
Chapter 6

Input/Output

Masalah-masalah Input/Output

- ⌘ Periferal yang bervariasi
 - ☑ Pengiriman jumlah data yang berbeda
 - ☑ Dengan kecepatan yang berbeda
 - ☑ Dalam format yang berbeda
- ⌘ Semua periferal I/O berkecepatan lebih lambat dari CPU dan RAM
- ⌘ Memerlukan modul I/O

Modul Input/Output

- ⌘ Interface ke CPU dan memori
 - ☑ Melalui sistem bus atau perpindahan utama
- ⌘ Interface ke satu atau lebih periferal
 - ☑ Melalui link yang sesuai

Peralatan External

⌘ Terbacca manusia

- ☑ Monitor, printer, keyboard

⌘ Terbacca mesin

- ☑ Pengawasan dan kontrol

- ☑ Sensor, aktuator, pita/disk magnetik

⌘ Komunikasi

- ☑ Modem

- ☑ Network Interface Card (NIC)

Fungsi Modul I/O Module

- ⌘ Kontrol dan timing

 - ☑ Mengkoordinasikan lalu lintas antara sumber daya internal dan perangkat external.

- ⌘ Komunikasi prosesor

- ⌘ Komunikasi perangkat

- ⌘ Data Buffering

- ⌘ Deteksi kesalahan

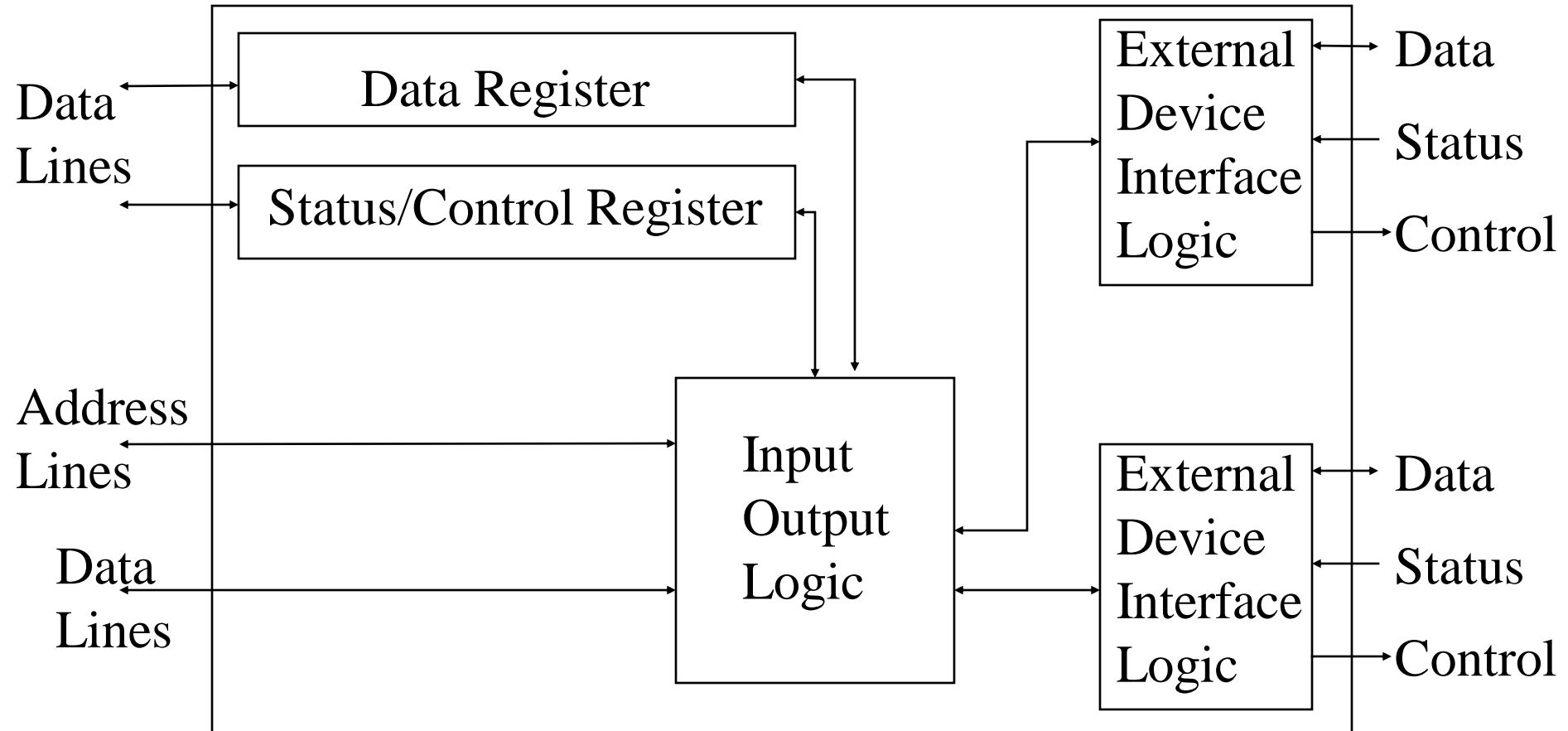
Langkah-langkah kontrol transfer data (external ke CPU) I/O

