

BUKU AJAR

MATA KULIAH ARSITEKTUR KOMPUTER

BAB 13

INPUT/OUTPUT DAN INTERKONEKSI SISTEM KOMPUTER

**Buku Ajar Arsitektur Komputer 2026
Berbasis Rencana Pembelajaran Semester (RPS)
IF221101 | 3 SKS**

IDENTITAS BAB

Komponen	Keterangan
Mata Kuliah	Arsitektur Komputer
Kode	IF221101
Bobot	3 SKS
Bab	Bab 13
Judul	Input/Output dan Interkoneksi Sistem Komputer
Fokus Pembelajaran	Memahami komunikasi antara CPU, memori, dan perangkat I/O melalui sistem interkoneksi serta berbagai metode transfer data
Level Pembelajaran	Pemahaman konsep, analisis alur transfer data, evaluasi metode I/O, dan simulasi mekanisme interrupt serta DMA

CAPAIAN PEMBELAJARAN BAB

Setelah mempelajari bab ini, mahasiswa diharapkan mampu:

1. Menjelaskan fungsi sistem input/output dalam organisasi komputer.
2. Mengidentifikasi komponen utama dalam proses komunikasi CPU, memori, dan perangkat I/O.
3. Menjelaskan konsep interkoneksi sistem dan bus.
4. Membedakan bus data, bus alamat, dan bus kontrol.
5. Menjelaskan fungsi modul atau controller I/O.
6. Membedakan programmed I/O, interrupt-driven I/O, dan Direct Memory Access.
7. Menganalisis alur penanganan interrupt.
8. Menjelaskan konsep interrupt priority dan interrupt masking.
9. Menjelaskan peran DMA controller dalam transfer blok data.
10. Menganalisis trade-off penggunaan programmed I/O, interrupt, dan DMA.
11. Membuat simulasi sederhana alur transfer data perangkat I/O.

PETA KONSEP

ALUR KONSEP BAB 13

CPU ↔ Interkoneksi Sistem ↔ Memori ↔ Modul I/O ↔ Perangkat Eksternal → Bus Data/Alamat/Kontrol → Programmed I/O → Interrupt-Driven I/O → DMA → Interrupt Priority → Transfer Data → Sinkronisasi dan Kinerja.

13.1 Pendahuluan

Sistem komputer tidak hanya terdiri atas CPU dan memori. Komputer harus menerima masukan, mengirimkan keluaran, berkomunikasi dengan perangkat penyimpanan, jaringan, sensor, dan berbagai perangkat eksternal lainnya. Komunikasi tersebut memerlukan mekanisme yang dapat menjembatani perbedaan karakteristik antara CPU dan perangkat I/O.

CPU bekerja dengan organisasi dan kecepatan tertentu, sedangkan perangkat I/O memiliki karakteristik transfer data yang sangat beragam. Karena itu, sistem komputer menggunakan interkoneksi, modul pengendali, register status, mekanisme sinkronisasi, interrupt, dan pada kondisi tertentu DMA.

PERTANYAAN PEMANTIK

Jika CPU harus menunggu setiap karakter dari keyboard atau setiap blok dari perangkat penyimpanan, bagaimana pengaruhnya terhadap efisiensi sistem? Bagaimana perangkat dapat memberi tahu CPU bahwa operasi telah selesai?

13.2 Konsep Input/Output

Input/output adalah mekanisme pertukaran data antara sistem komputer dengan lingkungan eksternal atau perangkat lain. Input membawa data menuju sistem, sedangkan output mengirimkan hasil atau informasi dari sistem ke luar.

Kategori	Contoh Fungsi
Input	Mengirim data atau sinyal dari perangkat menuju sistem
Output	Mengirim informasi dari sistem menuju perangkat
Storage I/O	Pertukaran data dengan media penyimpanan
Communication I/O	Pertukaran data melalui jaringan atau antarmuka komunikasi

Dalam organisasi komputer, perangkat I/O umumnya tidak dihubungkan langsung ke setiap bagian internal CPU. Modul atau controller I/O membantu mengelola komunikasi dan menyediakan antarmuka yang sesuai.

