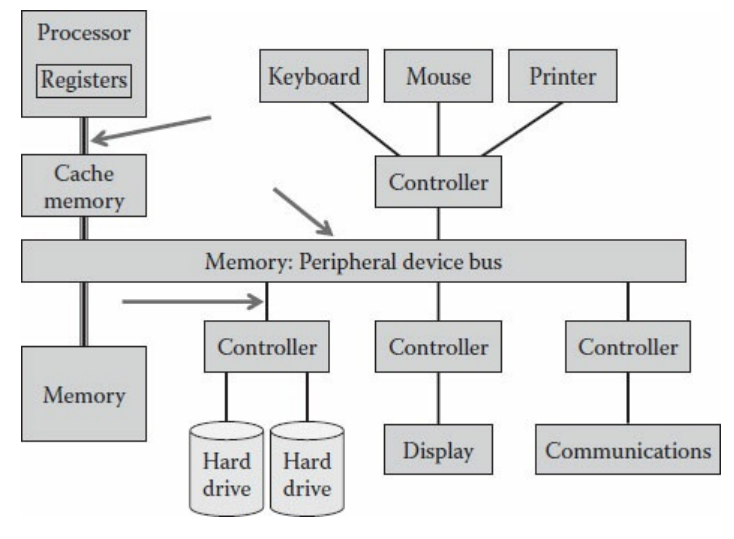
|  |  |  |
| --- | --- | --- |
| **Mata Kuliah** | **:** | **Arsitektur dan Organisasi Komputer** |
| **Bobot Sks** | **:** | **3 sks** |
| **Dosen Pengembang** | **:** | **Catur Nugroho, S.Kom., M.Kom** |
| **Tutor** | **:** |  |
| **Capaian Pembelajaran Mata Kuliah** | **:** | **Mahasiswa mampu menggunakan dan menerapkan konsep & definisi Sistem Bus.** |
| **Kompetentsi Akhir Di Setiap Tahap (Sub-Cpmk)** | **:** | **Mahasiswa mampu menggunakan dan menerapkan konsep & definisi Sistem Bus.** |
| **Minggu Perkuliahan Online Ke-** | **:** | **Sesi 3** |

1. **Definisi Sistem Bus.**

Pada sesi-3 memfokuskan berbagai aspek Bus, merupakan infrastruktur untuk transfer data dalam sistem. Dengan gambar arsitektur menghubungkan bus sistem (Gambar 3.1).



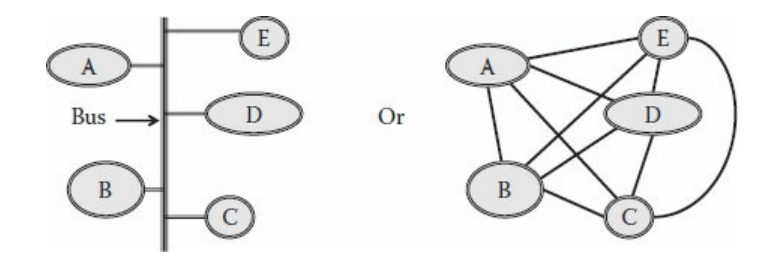
Gambar 3.1 Bus bagian dari arsitektur komputer

Sebuah Bus disebut saluran yanga artinya mekanisme untuk menghubungkan berbagai unit fungsional dalam sistem. beberapa unit yang menghubungkan dan mentransfer data, misalnya prosesor, yang membutuhkan data dari memori, baik untuk instruksi maupun operan. Contoh adalah berbagai jenis memori yang dihubungkan untuk mentransfer data dengan lancar dan baik.

Konsep modularitas, yang ditekankan oleh ***arsitektur von Neumann***, membutuhkan mekanisme untuk transfer data, dan diimplementasikan menggunakan berbagai bus.

**Bus** adalah kumpulan kabel listrik paralel yang digunakan untuk mentransfer data. Setiap kawat (*wire*) tersebut mentransfer satu bit. Kumpulan kabel mampu mentransfer byte atau kata sekaligus sebagai fungsi lebarnya. Bus menggunakan beberapa protokol atau aturan yang mendefinisikan perilakunya, seperti: kepemilikan, siapa yang dapat mengirim data dan dalam urutan apa, prioritas, dan sebagainya.

Gambar 3.2 menggambarkan dua jenis bus yang menghubungkan lima unit fungsional (perangkat) yang berbeda direpresentasikan sebagai elips.



3. 2 Jenis bus.

Diagram di sebelah kanan adalah mekanisme yang menyediakan jaringan koneksi, dan setiap perangkat terhubung langsung ke semua perangkat lain. Jenis ini memberikan yang sangat baik kecepatan transfer karena satu transfer antara dua perangkat tidak mempengaruhi transfer data antara perangkat lain.

Di sisi kiri, ada satu bus yang menghubungkan semua perangkat. Dalam hal ini, semua perangkat yang terhubung harus berbagi protokol yang sama. Setiap perangkat yang perlu mentransfer data harus memperoleh akses ke bus dan kemudian bertindak sesuai terhadap aturan yang mengatur, setiap perangkat harus meminta izin sebelum mulai menggunakan bus. Contoh realistis untuk bekerja dengan bus adalah **sistem kereta api**. Di asumsikan hanya ada satu jalur kereta api antara dua titik dan kereta api yang berjalan di kedua arah. Setiap kereta api yang meninggalkan stasiun yang hanya memiliki satu jalur kereta api harus memastikan, sebelum memulai perjalanannya, bahwa tidak ada kereta api lain yang menggunakan jalur tersebut. Hal yang sama berlaku untuk setiap perangkat yang perlu menggunakan bus.

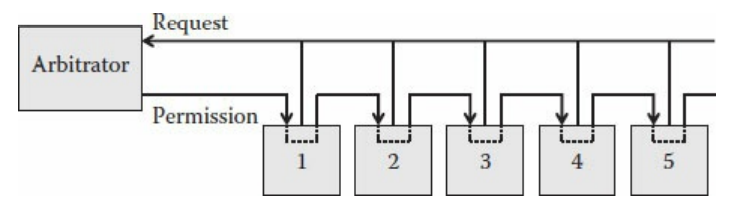
Pertama, ia harus meminta izin untuk menggunakan bus dan hanya setelah izin diberikan perangkat dapat memulai transfer data. Transfer sendiri digunakan untuk :

* Mengirim perintah *(read, write, etc.)*
* Mengirim alamat perangkat di ujung yang lain (seperti memanggil nomor saat menggunakan sistem telepon)
* Mentransfer data itu sendiri (setelah komunikasi terjalin)

Sebuah bus dapat dikelola dari lokasi pusat (*Central Arbitrase*), atau alternatifnya dapat menggunakan *arbitrase* terdistribusi.

*Central Arbitrase* (Gambar 3.3) dicirikan oleh satu unit pusat (*arbitrator*) yang mengelola transfer bus. Setiap perangkat yang ingin menggunakan bus terlebih dahulu akan meminta izin kepada *arbitrator*, dan hanya setelah izin diberikan, perangkat dapat menggunakan bus untuk keperluannya. Karena berbagai jenis transfer yang mungkin dilakukan di bus, secara logis dibagi menjadi tiga yang berbeda saluran:

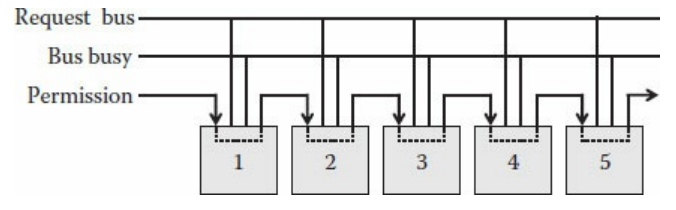
* Satu digunakan kontrol, perangkat menggunakannya untuk mengirim perintah.
* Yang kedua digunakan untuk alamat.
* Yang ketiga digunakan untuk transfer data.



3.3 *Central arbitration*

Semua perangkat terhubung secara serial dan mendengarkan saluran alamat. Saat perangkat alamat tertentu disiarkan, perangkat menjawab. Semua perangkat lain terus mendengarkan, tetapi jika ya bukan nomor mereka, mereka mengabaikannya. Dalam arti tertentu, ini seperti telepon seluler, yang mendengarkan jaringan dan merespons hanya jika nomor mereka dipanggil.

*Distributed arbitration* (Gambar 3.4) dicirikan oleh fakta bahwa tidak ada pusat manajemen untuk kegiatan bus. Alih-alih unit pusat, manajemen dilakukan secara kolaboratif oleh semua perangkat dengan mengikuti protokol yang ditentukan. Dalam *Distributed arbitration*, perangkat masih harus meminta izin untuk menggunakan bus; namun, proses konfirmasinya agak berbeda. Selain tiga bagian bus (saluran), ada satu tambahan yang digunakan untuk memberi isyarat jika bus digunakan atau gratis. Setiap perangkat memeriksa sinyal sibuk dan menunggu sampai bus menjadi bebas.