- ⌘ CPU meminta modul I/O untuk memeriksa status perangkat yang terhubung
- ⌘ Modul I/O menjawab status perangkat
- ⌘ Jika sedang on dan siap mengirim, CPU minta transfer data, dng perintah tertentu ke modul I/O
- ⌘ Modul I/O akan memperoleh unit data (mis 8 atau 16 bit) dari perangkat ext
- ⌘ Data akan ditransfer dari modul I/O ke prosesor

Diagram Blok Modul I/O

Systems Bus Interface

External Device Interface



Teknik Input Output

⌘ I/O Terprogram

⌘ I/O Interrupt driven

⌘ Direct Memory Access (DMA)

I/O Terprogram

- ⌘ Antara CPU dengan I/O saling menukarkan data
 - ☑ Status perangkat sensor
 - ☑ Perintah Read/write
 - ☑ Transfer data
- ⌘ Ketika CPU memberi perintah modul I/O, maka CPU menunggu modul I/O menyelesaikan operasinya
- ⌘ Jika CPU lebih cepat dari modul I/O, maka membuang waktu CPU

I/O Terprogram - detail

- ⌘ CPU meminta I/O melakukan operasi
- ⌘ Modul I/O melakukan operasi
- ⌘ Modul I/O menetapkan bit status
- ⌘ CPU memeriksa bit status secara periodik
- ⌘ Modul I/O tidak melaporkannya ke CPU
- ⌘ Modul I/O tidak meng-interrupt CPU
- ⌘ CPU akan menunggu atau kembali lagi

Perintah-perintah I/O (1)

⌘ CPU dan alamat

- ☑ Prosesor mengeluarkan alamat yang menspesifikasi modul I/O dan perangkat ext, serta perintah I/O

⌘ Perintah I/O

- ☑ Ketika modul I/O dialamati oleh CPU, yaitu:
- ☑ Control – mengaktifkan periferal dan memberi tahu apa yang harus dilakukan
 - ☒ e.g. unit pita magnetik yang diinstruksikan untuk menggulung ulang atau memajukan sebuah rekaman (perintah ini dikhususkan ke jenis perangkat periferalnya)

Perintah-perintah I/O (2)

- ☒ Test – menguji berbagai macam kondisi status yg berhubungan dengan perangkat periferalnya
 - ☒ e.g. power? Error?
- ☒ Read – modul I/O akan memperoleh data dari periferal dan menempatkannya pada buffer internal.
- ☒ Write – modul I/O mengambil data dari bus data dan kemudian mentransmisikan data tersebut ke periferal

Pengalamatan perangkat I/O

- ⌘ Pada I/O terprogram, transfer data sangat mirip dengan akses memori
- ⌘ Setiap perangkat diberi kode pengenalan yang unik
- ⌘ Perintah-perintah CPU terdiri dari kode pengenalan (alamat)