13.3 Komponen Sistem Input/Output

Komponen	Peran Konseptual
CPU	Mengendalikan operasi dan menjalankan program
Memori	Menyimpan instruksi serta data
System Interconnection	Menyediakan jalur komunikasi antar komponen
I/O Module atau Controller	Menjembatani sistem internal dengan perangkat eksternal
External Device	Perangkat yang menghasilkan atau menerima data

Modul I/O dapat menangani informasi data, status, dan kontrol. CPU tidak selalu berinteraksi langsung dengan karakteristik fisik perangkat.

13.4 Interkoneksi Sistem

Interkoneksi sistem adalah mekanisme yang memungkinkan CPU, memori, dan modul I/O bertukar informasi. Salah satu model penting dalam pembelajaran arsitektur komputer adalah bus.

Bus dapat dipahami sebagai kumpulan jalur sinyal dan aturan komunikasi yang digunakan oleh beberapa komponen. Organisasi nyata dapat menggunakan interkoneksi yang lebih kompleks, tetapi konsep bus tetap penting untuk memahami aliran data dan sinyal kontrol.

13.5 Sistem Bus

Dalam model dasar, bus dapat dikelompokkan menjadi bus data, bus alamat, dan bus kontrol.

Jenis Bus	Fungsi
Bus Data	Membawa data atau bagian instruksi antar komponen
Bus Alamat	Membawa informasi lokasi tujuan atau sumber yang dapat dialamatkan
Bus Kontrol	Membawa sinyal untuk mengatur operasi, pembacaan, penulisan, interrupt, dan fungsi kendali lainnya

CATATAN KONSEPTUAL

Lebar bus data memengaruhi jumlah bit yang dapat ditransfer dalam satu transaksi model tertentu, sedangkan lebar jalur alamat berkaitan dengan jumlah lokasi yang dapat direpresentasikan.

13.6 Operasi Dasar Bus

Sebuah transaksi bus secara konseptual melibatkan pihak yang meminta transfer, informasi alamat atau tujuan, data bila diperlukan, dan sinyal kontrol yang menunjukkan jenis operasi.

Tahap	Aktivitas Konseptual
Request	Komponen meminta penggunaan jalur komunikasi
Address/Command	Tujuan dan jenis operasi disampaikan
Data Transfer	Data dikirim atau diterima
Completion	Operasi dikonfirmasi selesai sesuai mekanisme sistem

13.7 Sinkronisasi Bus

Komunikasi antar komponen memerlukan koordinasi waktu. Secara konseptual, bus dapat menggunakan mekanisme synchronous atau asynchronous.

Pendekatan	Karakteristik
Synchronous	Transaksi dikaitkan dengan sinyal clock atau siklus waktu yang teratur
Asynchronous	Komunikasi menggunakan sinyal koordinasi tanpa harus bergantung pada satu clock global

Pemilihan mekanisme dipengaruhi oleh karakteristik komponen, jarak, dan kebutuhan fleksibilitas komunikasi.

13.8 Modul I/O dan Controller

Modul I/O berfungsi sebagai perantara antara interkoneksi sistem dan perangkat eksternal. Modul tersebut dapat menyimpan data sementara, menyediakan informasi status, menerima perintah kontrol, serta membantu sinkronisasi.

Bagian Konseptual	Fungsi
Data Register/Buffer	Menyimpan data yang sedang ditransfer
Status Register	Menyediakan informasi kondisi perangkat atau operasi
Control Register	Menerima perintah atau konfigurasi
Control Logic	Mengatur komunikasi dan urutan operasi

13.9 Memory-Mapped I/O dan Isolated I/O

Dalam organisasi I/O, register perangkat dapat diakses menggunakan model ruang alamat tertentu. Dua konsep yang umum digunakan untuk pembelajaran adalah memory-mapped I/O dan isolated I/O.

Pendekatan	Konsep
Memory-Mapped I/O	Register perangkat menggunakan bagian dari ruang alamat yang dapat diakses dengan mekanisme pengalamatan tertentu
Isolated I/O	Ruang atau mekanisme I/O dibedakan dari pengalamatan memori umum sesuai aturan arsitektur

Cara akses aktual bergantung pada ISA dan organisasi perangkat keras yang digunakan.

13.10 Programmed I/O

Pada programmed I/O, CPU menjalankan instruksi untuk memeriksa status perangkat dan melakukan transfer data. Salah satu pola yang sering dibahas adalah polling, yaitu CPU secara berulang memeriksa apakah perangkat siap.