3.4 Distributed arbitration.

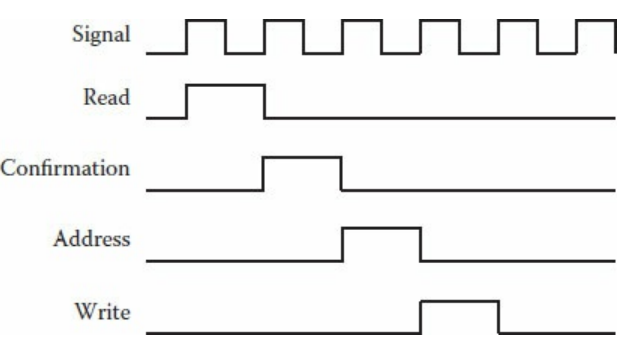
Baru kemudian ia mengirimkan permintaannya. Perangkat lain di bus menerima permintaan, dan mereka menjawab berdasarkan mekanisme prioritas yang telah ditentukan. Ketika transfer selesai, perangkat bertanggung jawab untuk mengubah sinyal bus kembali menjadi bebas sehingga perangkat lain dapat menggunakannya. Ini penting untuk perhatikan bahwa ini adalah definisi yang sangat dasar dari algoritma prioritas, dan pada kenyataannya ada tambahan algoritme untuk memastikan komunikasi yang lancar, termasuk manajemen prioritas dan mencegahan kemacetan.

Saat perangkat mentransfer data, perangkat menggunakan salah satu dari dua metode:

* Transaksi penuh (*Full transaction*): Ini berarti perangkat meminta bus, dan setelah diberikan, perangkat memegangnya selama komunikasi berlangsung. Ini berarti bahwa selama ini, beberapa blok dapat dikirim bolak-balik. Perangkat yang meminta mengirimkan alamat dan kemudian yang lain balasan perangkat. Pada tahap selanjutnya, perangkat yang meminta mengirimkan permintaan dan perangkat lainnya balasan, dan ini berlangsung sampai semua data yang diperlukan telah diterima. Hanya ketika keseluruhan komunikasi selesai akan perangkat yang meminta melepaskan bus. Modus operasi ini adalah mirip dengan komunikasi jalur, di mana telepon pemanggil menahan saluran untuk seluruh panggilan durasi, bahkan jika tidak ada yang benar-benar ditransfer di telepon. Jenis komunikasi ini adalah hanya berlaku untuk perangkat yang sangat cepat dan komunikasi singkat, karena selama transaksi ini, bus sedang sibuk dan semua perangkat lain yang membutuhkannya harus menunggu.
* Transaksi terpisah (*Split transaction*): Ini berarti perangkat meminta bus dan, ketika izin diberikan, perangkat akan mengirimkan perintah, alamat, atau data dan segera setelah itu akan melepaskan bus. Transaksi split adalah jenis yang lebih disukai ketika ada penundaan antara operasi. Untuk contoh, asumsikan memori cache membutuhkan blok data dari memori utama. Itu pengontrol memori cache akan meminta izin untuk menggunakan bus. Ketika izin diberikan, pengontrol cache akan mengirim perintah ke memori utama, meminta blok. Mengikuti transfer perintah ini, pengontrol memori cache akan melepaskan bus. Hal ini dilakukan sejak pengontrol memori utama membutuhkan waktu untuk memproses permintaan. Ada kemungkinan bahwa ia memiliki beberapa permintaan sudah menunggu, yang harus diproses sebelum permintaan saat ini diproses. Bahkan jika permintaan saat ini adalah yang pertama, itu akan memakan waktu untuk memori utama controller untuk mengakses memori dan mengambil blok yang diperlukan. Hanya ketika data tersedia akan pengontrol memori utama meminta izin untuk menggunakan bus, dan ketika izin diberikan, itu akan mentransfer blok ke pengontrol memori cache. Mode operasi ini mirip dengan yang digunakan oleh jaringan seluler modern, di mana saluran digunakan bersama oleh banyak telepon. Setiap pasangan (dengan asumsi tidak ada panggilan konferensi) telepon membuat saluran virtual menggunakan bus. Bus ini mendukung beberapa saluran virtual seperti itu yang bekerja secara bersamaan dalam mode pembagian waktu. Ide utama dari metode *split-transaction* adalah jika tidak ada yang ditransfer, virtual saluran akan melepaskan bus sehingga dapat digunakan oleh orang lain

Mentransfer data melalui bus dapat dilakukan dengan dua cara berdasarkan jenis bus:

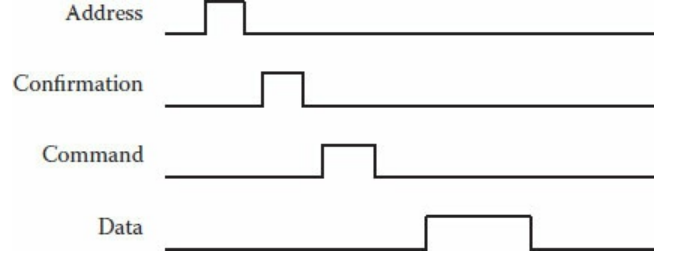
* Salah satu jenis bus yaitu *synchronous bus* yang disinkronkan dengan jam. Saluran kontrol termasuk sinyal konstan yang digunakan oleh semua perangkat yang terhubung ke bus. Setiap operasi pada bus seperti pengiriman data disinkronkan ke awal sinyal. Acara (mengirim perintah, alamat, atau data) akan berlangsung selama satu atau dua siklus bus (Gambar 3.5).



Gambar 3.5 *Synchronous bus.*

Baris pertama (atas) menggambarkan sinyal sinkronisasi, dan dapat dilihat berikut ini: baris yang semua operasi lain (*read, confirmation, write, etc*.) disinkronkan dan dimulai dari awal dari sebuah siklus.

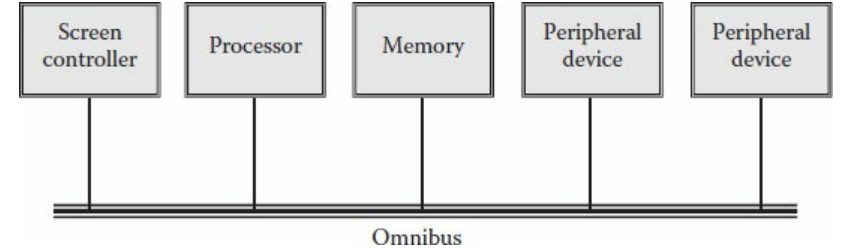
Jenis lainnya adalah *asynchronous bus*, yang tidak disinkronkan, dan setiap operasi mengikuti sebelumnya. Perangkat yang mengirim data menentukan tarif. Di satu sisi, pendekatan ini lebih rumit, tetapi di sisi lain, memungkinkan koneksi berbagai perangkat yang bekerja pada kecepatan berbeda (Gambar 3.6).



Gambar 3.6 *Asynchronous bus.*

1. **Principle Bus**

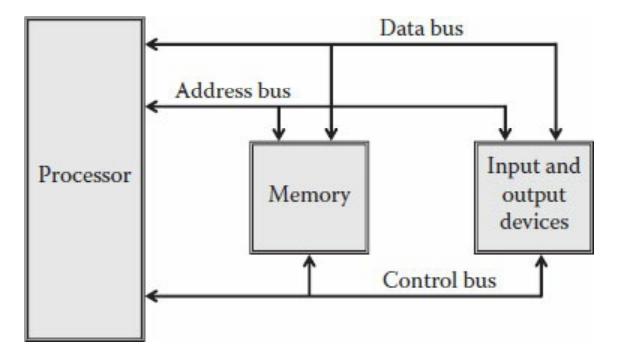
Bus pertama kali muncul pada tahun 1964 di komputer yang disebut *programmed data processor* (PDP),\* yang dirancang dan dibuat oleh *Digital Equipment Corporation* (DEC). Industri komputasi saat itu ditandai dengan komputer besar yang sangat mahal dan menggunakan sistem pendingin khusus. PDP, yang disebut ***minicomputer***, lebih murah dibandingkan dengan ***mainframe*** telah menerapkan beberapa ide revolusioner; misalnya, bisa bekerja di ruangan ber-AC biasa tanpa fasilitas pendingin khusus. Ide lain yang diterapkan adalah bus umum (disebut ***omnibus***). Sebelum pengenalan bus, sistem komputer menggunakan saluran yang membentuk matriks koneksi di mana setiap perangkat terhubung ke semua perangkat lain menggunakan kabel khusus (sangat mirip diagram kanan pada Gambar 3.2). Dengan menggunakan bus umum yang dapat digunakan bersama oleh semua perangkat, biaya dapat diturunkan; namun, ini menyebabkan beberapa masalah kemacetan lainnya. Gambar 3.7 menggambarkan arsitektur PDP dengan bus umum