Pemetaan I/O

⌘ Memori pemetaan I/O

- ☑ Perangkat I/O dan memori berbagi sebuah ruang alamat
- ☑ I/O terlihat mirip dengan memori read / write
- ☑ Tidak ada perintah khusus untuk I/O

⌘ I/O terisolasi

- ☑ Ruang alamat I/O terpisah dengan ruang alamat memori
- ☑ Memerlukan pemilihan jalur I/O atau memori
- ☑ Terdapat perintah khusus untuk I/O

I/O Interrupt - Driven

- ⌘ CPU harus menunggu
- ⌘ Tidak ada pengecekan ulang perangkat I/O
- ⌘ Modul I/O akan melakukan interrupt bila siap

I/O Interrupt - Driven

Operasi dasar

- ⌘ CPU memerintahkan read
- ⌘ Modul I/O mendapatkan data dari periferal dengan saat yang bersamaan CPU melakukan kerja yang lain
- ⌘ Modul I/O menginterrupt CPU
- ⌘ CPU meminta data
- ⌘ Modul I/O melakukan transfer data

CPU (Prosesor)

- ⌘ Memerintahkan read
- ⌘ Melakukan kerja yang lain
- ⌘ Cek untuk interrupt disetiap akhir putaran instruksi
- ⌘ Dilakukan interrupt, jika:
 - ☑ Menyimpan data (register)
 - ☑ Proses interrupt
 - ☒ Mendapatkan data & menyimpannya
- ⌘ Lihat catatan tentang OS

Masalah Perancangan

- ⌘ Bagaimana mengidentifikasi modul melakukan interrupt?
- ⌘ Bagaimana menangani multiple interrupt?

Identifikasi Modul Interupsi (1)

- ⌘ Jalur yang berbeda untuk setiap modul
- ⌘ Poll Software
- ⌘ Daisy Chain atau Hardware poll
- ⌘ Bus Master
 - ☒ Module must claim the bus before it can raise interrupt
 - ☒ e.g. PCI & SCSI