ALUR SEDERHANA

CPU mengirim perintah → CPU memeriksa status → perangkat siap → CPU membaca atau menulis data → proses dilanjutkan.

Pendekatan ini sederhana untuk dipahami, tetapi polling dapat menggunakan waktu CPU ketika perangkat bekerja jauh lebih lambat daripada prosesor.

13.11 Polling

Polling adalah teknik ketika CPU secara aktif melakukan pemeriksaan terhadap register status perangkat. Jika status belum menunjukkan kesiapan, CPU dapat mengulang pemeriksaan.

Kelebihan	Keterbatasan
Mekanisme relatif sederhana	CPU dapat menghabiskan waktu untuk pemeriksaan berulang
Kontrol alur mudah diamati	Kurang efisien jika waktu tunggu panjang
Sesuai untuk kondisi tertentu	Tidak selalu ideal untuk banyak perangkat

13.12 Interrupt-Driven I/O

Pada interrupt-driven I/O, CPU tidak harus terus-menerus melakukan polling. Perangkat atau modul I/O dapat mengirimkan sinyal interrupt ketika membutuhkan perhatian atau ketika operasi tertentu selesai.

Ketika interrupt diterima, CPU secara konseptual menyelesaikan titik aman sesuai mekanisme arsitektur, menyimpan state yang diperlukan, menentukan sumber atau jenis interrupt, lalu menjalankan routine penanganan sebelum kembali ke alur sebelumnya.

13.13 Siklus Penanganan Interrupt

Tahap	Aktivitas Konseptual
Interrupt Request	Perangkat atau controller mengirim permintaan interrupt
Recognition	CPU mengenali adanya interrupt sesuai aturan
Save Context	State yang diperlukan disimpan
Vector/Identification	Sumber atau handler ditentukan
Service Routine	Interrupt service routine dijalankan
Restore Context	State sebelumnya dipulihkan
Return	Eksekusi program utama dilanjutkan

KONSEP PENTING

Detail urutan dan register yang disimpan bergantung pada arsitektur prosesor serta mekanisme sistem operasi.

13.14 Interrupt Priority dan Masking

Dalam sistem dengan banyak sumber interrupt, beberapa permintaan dapat muncul pada waktu yang berdekatan. Sistem memerlukan aturan untuk menentukan urutan pelayanan.

Konsep	Fungsi
Priority	Menentukan tingkat kepentingan relatif suatu interrupt

Masking	Menunda atau menonaktifkan penerimaan kelompok interrupt tertentu
Nested Interrupt	Dalam kondisi tertentu, handler dapat menerima interrupt lain sesuai aturan prioritas

Desain mekanisme tersebut bertujuan menyeimbangkan kebutuhan respons cepat dengan konsistensi state sistem.

13.15 Direct Memory Access (DMA)

DMA digunakan ketika transfer data dalam jumlah lebih besar tidak efisien jika setiap unit harus ditangani langsung oleh CPU. DMA controller dapat mengelola transfer antara perangkat I/O dan memori sesuai parameter yang telah disiapkan.

ALUR KONSEPTUAL DMA

CPU mengonfigurasi DMA → perangkat menghasilkan atau menerima data → DMA mengelola transfer dengan memori → operasi selesai → CPU diberi informasi penyelesaian.

Dengan demikian, CPU dapat mengurangi jumlah instruksi yang diperlukan untuk menangani setiap unit transfer secara langsung.

13.16 Mode Transfer DMA

Berbagai model konseptual dapat digunakan untuk menjelaskan cara DMA berbagi akses terhadap interkoneksi sistem.

Mode	Gagasan Konseptual
Burst/Block Transfer	DMA mentransfer sejumlah data dalam satu rangkaian akses sesuai mekanisme arbitrase
Cycle Stealing	DMA memperoleh kesempatan akses tertentu sehingga CPU dan DMA berbagi sumber daya
Transparent/Idle Cycle	Transfer dilakukan ketika sumber daya tidak digunakan oleh aktivitas tertentu, bergantung organisasi sistem

Implementasi aktual dapat menggunakan istilah dan mekanisme yang lebih spesifik.