Gambar 3.7 PDP 8 architecture.

bus secara logis dibagi menjadi tiga saluran dengan fungsi yang berbeda (Gambar 3.8):

* + Saluran kontrol, digunakan untuk mengirim perintah dan permintaan
  + Saluran alamat, digunakan untuk mengirim alamat perangkat
  + Saluran data, digunakan untuk mengirim data

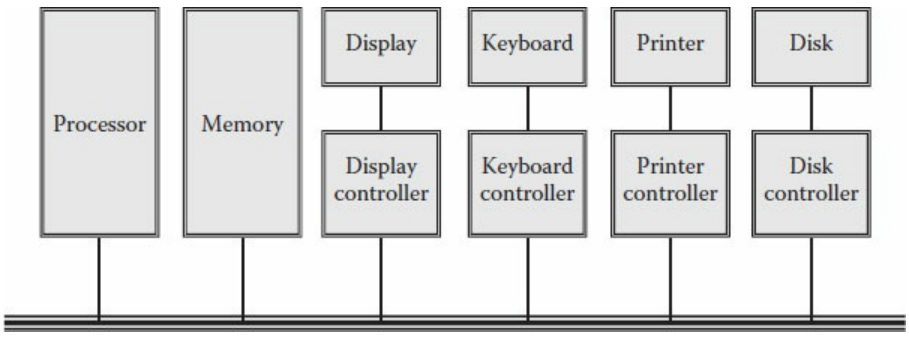


Gambar 3.8Channel types.

Perlu dicatat bahwa secara fisik, bus terdiri dari sekelompok kabel di mana pembagian logis ini dipertahankan.

***Prinsip bus*** adalah kontributor yang signifikan dan penting untuk pengembangan industri komputasi. Sebelum munculnya *personal computer* (PC), dunia komputasi didominasi oleh beberapa vendor yang memproduksi dan menjual sistem *turnkey*. Dengan demikian, bus dan saluran yang digunakan adalah milik, sehingga perusahaan lain tidak dapat menyediakan perangkat atau komponen apa pun untuk terhubung ke sistem berpemilik ini. Berlawanan dengan pendekatan ini, PC dikembangkan menggunakan bus standar yang menyediakan komponen dan perangkat tanpa akhir yang dapat dihubungkan dengan mulus ke sistem. Lingkungan yang dulu tertutup saat ini sangat terbuka, memberikan banyak vendor, bahkan yang kecil, akses ke pasar—dengan asumsi solusi mereka lebih baik. Lingkungan terbuka ini berhasil menurunkan harga karena persaingan yang ketat, seperti yang kita semua tahu. PC yang diproduksi oleh vendor mana pun dapat terhubung ke memori yang diproduksi oleh vendor lain serta disk yang diproduksi oleh vendor lain dan seterusnya.

Standar Bus diimplementasikan di PC pada mulanya dimaksudkan untuk menurunkan Biaya PC, seperti yang terjadi pada PDP 20 tahun sebelumnya. Faktanya, PC pertama diimplementasikan arsitektur yang menyerupai arsitektur PDP (Gambar 3.9).



Gambar 3.9 Arsitektur PC pertama

seperti yang terjadi pada PDP 20 tahun sebelumnya. Faktanya, PC pertama mengimplementasikan arsitektur yang menyerupai arsitektur PDP (Gambar 3.9).

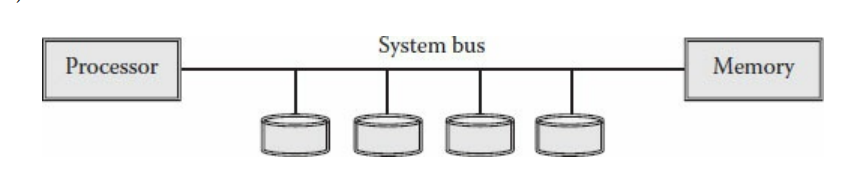
Kontroler yang merupakan bagian dari arsitektur utama digunakan untuk menjembatani antara protokol bus dan perangkat (*keyboard, display, disk, dll*). Karena setiap perangkat bekerja secara berbeda, perlunya menerjemahkan dan mengintegrasikan perangkat agar dapat berfungsi di lingkungan bus.

Biasanya, produsen perangkat mengembangkan pengontrol, kecuali jika perangkat tersebut adalah perangkat standar, yang mungkin sudah memiliki pengontrol yang tersedia. Selain keuntungan ekonomi yang terkait dengan bus, juga memungkinkan kemudahan penggantian perangkat serta transfer perangkat dari satu sistem ke sistem lain (dengan asumsi kedua sistem menggunakan bus yang sama). Sayangnya, beberapa kelebihan berubah menjadi kekurangan karena perangkat kerasnya canggih. Karena semakin banyak perangkat yang terhubung ke bus, terutama perangkat dengan kecepatan dan kecepatan transfer yang bervariasi, bus menjadi hambatan. Misalnya, ada kasus dimana prosesor menunggu data ditransfer dari memori hanya karena bus sibuk mentransfer karakter yang diketik di keyboard. Solusi yang diadopsi, seperti kasus memori membuat hierarki bus berdasarkan fungsionalitas yang diperlukan :

* Bus prosesor-memori, yang harus menyediakan kecepatan transfer maksimum yang mungkin. Dalam banyak kasus, ini adalah bus berpemilik yang diimplementasikan oleh pabrikan motherboard\*. Seperti tentang memori dan memori cache, bus yang menghubungkan prosesor ke memori dapat menjadi hambatan parah yang berdampak negatif pada kinerja. Dalam kecepatan modern, panjang elektron bergerak mulai menjadi faktor waktu yang relevan.
* ***Bus input dan output*** (I/O) biasanya merupakan bus standar. Ini digunakan untuk meningkatkan potensi perangkat I/O yang dapat dihubungkan ke sistem.
* Bus sistem, yang dapat berupa bus standar atau bus berpemilik yang terletak di motherboard. Dalam beberapa implementasi, tidak ada bus sistem, dan kemudian bus I/O dan bus memori prosesor dihubungkan.

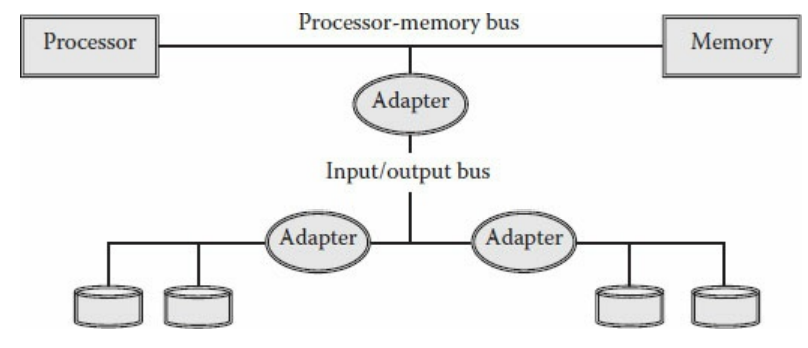
1. **BUS EVOLUTION**

Implementasi pertama bus (PDP 1960-an dan di PC pada awal 1980-an) menggunakan *single bus*, dimana semua perangkat sistem terhubung ke bus ini (Gambar 3.10). Keuntungan menggunakan *single bus* sudah jelas (kesederhanaan, yang menyebabkan biaya lebih rendah). Bus yang bertanggung jawab atas semua transfer data dalam sistem (*prosesor, memori, disk, keyboard,* dll.). Banyaknya perangkat yang terhubung ke bus tidak hanya memperlambatnya tetapi juga sangat memengaruhi kecepatan prosesor. Sebuah *single bus* harus lebih panjang, yang mempengaruhi waktunya. Bahkan dalam kasus di mana bus bekerja secara efisien, volume data yang ditransfer oleh semua perangkat yang berbeda mendekati kapasitas maksimumnya. Selain itu, kecepatan berbagai perangkat mewakili besar spektrum nilai. Misalnya, ***keyboard*** yang bergantung pada klik manusia dapat menghasilkan kurang dari 10 penekanan tombol per detik. Setiap penekanan tombol diterjemahkan ke dalam karakter delapan (8-bit) atau bandwidth maksimum 80 bit perdetik. Di ujung yang lebih tinggi, prosesor yang bekerja pada kecepatan clock 3 GHz membutuhkan puluhan miliar bit per detik.