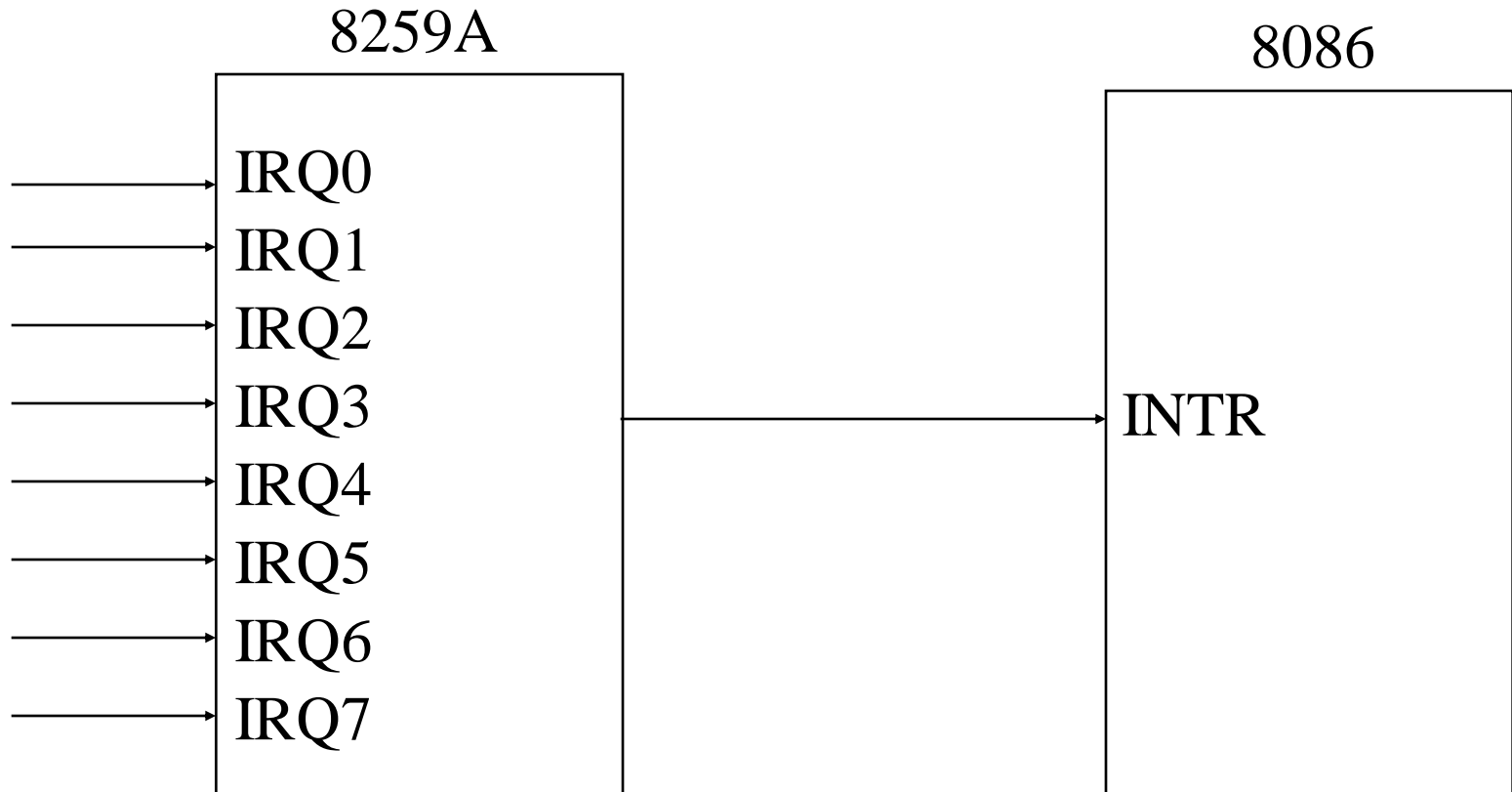
Multiple Interrupts

- ⌘ Setiap jalur interrupt mempunyai prioritas
- ⌘ Jalur prioritas utama bisa melakukan interrupt jalur yang berprioritas lebih rendah
- ⌘ Jika

Example - PC Bus

- ⌘ 80x86 mempunyai satu jalur interrupt
- ⌘ Sistem 8086 menggunakan sebuah 8259A interrupt controller
- ⌘ 8259A mempunyai 8 jalur interrupt

PC Interrupt Layout



Rangkaian kejadian

- ⌘ 8259A menerima interrupt
- ⌘ 8259A menentukan prioritas
- ⌘ 8259A memberi sinyal ke CPU (peningkatan jalur INTR)
- ⌘ CPU melakukan Acknowledges (jalur INTA)
- ⌘ 8259A menempatkan informasi vector yang sesuai pada bus data
- ⌘ CPU melakukan proses interrupt

Foreground Reading

⌘ <http://www.pcguide.com/ref/mbsys/res/irq/func.htm>

⌘ In fact look at <http://www.pcguide.com/>

Direct Memory Access (Akses memori langsung)

- ⌘ I/O terprogram dan I/O Interrupt driven memiliki kelemahan
 - ☒ Kecepatan Transfer I/O terbatas. Dimana dengan kecepatan itu prosesor dapat menguji dan melayani perangkat
 - ☒ CPU ditentukan oleh pengaturan transfer I/O
- ⌘ DMA is the answer, jika data yang akan dipindahkan sangat besar

Fungsi DMA

- ⌘ Modul tambahan pada bus sistem
- ⌘ Modul DMA dapat menirukankan CPU dan mengambil alih kontrol sistem dari CPU

Operasi DMA

- ⌘ CPU mengirim perintah ke DMA :
 - ⊠ Read/Write
 - ⊠ Alamat perangkat
 - ⊠ Penempatan awal memori
 - ⊠ Jumlah data (word) yang akan ditransfer
- ⌘ CPU melanjutkan pekerjaan lain
- ⌘ DMA controller akan memindahkan data tanpa melalui CPU
- ⌘ DMA controller setelah selesai mengirim sinyal interupsi ke CPU
- ⌘ CPU hanya terlibat di awal dan akhir transfer