13.17 Bus Arbitration

Jika lebih dari satu komponen ingin menggunakan interkoneksi bersama, diperlukan proses arbitrase. Arbitrase menentukan pihak yang memperoleh hak menggunakan jalur pada suatu waktu.

Pendekatan	Gagasan
Centralized	Keputusan diberikan oleh satu mekanisme atau arbiter pusat

Distributed	Komponen menggunakan aturan koordinasi tanpa satu pengendali tunggal
-------------	--

Tujuan arbitrase adalah menghindari konflik dan menyediakan urutan akses yang sesuai dengan kebijakan sistem.

13.18 Perbandingan Metode I/O

Metode	Peran CPU	Kesesuaian Konseptual
Programmed I/O	CPU aktif mengelola dan memeriksa operasi	Transfer sederhana atau kondisi yang tidak membutuhkan mekanisme kompleks
Interrupt-Driven I/O	CPU bekerja pada tugas lain hingga interrupt diterima	Operasi yang tidak perlu terus-menerus dipantau
DMA	CPU melakukan konfigurasi, transfer utama ditangani controller	Transfer blok atau volume data yang lebih besar

13.19 Studi Kasus 1: Keyboard dan Polling

Sebuah program membutuhkan masukan dari perangkat. Dalam model polling, CPU memeriksa status perangkat hingga tersedia data. Ketika status siap, CPU membaca data dari register perangkat.

Mahasiswa diminta menggambarkan berapa kali pemeriksaan dilakukan dalam sebuah simulasi dan menganalisis waktu CPU yang digunakan untuk menunggu.

13.20 Studi Kasus 2: Penyelesaian Transfer dengan Interrupt

Sebuah perangkat memulai operasi yang membutuhkan waktu lebih lama daripada beberapa instruksi CPU. CPU dapat melanjutkan tugas lain. Setelah perangkat selesai, controller mengirim interrupt dan handler memproses status operasi.

ANALISIS

Bandingkan jumlah aktivitas CPU pada model polling dan interrupt-driven I/O. Identifikasi keuntungan dan biaya tambahan dari penyimpanan serta pemulihan context.

13.21 Studi Kasus 3: Transfer File Menggunakan DMA

Sebuah perangkat perlu memindahkan blok data ke memori. CPU terlebih dahulu memberikan alamat awal, ukuran transfer, dan parameter operasi kepada DMA controller. Controller kemudian menangani proses transfer sesuai aturan sistem.

Mahasiswa membuat diagram alur CPU → DMA Controller → Memori ↔ Perangkat dan menunjukkan kapan CPU menerima notifikasi penyelesaian.

13.22 Hubungan I/O dengan Kinerja Sistem

Kinerja sistem tidak hanya ditentukan oleh kecepatan CPU. Perangkat I/O, bandwidth interkoneksi, waktu tunggu, frekuensi interrupt, dan strategi transfer dapat memengaruhi keseluruhan waktu penyelesaian pekerjaan.

Sistem dengan CPU cepat tetap dapat mengalami hambatan jika transfer data melalui perangkat atau jalur komunikasi menjadi bottleneck. Oleh karena itu, analisis arsitektur harus mempertimbangkan aliran data secara menyeluruh.

13.23 Aktivitas Pembelajaran Mahasiswa

Aktivitas 1: Peta Interkoneksi

Buat diagram hubungan CPU, memori, bus, controller I/O, dan dua perangkat eksternal.

Aktivitas 2: Analisis Bus

Identifikasi informasi yang dibawa oleh bus data, alamat, dan kontrol dalam tiga contoh transaksi.

Aktivitas 3: Simulasi Polling

Buat pseudocode yang melakukan polling pada register status perangkat.

Aktivitas 4: Simulasi Interrupt

Buat diagram urutan dari interrupt request hingga return ke program utama.

Aktivitas 5: DMA

Simulasikan transfer blok dengan parameter alamat awal, jumlah data, dan status selesai.

Aktivitas 6: Perbandingan

Bandingkan programmed I/O, interrupt-driven I/O, dan DMA berdasarkan penggunaan CPU dan jumlah transfer.

Aktivitas 7: Analisis Bottleneck

Identifikasi kemungkinan bottleneck pada sistem dengan CPU cepat tetapi perangkat I/O lambat.