Gambar 3.10 ***single bus.***

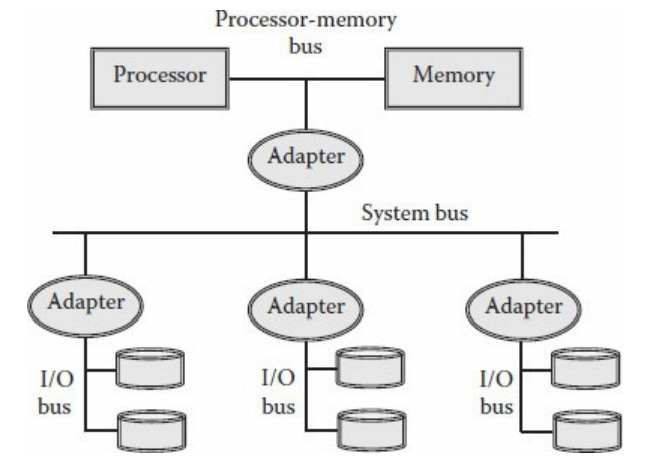
Dalam sistem *two buses*, akan ada bus memori prosesor cepat dan bus I/O, keduanya akan menghubungkan adaptor khusus untuk menjembatani keduanya (Gambar 3.11).



Gambar 7.11 ***system two buses***

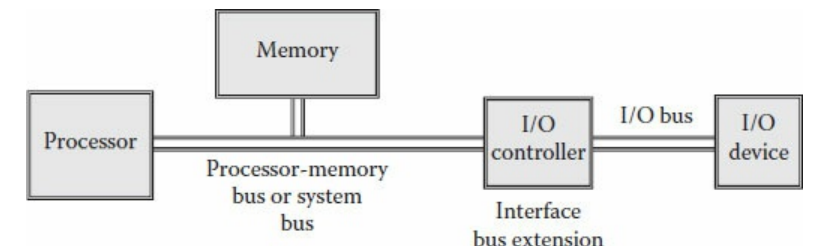
Sebuah sistem yang memiliki lebih dari satu bus akan selalu menggunakan beberapa hierarki di antara bus-bus tersebut. Bus cepat menghubungkan prosesor dan memori. Bus lain akan terhubung ke bus memori prosesor.

***Adapter*** adalah semacam pengontrol yang menghubungkan dan menyinkronkan antara dua bus dan protokolnya. Adapter menyertakan beberapa ***buffer internal*** yang digunakan untuk mengumpulkan data sementara pada bus yang lebih lambat. Hanya setelah ***buffer*** penuh, adaptor akan mengirim data sebagai satu blok melalui bus yang lebih cepat. Dengan cara ini, meski data dikumpulkan dari perangkat yang lambat, bus prosesor-memori tidak terganggu sampai seluruh blok siap untuk ditransfer. Mekanisme ini meminimalkan dampak perangkat yang lebih lambat pada bus memori prosesor, beberapa sistem menggunakan lebih banyak bus untuk mengurangi kemungkinan dampak negatif (Gambar 3.12).



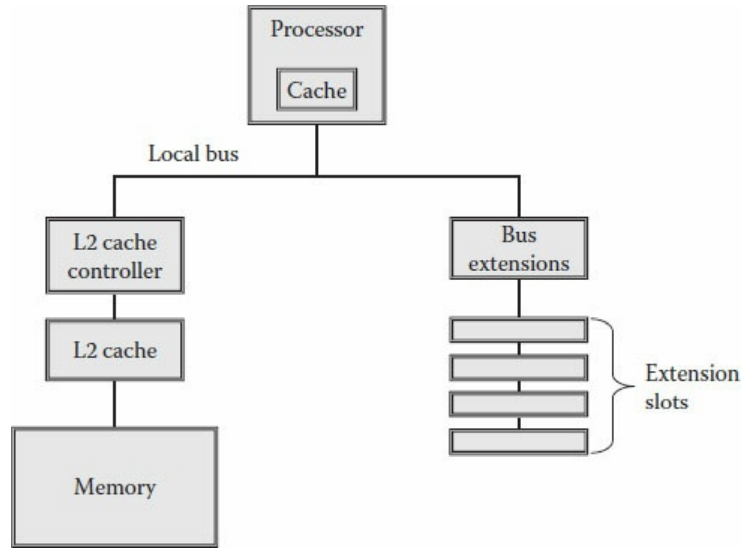
Gambar 3.12 *system three buses*

Pada contoh yang digambarkan pada Gambar 3.13, pengontrol I/O bertindak sebagai konverter protokol. Di satu ujung, ada bus yang menghubungkan adaptor, dan di ujung lainnya adalah perangkat I/O spesifik. Pengontrol juga menyediakan kemampuan menghubungkan perangkat ke berbagai bus standar. Dalam banyakan kasus, pengontrol dikembangkan dan dijual oleh produsen dalam upaya agar perangkat bekerja pada berbagai bus. Dalam beberapa kasus, untuk contoh



***3.13*** *extension bus.*

Sebagian besar PC, beberapa bus terletak di motherboard, dan menghubungkan pengontrol dilakukan dengan memasukkannya ke dalam slot khusus seperti yang ditunjukkan pada Gambar 3.14



Gambar 3.14 *Extension slots.*

Ada berbagai macam bus standar dan jumlahnya terus bertambah. Beberapa dari bus dan protokol standar ini adalah

† *Industry Standard Architecture* (ISA)\*

† *Extended* ISA (EISA)

† *Micro Channel Architecture* (MCA)

† *Personal Computer Interconnect* (PCI)

† *Single Inline Memory Module* (SIMM)

† *Dual Inline Memory Module* (DIMM)

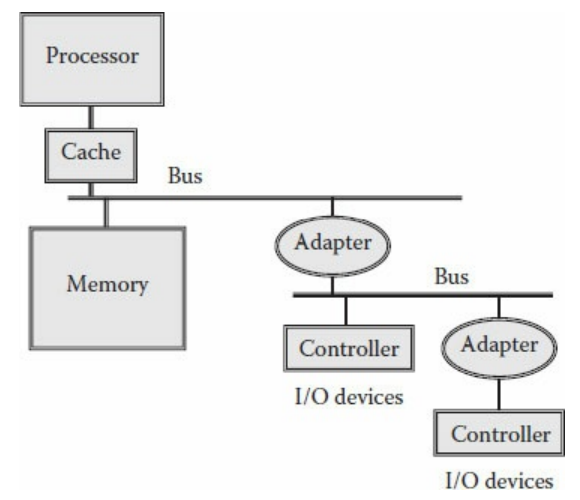
† *Accelerated Graphics Port* (AGP)

† *Personal Computer Memory Card International Association* (PCMCIA)

† *Universal Serial Bus* (USB)

Standar-standar ini hanyalah beberapa contoh dari solusi yang dirancang dan fakta bahwa solusi ini terus berkembang seiring berkembangnya teknologi. Selain fleksibilitas tinggi dalam menghubungkan banyak perangkat periferal I/O, berbagai standar mendukung ***hierarki bus.*** Ini menunjukan bahwa koneksi memfasilitasi cara yang efisien untuk gunakan *fast buses* pada kecepatan maksimumnya dengan sedikit gangguan yang disebabkan bus menjadi lebih lambat.

Ide awalnya adalah untuk mendukung data yang diperlukan untuk prosesor tanpa penundaan sambil memberikan infrastruktur umum untuk mentransfer data antara semua komponen sistem. ***Keyboard,*** misalnya, hampir tidak mengalami perubahan selama bertahun-tahun. Dengan menggunakan adaptor yang tepat, tidak perlu mendesain keyboard baru yang mampu bekerja dengan bus baru.

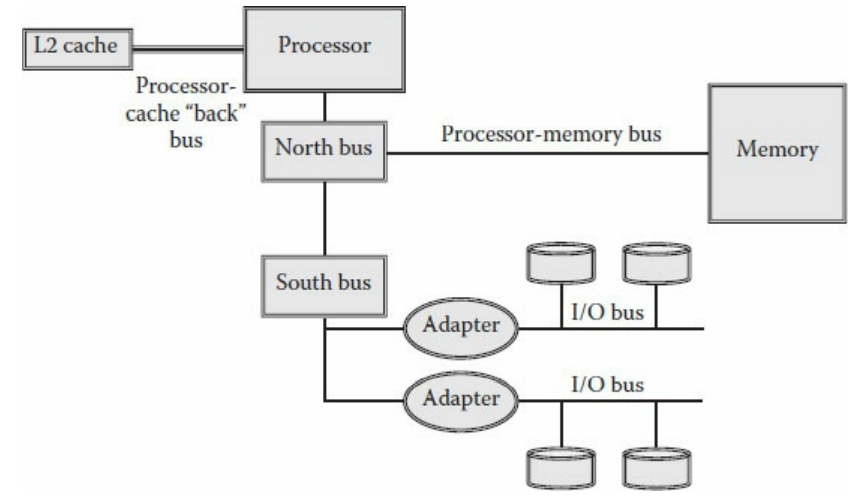


Gambar 3.15hierarki bus

Gambar 3.15menggambarkan sistem yang mendukung hierarki bus. Di sudut kiri atas adalah ***prosesor*** dengan ***cache memori***. Ada koneksi langsung antara keduanya karena tidak ada perangkat tambahan yang terhubung. Biasanya, akan menjadi koneksi yang sangat cepat. ***Cache memori*** terhubung ke memori menggunakan bus cepat. Satu-satunya perangkat yang terhubung ke bus ini adalah ***cache memori***, ***memori utama***, dan ***adapter*** yang menghubungkan bus tingkat rendah. Setiap kali data yang dibutuhkan ***prosesor*** tidak ada dalam ***cache memori***, bus akan digunakan untuk mentransfer data dari memori ke ***cache memori*** dan ***prosesor***. Dari waktu ke waktu, adapter akan menggunakan bus untuk mentransfer data, tetapi itu jarang terjadi, dan ketika itu terjadi, ***adapter*** bekerja pada kecepatan bus. Bus yang lebih rendah adalah bus I/O, yang menghubungkan berbagai pengontrol. Kontroler ini menghubungkan perangkat periferal, atau dapat digunakan untuk menjembatani ke bus yang lebih rendah dan lebih lambat, misalnya, seperti pada

pengontrol komunikasi lambat yang terhubung ke bus dan yang, terhubung ke berbagai perangkat bus komunikasi tambahan (ini bukan bagian dari gambar).