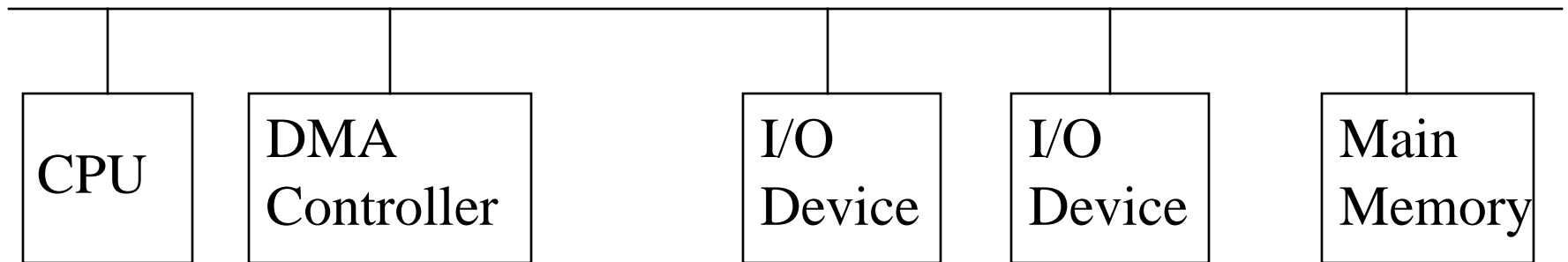
Pencurian siklus DMA Transfer

- ⌘ DMA controller mengambil alih bus sistem untuk sebuah siklus dari CPU
- ⌘ Mengirim satu word data
- ⌘ Tidak ada interrupt (terhadap CPU)
- ⌘ CPU menghentikan operasi untuk sementara
 - ⏏ i.e. sebelum sebuah instruksi atau mengambil data atau menulis data

Pandangan lain

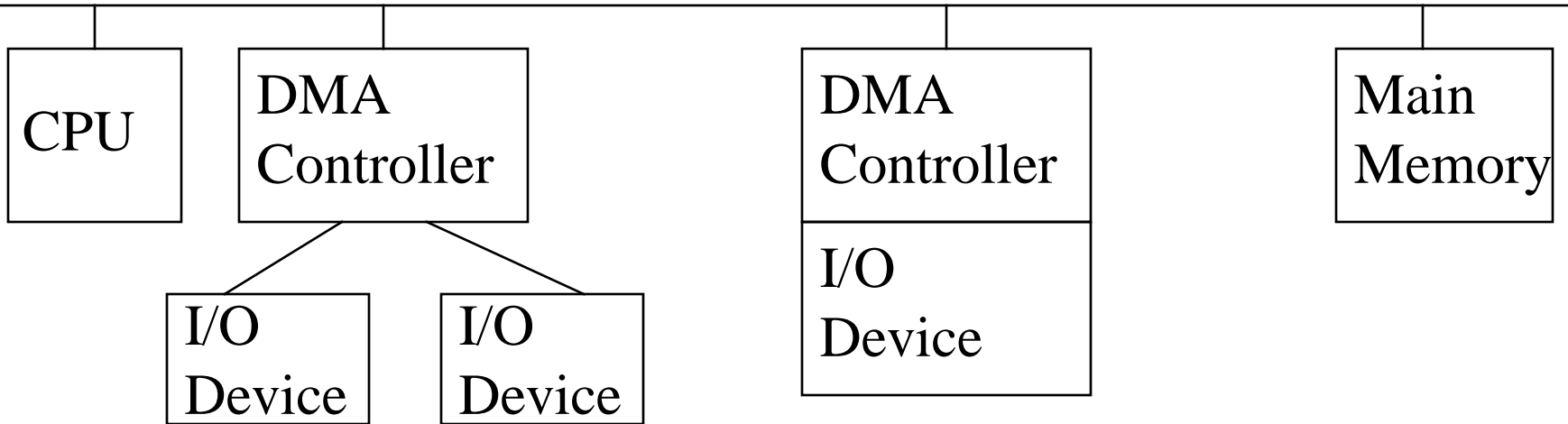
- ⌘ Apa akibat dari cache memori mempunyai DMA
- ⌘ Hitung: berapa banyak bus sistem yang bisa digunakan

