dihubungkan ke komputer host, root hub-nya dipasangkan ke bus prosesor, dimana root hub tersebut tampak sebagai perangkat tunggal. Host software berkomunikasi dengan perangkat individu yang dipasang ke USB dengan mengirim paket informasi, yang diteruskan root hub ke perangkat yang tepat dalam pohon USB.

Tiap perangkat pada USB, baik hub atau perangkat I/O, diberi alamat 7-bit. Alamat ini lokal terhadap pohon USB dan tidak berhubungan dengan cara apapun ke alamat yang digunakan pada bus prosesor. Suatu hub dapat dihubungkan dengan berapapun jumlah perangkat atau hub lain, dan alamat ditetapkan secara berubah-ubah. Pada saat perangkat pertama kali dihubungkan ke hub, atau pada saat dinyalakan, alamatnya adalah 0. Hardware hub yang dihubungkan dengan perangkat ini mampu mendeteksi bahwa suatu perangkat telah terhubung dan mencatat fakta ini sebagai bagian dari informasi statusnya sendiri. Secara periodik, host mem-poll tiap hub untuk mengumpulkan informasi status dan mempelajari tentang perangkat baru yang mungkin telah ditambahkan atau di-disconnect. Pada saat host diberitahu bahwa perangkat baru telah terhubung, maka host menggunakan suatu rangkaian perintah untuk mengirim sinyal reset pada port hub yang sesuai, membaca informasi dari perangkat tersebut tentang

kemampuannya, mengirim informasi konfigurasi ke perangkat tersebut, dan memberi perangkat tersebut alamat USB yang unik. Setelah rangkaian ini diselesaikan perangkat memulai operasi normal dan merespon hanya pada alamat baru.

Prosedur koneksi awalan yang baru saja dideskripsikan adalah fitur utama yang membantu memberikan pada USB kemampuan plug-and-play-nya. Software host memiliki kontrol penuh terhadap prosedur tersebut. Software tersebut dapat mengetahui bahwa suatu perangkat telah dihubungkan, membaca informasi tentang perangkat tersebut, yang biasanya disimpan dalam memori read-only kecil dalam hardware perangkat, mengirim perintah yang akan mengkonfigurasi perangkat tersebut dengan meng-enable atau men-disable fitur atau kemampuan tertentu, dan akhirnya menetapkan alamat USB yang unik terhadap perangkat tersebut. Satu-satunya tindakan yang diperlukan dari user adalah mencolokkan perangkat tersebut ke port hub dan menyalakan switch dayanya.

Pada saat suatu perangkat dimatikan, maka dilakukan prosedur yang mirip. Hub yang sesuai melaporkan fakta ini ke software sistem USB, yang selanjutnya meng-update tabelnya. Tentu saja, jika perangkat yang telah di-disconnect adalah suatu hub, maka semua perangkat yang terhubung melalui hub tersebut harus juga dicatat sebagai disconnect. Software USB harus mempertahankan gambaran lengkap topologi bus dan perangkat yang terkoneksi sepanjang waktu.

Lokasi dalam perangkat ke atau dari mana transfer data dapat berlangsung, seperti register status, kontrol, dan data, disebut endpoint. Endpoint tersebut diidentifikasi dengan bilangan 4bit. Sebenarnya, tiap nilai 4-bit mengidentifikasi sepasang endpoint, satu untuk input dan satu untuk output. Sehingga suatu perangkat dapat memiliki hingga 16 pasang endpoint input/output. Pipe USB, yang merupakan saluran transfer data dua arah, dihubungkan ke satu pasangan tersebut. Pipe yang terhubung dengan endpoint nomor 0 tetap ada sepanjang waktu, termasuk segera setelah suatu perangkat dinyalakan atau di-reset. Ini adalah pipe kontrol yang digunakan software USB dalam prosedur power-on. Sebagai bagian dari prosedur itu, dapat dibentuk

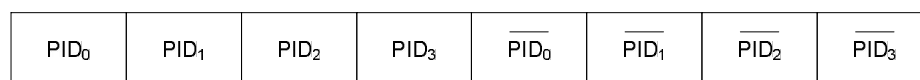
pipe lain menggunakan pasangan endpoint lain, tergantung pada kebutuhan dan kompleksitas perangkat tersebut. Bilangan endpoint 4-bit adalah bagian dari informasi pengalaman yang dikirim oleh host, sebagaimana yang akan kita lihat sebentar lagi.

14.3.6. Protokol USB

Semua informasi yang ditransfer melalui USB diatur di dalam paket, dimana suatu paket terdiri dari satu atau lebih byte informasi. Terdapat banyak tipe paket yang melakukan berbagai fungsi kontrol. Kita mengilustrasikan operasi USB dengan memberikan beberapa contoh tipe paket utama dan menunjukkan bagaimana paket tersebut digunakan.

Informasi yang ditransfer pada USB dapat dibagi menjadi dua kategori besar: kontrol dan data. Paket kontrol melakukan tugas seperti pengalaman perangkat untuk menginisiasi transfer data, memberitahu bahwa data telah diterima dengan tepat, atau mengindikasikan error. Paket data membawa informasi yang dikirim ke suatu perangkat. Misalnya, data input dan output ditransfer di dalam paket data.