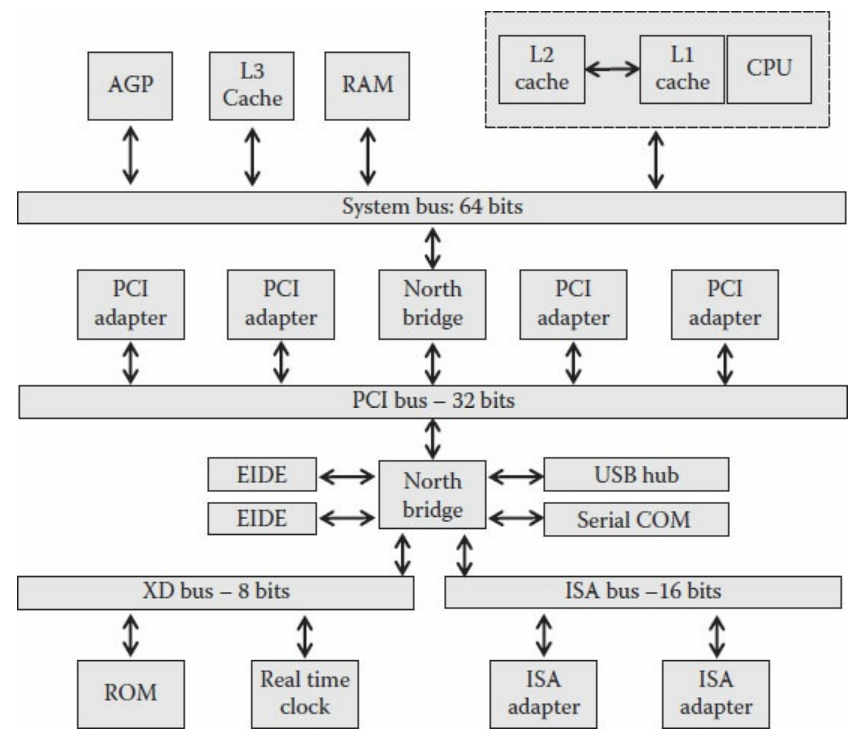
Kebutuhan terpenting yang mengarah pada pengembangan hierarki bus adalah meminimalkan sebanyak mungkin penundaan dalam mentransfer data ke prosesor. Masalah penting dalam merancang arsitektur dan hierarki memori. Hirarki bus harus, di satu sisi, menyediakan akses terbuka, di mana data dari perangkat yang lebih lambat bisa sampai ke prosesor; dan di sisi lain, perangkat yang lebih lambat tidak boleh memperlambat prosesor. Gambar 3.16 adalah contoh desain yang jauh dari optimal.



Gambar 3.16*Bus conflicts*

Gambar 3.16 menjelaskan perancang menyediakan bus yang sangat cepat antara prosesor dan *cache memory*, tetapi prosesor, di sisi lain, terhubung ke bis utara. Bus prosesor-memori juga terhubung ke bus utara. Dengan bus utara sebenarnya adalah bus sistem, dan sebagian besar transfer data dalam sistem melewatinya. Selanjutnya, ***cache memori*** harus terhubung langsung ke memori untuk menyediakan komunikasi cepat yang diperlukan, danmemori tersebut hilang.

Gambar 3.17 mengambarkan contoh arsitektur bus di sistem berbasis ***Pentium***. Di sudut kanan atas, prosesor dengan ***cache on-chip* Level 1**. Saluran komunikasi cepat khusus menghubungkan dua cache (Level 1 dan Level 2). Komponen (terdiri dari prosesor dan dua cache) terhubung ke bus **sistem 64-bit** yang cepat. Bus digunakan untuk menghubungkan memori utama, adaptor grafis dan **cache Level 3**. Adaptor grafis untuk mendukung tampilan grafis yang sangat cepat, misalnya untuk para gamer, monitor biasa akan terhubung menggunakan USB yang lebih lambat.

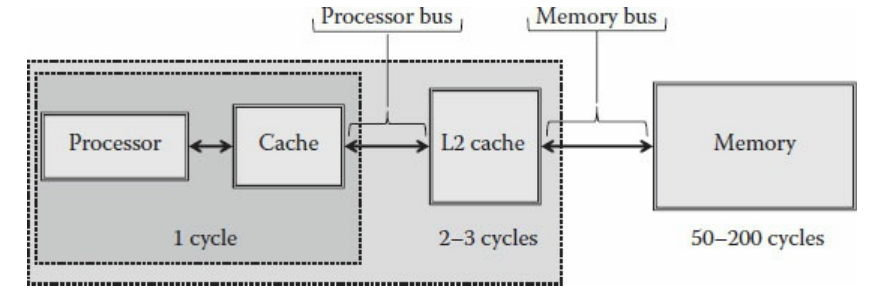


Gambar 3.17 *Pentium architecture*

Kecepatan memori dipengaruhi oleh dua aspek utama: ***kecepatan bus*** dan **lebarnya**. Kecepatan menentukan berapa banyak unit data yang dapat ditransfer setiap siklus, dan lebar menentukan unit data (***byte, word,, dll***.). Misal, jika kecepatan bus adalah 800 MHz, berarti bus tersebut mampu mentransfer 800 juta blok perdetik. Blok memiliki lebar 64 bit. Bus yang lebih lebar lebih mahal (membutuhkan lebih banyak kabel serta konektor yang besar menempati lebih banyak ruang pada motherboard).

Bus sistem yang cepat terhubung ke bus PCI 32-bit. ***North bridge*** digunakan untuk menghubungkan dua bus, menyesuaikan kecepatan dan ukuran blok, dan menerjemahkan protokol. Di sisi bawahnya, bus PCI terhubung ke jembatan selatan, yang menghubungkan genap tambahan bus yang lebih lambat. Salah satu fitur kesuksesan lini PC komputer adalah kompatibilitas ke belakang. Komputer modern masih mendukung beberapa yang lama dan bahkan periferal usang, seperti ***keyboard, floppy disk***, dan sebagainya. ***South bridge,*** menghubungkan satu sisi bus PCI dan di sisi lain terhubung ke bus modern seperti USB yang menyediakan kompatibilitas dengan menghubungkan ke bus 16-bit dan bahkan 8-bit yang memungkinkan penggunaan kembali beberapa periferal yang, meskipun bekerja relatif lambat, tidak perlu diganti, seperti jam waktu nyata yang menyinkronkan aktivitas system.

Gambar 3.18 menguraikan aspek waktu dari bus yang dekat prosesor. Dilihat bahwa mengakses ***cache internal*** (Level 1) membutuhkan satu siklus, mirip dengan mengakses register. Ketika data cache tingkat kedua, diperlukan dua atau tiga siklus tergantung implementasi dan kedekatannya. Dalam kasus di mana data tidak ditemukan di Level 1 dan Level 2, itu harus dibawa dari memori. Dalam hal ini, "biaya" bisa menjadi 50–200 siklus berdasarkan implementasi tertentu, konflik bus, dan sebagainya. Angka tersebut membahas pentingnya memori cache bagi kinerja sistem.