DMA Configurations (1)



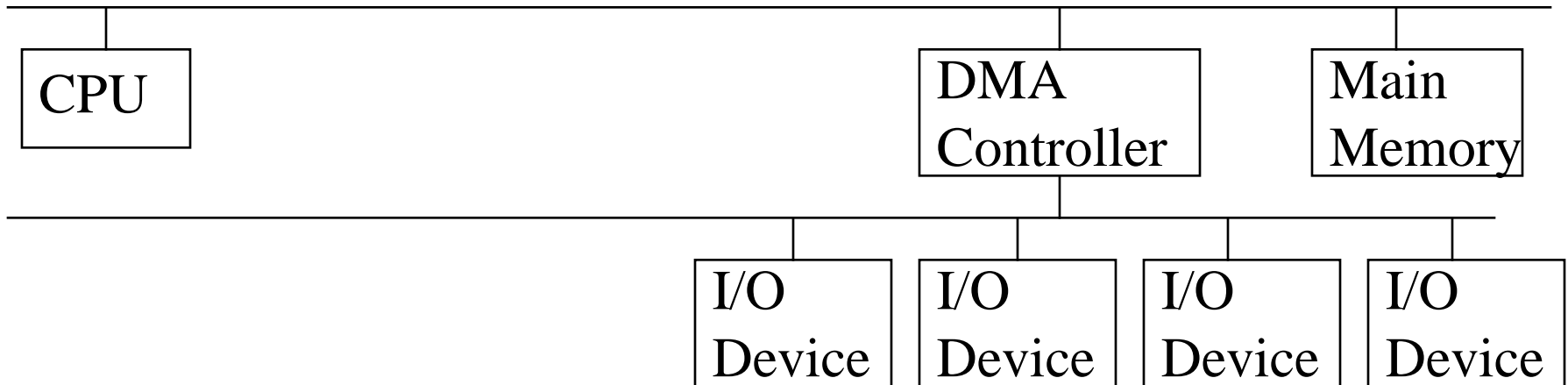
- ⌘ Single Bus, DMA controller terpisah
- ⌘ Setiap transfer menggunakan bus dua kali
 - ☑ I/O ke DMA kemudian DMA ke memory
- ⌘ CPU terhenti kerjanya dua kali

DMA Configurations (2)



- ⌘ Single Bus, Integrated DMA controller
- ⌘ Controller bisa menangani >1 perangkat
- ⌘ Setiap transfer menggunakan bus satu kali
 - ☑ DMA ke memory
- ⌘ CPU terhenti kerjanya satu kali

DMA Configurations (3)



- ⌘ Pemisahan bus I/O
- ⌘ Bus mendukung semua perangkat DMA
- ⌘ Setiap transfer menggunakan bus satu kali
 - ⌘ DMA ke memory
- ⌘ CPU terhenti kerjanya satu kali

I/O Channels

- ⌘ Perangkat I/O semakin rumit
 - ☑ e.g. 3D graphics cards
- ⌘ CPU memerintahkan I/O controller melakukan transfer
- ⌘ I/O controller melakukan semua transfer
- ⌘ Kinerja
 - ☑ Beban kerja CPU berkurang
 - ☑ Kinerja keseluruhan meningkat

Interfacing (antar muka)

- ⌘ Menghubungkan beberapa perangkat menjadi satu
- ⌘ Serial atau paralel?
- ⌘ Diperuntukkan processor/memory/buses?
 - ☑ E.g. SCSI, FireWire, Infiniband

Foreground Reading

- ⌘ Check out Universal Serial Bus (USB)
- ⌘ Compare with other communication standards
e.g. Ethernet