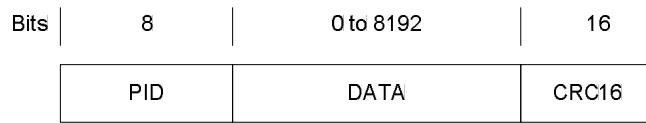
Suatu paket yang terdiri dari satu field atau lebih berisi berbagai jenis informasi yang berbeda. Field pertama dari tiap paket disebut packet identifier, PID, yang mengidentifikasi tipe paket tersebut. Terdapat empat bit informasi dalam field ini, tetapi bit tersebut ditransfer dua kali. Pertama kali bit tersebut dikirim dengan nilai sebenarnya, dan kedua kali dengan tiap bit dikomplementasi, sebagaimana ditunjukkan pada Gambar 14.6a. Hal ini meng-enable perangkat penerima untuk memverifikasi bahwa byte PID telah diterima dengan tepat.



(a) Packet identifier field



(b) Token packet, IN atau OUT

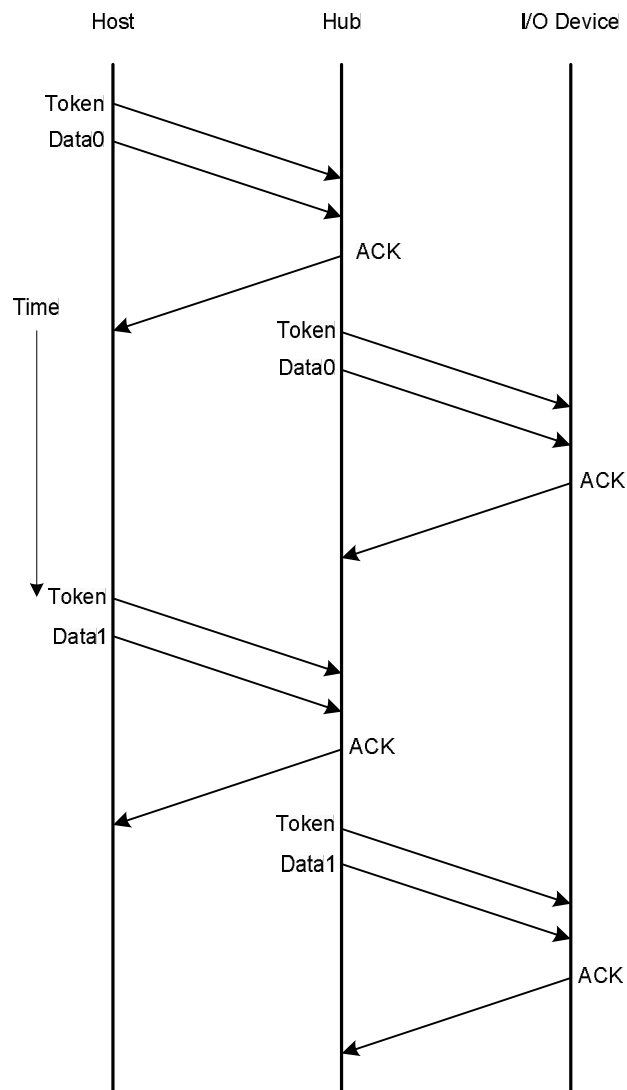


(c) Data packet

Gambar 14.6. Format paket USB

Keempat bit PID mengidentifikasi salah satu dari 16 tipe paket yang berbeda. Beberapa paket kontrol seperti ACK (Acknowledge), hanya terdiri dari byte PID. Paket kontrol yang digunakan untuk mengontrol operasi transfer data disebut paket token. Paket tersebut menggunakan format sebagaimana ditunjukkan pada Gambar 14.6b. Paket token mulai dengan field PID, menggunakan salah satu dari dua nilai PID untuk membedakan antara paket IN dan paket OUT, yang masing-masing mengontrol transfer input dan output. Field PID diikuti dengan alamat 7bit suatu perangkat dan alamat endpoint 4-bit dalam perangkat tersebut. Paket tersebut berakhir dengan 5 bit untuk pemeriksaan error, menggunakan metode yang disebut cyclic redundancy check (CRC). Bit CRC dihitung berdasarkan isi field alamat dan endpoint. Dengan melakukan komputasi terbalik, perangkat penerima dapat menentukan apakah paket tersebut telah diterima dengan tepat.

Paket data, yang membawa data input dan output, memiliki format yang ditunjukkan pada Gambar 14.6c. Field packet identifier diikuti hingga 8192 bit data, kemudian 16 bit error-checking. Tiga pola PID yang berbeda digunakan untuk mengidentifikasi paket data, sehingga paket data mungkin bernomor 0, 1, atau 2. Perhatikanlah bahwa paket data tidak membawa alamat perangkat atau bilangan endpoint. Informasi ini disertakan dalam paket token IN dan OUT yang menginisiasi transfer tersebut.