Gambar 3.18 *Faster bus access time.*

1. **HARD DRIVE BUSES**

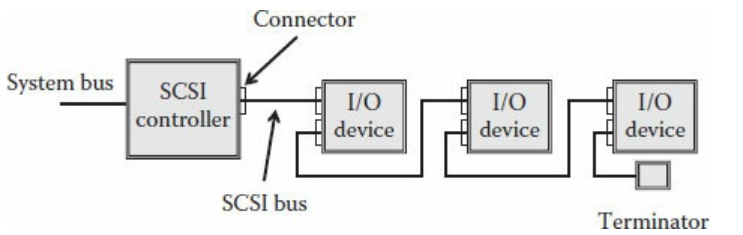
Disk, terutama ***hard drive*** serta ***optical disk***, ***solid-state disk***, dan ***disk-on-key***, termasuk di antara perangkat paling penting yang dikembangkan secara paralel dengan prosesor dan memori. Banyak produsen berkonsentrasi pada disk dan menyediakan spektrum besar perangkat untuk diintegrasikan ke dalam banyak sistem (*computers, handheld appliances, mobile phones, cameras*, dll.). Untuk menyediakan konektivitas untuk menghubungkan disk secara eksplisit. Sejak disk memiliki atribut, ada kebutuhan untuk bus khusus yang dirancang khusus dari waktu ke waktu dalam proses evolusi yang mengikuti perkembangan teknologi. Karena pesatnya perkembangannya perlu perbaikan yang lebih penting relatif baru dari :

* *AT Attachment* (ATA)\*
* *Integrated Drive Electronics* (IDE)
* *Enhanced IDE* (EIDE)
* *Small Computer Systems Interface* (SCSI)

Beberapa dari standar ini tidak ada lagi, telah diganti, permintaan yang terus meningkat untuk disk yang lebih cepat dan berkapasitas tinggi berdampak pada bus standar, yang harus ditingkatkan untuk mengakomodasi fitur-fitur baru dan kemampuan, di mana produsen "mendorong" produknya dan mendapatkan pasar besar penerimaan. Biasanya, produsen ***hard drive*** tidak tertarik hanya untuk mengembangkan standar baru;

Hal ini dilakukan dalam rangka mempromosikan produk mereka. Pada abad sekarang, industri komputasi lebih menyukai standar terbuka berbagi perangkat produsen.

Salah satu bus yang banyak digunakan untuk sistem performa tinggi dan saat ini sedang digunakan terutama untuk server adalah ***SCSI***. Bus yang dikembangkan beberapa dekade lalu ini telah banyak digunakan oleh IBM untuk mencoba membedakan produknya dan memberikan kinerja yang unggul. (Gambar 3.19).



Gambar 3.19. SCSI bus

Secara umum, bus ini tidak berbeda dengan bus lainnya. memiliki adaptor (pengontrol SCSI) yang mengubah dan menjembatani antara bus (seperti PCMCIA atau USB dan SCSI). Kelebihannya adalah kemampuannya untuk menggabungkan beberapa perangkat di bus. Perangkat terakhir terhubung ke terminator yang mengidentifikasi ujung bus (Gambar 3.19, pojok kanan bawah). Karena throughputnya yang relatif tinggi Bus itu berhasil dibandingkan dengan bus lain. Seperti lainnya perangkat dan bus di pasar PC, standar mengalami serangkaian peningkatan, banyak variasi dikembangkan. Dalam kebanyakan kasus, variasinya adalah untuk mengakomodasi fitur seperti kecepatan transfer yang lebih cepat, meningkatkan jumlah maksimum perangkat di bus.

Bus, antarmuka, dan protokol yang lebih umum digunakan di PC adalah

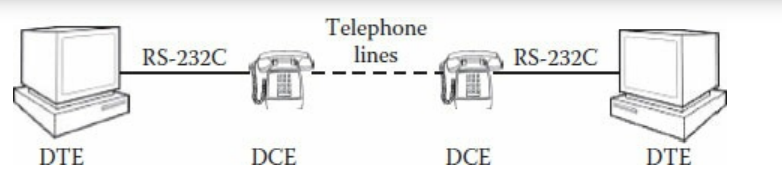
* Parallel (line printer or LPT)\*
* Serial (typically RS232C)
* PS/2 (for keyboards and mice)
* Universal Serial Bus (USB)
* Infrared Data Association (IrDA)\*
* Bluetooth (wireless communication)
* FireWire, or IEEE standard 1394
* WiFi (Wireless Fidelity)

Sebagai contoh, penjelasan yang lebih rinci tentang koneksi dan protokol disediakan untuk serial bus yang sangat sederhana.

1. **Serial Bus**

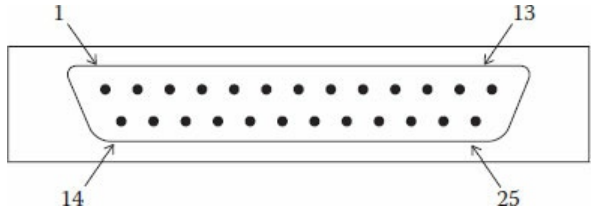
Serial bus diimplementasikan pada model awal PC. Gagasan di balik serial bus didasarkan pada mentransfer data satu per satu. Jika datum 1-byte harus ditransfer, maka pada setiap siklus-bus, satu bit akan dikirim. Mengirim seluruh byte akan membutuhkan delapan siklus bus. Ini adalah cara komunikasi yang sangat lambat, tetapi sederhana dan memberikan penjelasan yang baik tentang prinsip komunikasi PC. Komunikasi PC dilakukan menggunakan ***communication (COM) ports***). Komputer pertama termasuk empat port yang terhubung dengan perangkat periferal.

Protokol yang digunakan untuk berkomunikasi pada port serial disebut RS-232, dikembangkan oleh ***Electronic Asosiasi Industri*** (EIA) untuk mempromosikan menghubungkan dan menggunakan peralatan elektronik. Salah satu contoh awal adalah penggunaan modem yang berfungsi sebagai adaptor antara komputer dan jaringan telepon. Modem dan saluran telepon bertanggung jawab untuk menghubungkan *Data Communication Equipment* (DCE) jarak jauh dengan *Data Terminal Equipment* (DTE). DCE mewakili komputer sedangkan DTE mewakili terminal, seperti dapat dilihat pada Gambar 3.20. ide utama, yang saat ini dicapai dengan cara yang sangat sederhana dan pada kecepatan yang jauh lebih tinggi, adalah menciptakan jaringan pertama.



Gambar 3.20. Serial communication

Serial Bus bekerja secara sinkron, di mana transfernya disinkronkan oleh sinyal eksternal / secara asinkron mentransfer bit satu demi satu di kecepatan bus. Perangkat terhubung konektor standar yang digunakan untuk RS-232C protokol (Gambar 3.21).

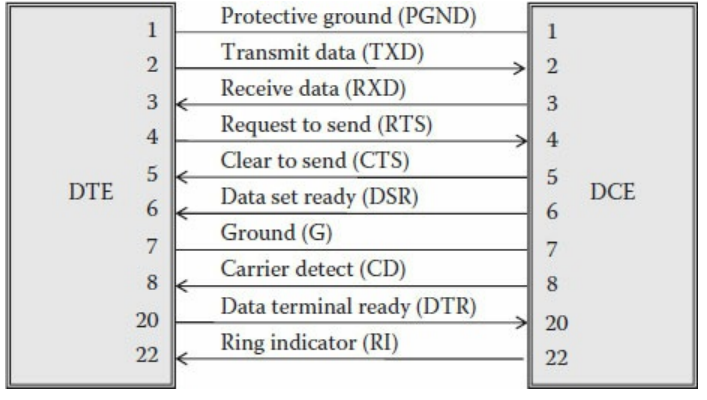


Gambar 3.21 RS-232C connector.

Perlu dicatat bahwa bus serial pada PC bekerja dalam dupleks penuh, dan ini berarti bus mengirim dan menerima data secara bersamaan. Untuk mengaktifkan komunikasi paralel, dua jalur terpisah digunakan mengirim dan menerima data. Ada kasus bus bekerja di *mode half-duplex*, yang berarti ia mampu mengirim atau menerima, dan kemudian hanya satu set kabel yang cukup.

Untuk half duplex, hanya bit 2 dan 7 yang digunakan. Bus serial, yang dimaksudkan untuk menghubungkan berbagai perangkat dan menyediakan awal komunikasi menggunakan jaringan telepon, adalah mampu terhubung ke computer.

Ketika PC pertama mulai menggunakan saluran telepon, karena keterbatasan saluran komunikasi adalah setengah dupleks. Selain menggunakan konektor standar, protokol mendefinisikan berbagai jalur koneksi dan harus terhubung di kedua sisi, seperti dapat dilihat pada Gambar 3.22.

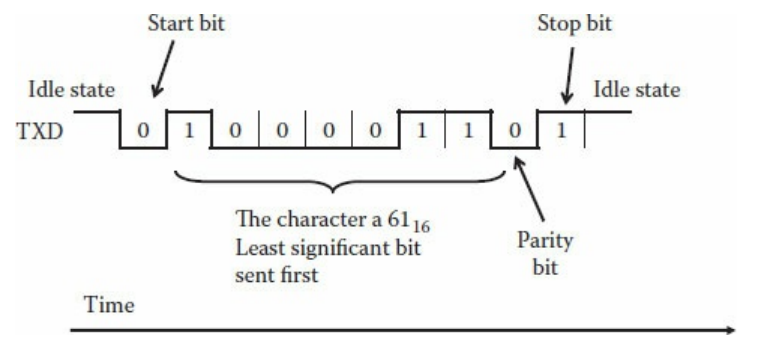


Gambar 3.22. koneksi RS-232

Arti dari berbagai bit (sinyal) yang dikirimkan pada bus adalah sebagai berikut:

* *Protective ground* (PGND) adalah sinyal referensi yang digunakan untuk menyinkronkan tegangan antara keduanya sisi. Di beberapa konektor, PGND terhubung ke ground (G).
* *Transmit data* (TXD) adalah sinyal yang digunakan mentransmisikan data (bit) dari DTE ke perangkat. Saat tidak ada komunikasi, sinyal diatur ke "ON" (logis 1).
* *Receive data* (RXD) adalah sinyal yang digunakan untuk menerima data (bit). Terhubung ke TXD on sisi lain.
* *Request to send* (RTS) adalah jalur kontrol yang digunakan untuk berkomunikasi dengan pihak lain dan untuk meminta bus untuk transmisi data.
* *Clear to send* (CTS) adalah jalur kontrol tambahan yang digunakan untuk memberi sinyal ke ujung lain bahwa perangkat siap menerima komunikasi. Dua garis (RTS & CTS) digunakan sebagai bagian dari kontrol aliran perangkat keras.
* *Data set ready* (DSR) adalah bit yang menandakan bahwa perangkat di ujung lain sudah siap (dinyalakan, terhubung ke bus, dan siap). Jika saluran telepon terputus, sinyal akan mati karena perangkat di ujung yang lain tidak terhubung.
* *Ground* (G) adalah jalur sinyal yang digunakan untuk menyinkronkan perangkat di kedua ujungnya. Dalam banyak kasus sinyal ini terhubung ke PGND.
* *Carrier detect* (CD) adalah saluran kontrol yang memberi sinyal bahwa perangkat telah mendeteksi pasangannya pada ujung yang lain. Dalam analogi dengan mesin faks, bit ini diatur ketika mesin berada di satu ujung berhasil terhubung ke mesin faks di ujung lain. Ini berfungsi sebagai lapisan tambahan komunikasi.
* *Data terminal ready* (DTR) adalah jalur kontrol yang memberi sinyal bahwa perangkat periferal siap untuk memulai komunikasi. Sinyal ini mirip dengan DSR, tetapi sementara DSR terkait dengan perangkat komunikasi di sisi local.
* *Ring indicator* (RI) adalah jalur kontrol yang memberi sinyal bahwa perangkat komunikasi telah mendeteksi panggilan masuk. Dalam analogi dengan standar telepon, sinyal inilah yang menyebabkan telepon mulai berdering. Hanya setelah kami menerima panggilan dapatkah sinyal telepon ke ujung lain bahwa panggilan dan orang di ujung lain dapat mulai berbicara.

Setiap grup yang dikirim pada serial bus dimulai dengan mengirimkan bit awal dan diakhiri dengan bit stop. Permulaan bit adalah logika "nol" yang menandakan bit berikut adalah data aktual. Bit stop, yang merupakan logis "satu," menandakan akhir blok. Dalam bit paritas tambahan yang ditambahkan ke transmisi untuk melindungi integritasnya, seperti yang akan dijelaskan nanti dalam bab ini. Gambar 3.23 menunjukkan komunikasi serial, menggambarkan transmisi karakter **"a."** Nilai biner adalah **1100001** dan bit paling tidak signifikan ditransmisikan terlebih dahulu.



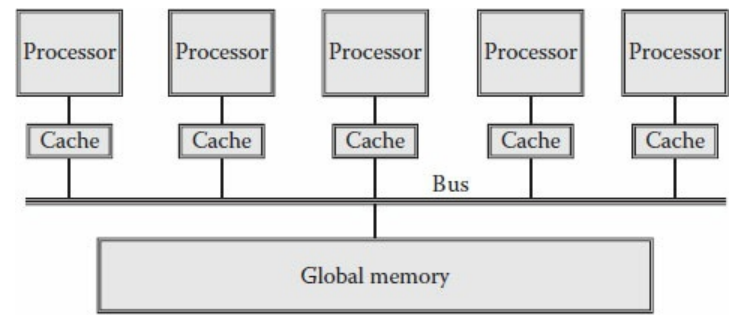
Gambar 3.23 komunikasi serial

Transmisi dimulai dengan mengirimkan bit awal, yang menandakan bahwa bit tambahan yang ditransmisikan mewakili data. Setelah bagian awal, di siklus berikut bit data ditransmisikan. Karena transmisi dimulai dengan yang paling tidak signifikan bit, bit pertama yang dikirim adalah **"satu",** yang akan diikuti oleh empat "nol" dan kemudian dua tambahan **" satu."** Panjang byte berupa 8 bit byte. paritas didefinisikan sebagai paritas genap atau ganjil. Dalam contoh ini, pengaturan transmisi diatur ke tujuh bit byte, sehingga karakter **"a"** diwakili oleh tujuh bit. Bit yang ditransmisikan berikutnya adalah paritas dan bit terakhir (***stop bit***) menandakan akhir komunikasi.

Meskipun penggunaan untuk bus serial ***RS-232C*** digantikan oleh USB, yang menjadi standar industri de facto. Popularitasnya fakta bahwa digunakan untuk komunikasi antara periferal komputer tetapi juga mendukung banyak peralatan lain, seperti: kamera, ponsel, pemutar media, dan sebagainya. Selanjutnya, USB juga digunakan untuk memasok daya listrik ke beberapa perangkat konsumsi daya yang lebih rendah.

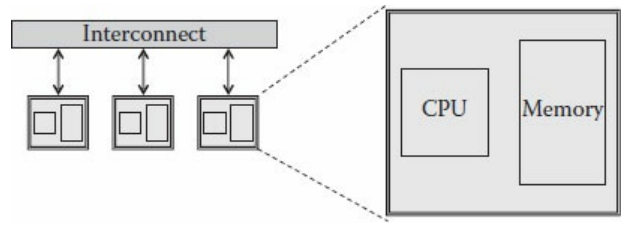
1. **Konsep Bus**

Sistem komersial modern didasarkan pada arsitektur PC, yang berarti bahwa komponen adalah orang-orang yang ada di PC. Bahkan dengan PC, bus terkadang membuat kemacetan jadi hirarki bus digunakan. Dalam sistem paralel besar yang mencakup banyak PC yang bekerja pada sistem umum memori, terkadang bahkan hierarki bus tidak mencukupi seperti yang dapat dilihat Gambar 3.24 menggambarkan arsitektur superkomputer J90\* yang dirancang dan diproduksi oleh Cray. Sistem menyertakan beberapa (8, 16, atau 32-bit) prosesor masing-masing dengan memori cache sendiri.



Gambar 3.24. Contoh sistem paralel

Setiap memori cache terhubung ke prosesor yang sesuai menggunakan bus internal. Sistem memiliki memori global yang sama untuk berbagi aplikasi dan data di antara berbagai prosesor. Komunikasi antara memori global dan prosesor dilakukan menggunakan bus cepat (1,6 GB/detik untuk setiap prosesor). Meskipun kecepatan transfernya tinggi, jelas bahwa untuk sistem yang lebih besar dengan jumlah prosesor yang lebih banyak, bus akan menjadi hambatan. Untuk itu Alasannya, ***superkomputer*** berikutnya yang dirancang oleh Cray (T3E) menerapkan pendekatan yang berbeda. Itu T3E menyertakan beberapa versi dan awalnya menggunakan ***chip Alpha*** yang sangat cepat dalam perhitungan *floating-point*. T3E dirancang untuk mendukung banyak prosesor, dan ***model entry-level*** memiliki 1480 prosesor. Itu juga komputer pertama yang mencapai 1 Teraflops (1012 floating-point perdetik). Untuk mencapai ini kecepatan, arsitektur itu harus dimodifikasi. Gambar 3.25 menjelaskan bagian dari arsitektur baru.



Gambar 3.25 Cray T3E architecture.

Struktur interkoneksi harus mendukung perpindahan (pertukaran) data yang diperlukan oleh modul-modul komputer sebagai berikut :

• Memori ke CPU

• CPU ke memori

• Perangkat I/O ke CPU

• CPU ke perangkat I/O

• Perangkat I/O ke memori atau dari

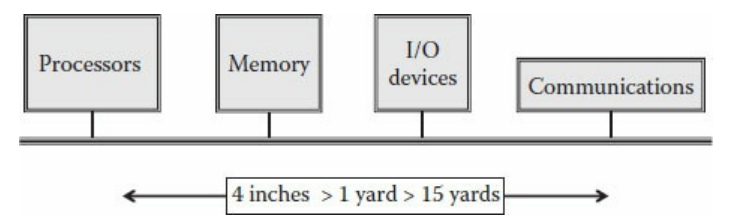
• Memori

perlu dicatat bahwa arsitektur bus yang didasarkan pada grid di mana setiap *processing elements (*PE) terhubung ke satu sama lain *processing elements (*PE) lebih cepat. Namun, jumlah komponen yang dihubungkan grid jenis ini sangat besar, implementasi grid menjadi sangat rumit karena jumlah koneksi yang diperlukan, nomor koneksi yang diperlukan untuk n elemen diberikan oleh :

Jadi, untuk komponen 1480 dan memori global, seperti yang dirancang untuk entry level T3E, nomornya saluran yang dibutuhkan adalah 1.094.460!