Gambar 14.7. Transfer Output

Perhatikanlah suatu perangkat output yang dihubungkan ke hub USB, yang kemudian dihubungkan ke komputer host. Suatu contoh operasi output ditunjukkan pada Gambar 14.7. Komputer host mengirim paket token tipe OUT ke hub, diikuti dengan paket data yang berisi data output. Field PID paket data mengidentifikasinya sebagai paket data nomor 0. Hub tersebut memverifikasi bahwa transmisi tersebut telah error free dengan memeriksa bit kontrol error, kemudian mengirim paket pemberitahuan (ACK) kembali ke host. Hub meneruskan token dan paket data ke bagian bawah. Seluruh perangkat I/O menerima rangkaian paket ini, tetapi hanya perangkat yang mengenali alamatnya dalam paket token menerima data dalam paket selanjutnya.

jutnya. Setelah memverifikasi bahwa transmisi tersebut error free, maka perangkat tersebut mengirim paket ACK ke hub.

Paket data yang berurutan pada pipe full-speed atau low-speed membawa bilangan 0 dan 1, secara bergantian. Hal ini menyederhanakan pemulihan dari error transmisi. Jika paket token, data, atau pemberitahuan hilang akibat error transmisi, maka pengirim mengirim ulang seluruh rangkaian tersebut. Dengan memeriksa nomor paket data dalam field PID, maka penerima dapat mendeteksi dan membuang paket duplikat. Paket data high-speed secara berurutan bernomor 0, 1, 2, 0, dan seterusnya.

Operasi input mengikuti prosedur serupa. Host mengirim paket token tipe IN yang berisi alamat perangkat. Hasilnya, paket ini adalah poll yang meminta perangkat untuk mengirim data input apapun yang dimilikinya. Perangkat merespon dengan mengirim paket data yang diikuti dengan ACK. Jika tidak ada data yang siap, maka perangkat merespon dengan mengirim negative acknowledgment (NAK).

Pada pembahasan sebelumnya, kita menekankan bahwa bus yang memiliki campuran link full/low-speed dan high-speed menggunakan mode operasi split-traffic agar tidak menunda pesan pada link high-speed. Pada kasus tersebut, suatu paket IN dan OUT yang dimaksudkan untuk perangkat full- atau low-speed didahului dengan paket kontrol khusus yang memulai mode split-traffic.

Pembahasan ini setnoga memberi ide pada petnbaca tentang protokol transfer data yang digunakan dalam USB. Terdapat banyak cara yang berbeda untuk melangsungkan transaksi tersebut dan banyak protokol yang mengatur kelakuan perangkat yang terlibat.

14.3.7. ISOCHRONOUS TRAFFIC PADA USB

Salah satu tujuan utama USB adalah mendukung transfer data isochronous, seperti sampled voice, dalam cara sederhana. Perangkat yang menghasilkan atau menerima data isochronous memerlukan referensi waktu untuk mengontrol proses sampling. Untuk menyediakan referensi ini, transmisi melalui USB dibagi dalam frame dengan panjang yang setara. Suatu frame memiliki panjang 1 ms untuk data

Sebagai contoh, paket data untuk perangkat 3 dapat berisi 8 byte data. Satu paket semacam itu dikirim dalam tiap frame, dengan saluran isochronous 64-kilobit/det. Saluran semacam itu dapat digunakan untuk koneksi suara. Transmisi 8 byte data memerlukan paket token 3-byte diikuti dengan paket data 11-byte (termasuk field PID dan CRC), mencapai total 132 bit. Minimal tiga byte lagi diperlukan untuk sinkronisasi clock dan untuk menandai akhir rangkaian paket. Pada kecepatan 12 megabit/det, transmisi memerlukan 13 μ s. Jelas sekali tidak ada ruang dalam frame untuk men-support perangkat tersebut. Setelah melayani seluruh perangkat isochronous pada bus tersebut, berapapun waktu yang tersisa dalam frame digunakan untuk melayani perangkat asynchronous dan mempertukarkan informasi kontrol dan status.

Data isochronous hanya diijinkan pada link full-speed dan high-speed. Untuk link highspeed, paket SOF diulang delapan kali pada interval yang setara dalam frame 1-ms untuk menghasilkan delapan microframe masing-masing 125 μ s.

14.3.8. KARAKTERISTIK ELEKTRIK

Kabel yang digunakan untuk koneksi USB terdiri dari empat kawat. Dua digunakan untuk membawa daya, +SV dan Ground. Sehingga, hub atau perangkat I/O dapat langsung mendapat daya dari bus, atau memiliki koneksi daya eksternal sendiri. Dua kawat yang lain digunakan untuk membawa data. Skema signaling yang berbeda digunakan untuk kecepatan transmisi yang berbeda. Pada kecepatan rendah, 1 dan 0 ditransmisikan dengan mengirim keadaan tegangan tinggi (SV) pada satu bagian atau bagian lain dari dua kawat sinyal tersebut. Untuk link high-speed, digunakan transmisi diferensial.