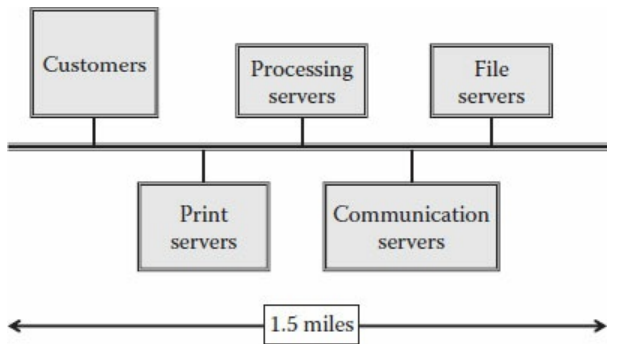
1. **Batas Sistem Ekspansi Bus**

Bus didefinisikan sebagai sarana untuk menghubungkan berbagai komponen dari sistem. Faktanya, komputer tahun 1970-an dan 1980-an menggunakan bus untuk menghubungkan komponen internal sistem. Dengan demikian, panjang bus sangat terbatas, berukuran maksimum beberapa kaki. Seiring perkembangan sistem, bus harus lebih Panjang, misalnya untuk menghubungkan printer yang terletak di ruangan yang berbeda. Perkembangan tambahan yang termasuk, misalnya, perangkat penyimpanan memerlukan peningkatan tambahan dalam panjang bus hingga 30 kaki (Gambar 3.26).



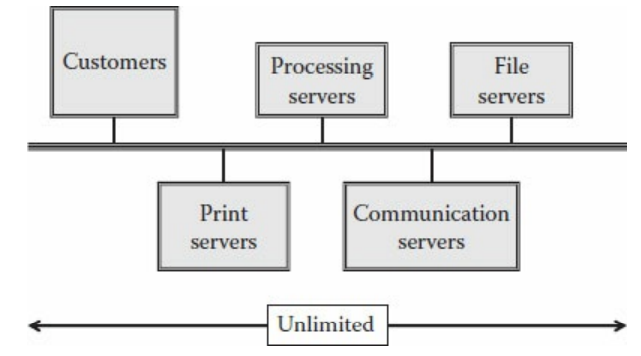
Gambar 3.26 Panjang bus selama tahun (1970-1990).

Tahun 1990-an dianggap sebagai era jaringan. ***Sun Microsystems*** bahkan mendefinisikan jaringan sebagai computer. Saat ini, dengan modern sistem, pengguna bekerja pada sistem virtual yang dapat mencakup berbagai server, masing-masing didedikasikan ke fungsi komputasi yang berbeda. Dalam, jaringan organisasi dan sistem yang terhubung dengannya membentuk ***sistem virtual kolaboratif***. Modus operasi ini dimulai pada 1990-an, di mana LAN dikembangkan. Awalnya, LAN dibatasi kurang dari satu mil, tetapi, berbagai repeater dan booster, panjangnya diperpanjang hingga satu setengah mil.



Gambar 3.27. 1990 bus (local area network).

Selama awal tahun 2000, Internet maju dengan sangat cepat menggantikan banyak konsep komputasi. Peningkatan kecepatan transmisi data dan variasi komunikasi alternatif telah mengubah industri komputasi. Saat ini, banyak sistem organisasi yang tersebar di seluruh dunia karena jarak bukan objek lagi. Selanjutnya, komputasi awan teknologi perlahan-lahan mengubah departemen komputasi lokal dengan mengalihdayakan potongan yang lebih besar aplikasi dan infrastruktur. Sejauh panjang bus dipertimbangkan, itu berarti "bus" baru telah menjadi tak terbatas (Gambar 3.28).



Gambar 3.28. tahun 2000 bus (Internet).

Untuk meningkatkan keandalan transmisi, beberapa metode dikembangkan selama bertahun-tahun. Metode dijelaskan tentang bus, metode ini juga berlaku untuk perangkat lain, misalnya, disk drive. Dalam kasus disk, controller harus memastikan bahwa data yang diterima dari disk benar-benar data yang ditulis dan tidak ada bit yang diubah dalam prosesnya. Selama bertahun-tahun, beberapa metode untuk perlindungan dan peningkatan keandalan dikembangkan. Yang paling metode dasar didasarkan pada ***error correction code*** (ECC). Beberapa metode menerapkan ECC, dan secara keseluruhan nkasus idenya adalah untuk menambah data yang dikirimkan dengan beberapa bit tambahan yang akan memungkinkan penerima untuk memeriksa bahwa transmisi telah diterima dengan benar.

KESIMPULAN

1. BUS

BUS adalah Jalur komunikasi yang dibagi pemakai dalam suatu jaringan, Suatu set kabel tunggal yang digunakan untuk menghubungkan berbagai sub-sistem, Dalam satu bus berisi sejumlah kanal (jalur), contohnya bus data 32 bit berisi 32 jalur, Jalur sumber tegangan pada bus biasanya tidak diperlihatkan.

1. SYSTEM BUS

SYSTEM BUS Sebuah Bus yang menghubungkan komponen-komponen utama komputer (CPU, Memori ,I/O), System bus dalam arsitektur komputer merujuk pada bus yang digunakan oleh sistem komputer untuk menghubungkan semua komponennya dalam menjalankan tugasnya, Sebuah bus adalah sebutan untuk jalur di mana data dapat mengalir dalam komputer, Jalur-jalur ini digunakan untuk komunikasi dan dapat dibuat antara dua elemen atau lebih.

Data atau program yang tersimpan dalam memori dapat diakses dan dieksekusi oleh CPU melalui perantara system bus, Sebuah komputer memiliki beberapa bus, agar dapat berjalan, Banyaknya bus yang terdapat dalam sistem, tergantung dari arsitektur sistem komputer yang digunakan.

1. CONTOH SYSTEM BUS

Sebuah komputer PC dengan prosesor umumnya ***Intel Pentium 4*** memiliki bus prosesor (front-side bus), bus AGP, bus PCI, bus USB, bus ISA (yang digunakan oleh keyboard dan mouse), dan bus-bus lainnya.

1. ORGANISASI BUS
2. Jalur Kontrol

Berisi signal request dan sinyal acknowledgements

Mengindikasikan tipe informasi pada jalur data.

1. Jalur Data

Membawa informasi antara sumber dan tujuan data dan alamat dan perintah-perintah kompleks

1. MASTER VS SLAVE

Suatu transaksi bus meliputi 2 komponen

* + Mengeluarkan perintah dan alamat – request (permintaan)
  + Memindahkan dat – action (tindakan)

Master :

Bus yang memulai transaksi bus dengan cara

- Mengeluarkan perintah dan alamat

Slave :

Bus yang bereaksi terhadap alamat dengan cara

- Mengirimkan data kepada master jika master meminta data

- Menerima data dari master jika master mengirim data

STRUKTUR BUS

1. BUS DATA

Saluran data memberikan lintasan bagi perpindahan data antara dua modul sistem .

Besar jalur bus data mempengaruhi kinerjanya, Lebar jalur bus menentukan performance : 8, 16, 32 dan 64 bit

1. ADDRESS BUS

Berfungsi menentukan asal atau tujuan dari data Misalkan: CPU perlu membaca instruksi (data) dari memori pada lokasi tertentu.

Lebar jalur bus menentukan kapasitas memori maksimum dari sistem

Contoh: 8080 memiliki 16 bit address bus artinya ruang memori maksimum yang tersedia adalah 64K.

Arsitektur Harvard menggunakan memori terpisah untuk program dan data dengan alamat dan bus data berdiri sendiri, Adanya dua perbedaan aliran data dan alamat menyebabkan tidak diperlukannya multiplexing alamat dan bus data.

1. BUS ALAMAT

Identifikasi sumber atau tujuan data pada bus data, mengalamati port-port I/O

Contoh : jika CPU akan membaca word ( 8, 16 atau 32 bit) data dari memori maka CPU akan menaruh alamat word yang dimaksud pada saluran alamat

**REFERENSI/DAFTAR PUSTAKA**

1. Andrew S. Tanenbaum, Structured Computer Organization Fifth Edition, Pearson Prentice Hall 2005
2. Willam Stallings, Organisasi&ArsitekturKomputerEdisikeenam, Prentice Hall 2003
3. Syahrul, Organisasi dan Arsitektur Komputer, Andi offset 2010,