13.24 Rangkuman

- Input/output memungkinkan sistem komputer berkomunikasi dengan perangkat eksternal.
- Komunikasi antar CPU, memori, dan I/O membutuhkan mekanisme interkoneksi.
- Bus data membawa data, bus alamat membawa informasi lokasi, dan bus kontrol membawa sinyal pengendali.
- Modul I/O menyediakan antarmuka antara sistem internal dan perangkat eksternal.
- Register data, status, dan kontrol membantu CPU berkomunikasi dengan perangkat.
- Programmed I/O menggunakan keterlibatan aktif CPU dalam proses transfer.
- Polling merupakan pemeriksaan berulang terhadap status perangkat.
- Interrupt-driven I/O memungkinkan CPU melakukan pekerjaan lain hingga perangkat memerlukan perhatian.

- Penanganan interrupt melibatkan pengenalan, penyimpanan context, pelayanan, pemulihan state, dan kembali ke alur sebelumnya.
- Priority dan masking digunakan untuk mengelola banyak sumber interrupt.
- DMA memungkinkan controller melakukan transfer data antara perangkat dan memori dengan keterlibatan CPU yang lebih rendah.
- Bus arbitration mengatur akses ketika beberapa komponen menggunakan sumber daya interkoneksi bersama.
- Pemilihan metode I/O memerlukan pertimbangan volume transfer, penggunaan CPU, kompleksitas, dan kebutuhan respons sistem.

13.25 Glosarium Bab

Istilah	Pengertian
Bus	Jalur dan aturan komunikasi antar komponen sistem.
Bus Address	Jalur yang membawa informasi lokasi atau tujuan transaksi.
Bus Arbitration	Mekanisme penentuan pihak yang memperoleh akses terhadap interkoneksi bersama.
Bus Control	Jalur sinyal untuk mengendalikan operasi komunikasi.
Bus Data	Jalur yang membawa data antar komponen.
DMA	Mekanisme transfer data antara perangkat dan memori dengan controller khusus.
I/O Controller	Komponen yang mengelola komunikasi dengan perangkat I/O.
Interrupt	Sinyal yang meminta CPU menangani suatu kejadian.
Interrupt Handler	Routine yang menangani permintaan interrupt.
Interrupt Priority	Aturan tingkat kepentingan relatif antar interrupt.
Memory-Mapped I/O	Pendekatan akses register perangkat melalui organisasi ruang alamat tertentu.
Polling	Pemeriksaan status perangkat secara berulang oleh CPU.
Programmed I/O	Transfer I/O yang dikendalikan secara langsung melalui aktivitas program CPU.
Status Register	Register yang menyimpan informasi kondisi perangkat.
System Interconnection	Mekanisme komunikasi antara CPU, memori, dan I/O.

13.26 Soal Latihan

A. Soal Pemahaman

12. Jelaskan fungsi sistem input/output dalam komputer.

13. Apa fungsi modul atau controller I/O?
14. Bedakan bus data, bus alamat, dan bus kontrol.
15. Jelaskan perbedaan komunikasi synchronous dan asynchronous.
16. Apa yang dimaksud programmed I/O?
17. Jelaskan mekanisme polling.
18. Jelaskan urutan konseptual penanganan interrupt.
19. Mengapa sistem membutuhkan interrupt priority?
20. Apa fungsi DMA controller?
21. Jelaskan perbedaan memory-mapped I/O dan isolated I/O secara konseptual.

B. Soal Analisis

22. Sebuah perangkat baru siap setelah 100 siklus pemeriksaan. Analisis penggunaan CPU jika polling dilakukan setiap siklus.
23. Buat diagram transaksi pembacaan data dari perangkat I/O melalui bus.
24. Bandingkan programmed I/O dan interrupt-driven I/O untuk perangkat yang menghasilkan data secara tidak teratur.
25. Sebuah sistem harus memindahkan blok data besar. Jelaskan alasan penggunaan DMA dibandingkan programmed I/O.
26. Analisis dampak jika beberapa perangkat dengan prioritas berbeda mengirim interrupt hampir bersamaan.
27. Jelaskan bagaimana bus arbitration dapat menghindari konflik akses antara CPU dan DMA.

C. Soal HOTS dan Studi Kasus

28. Rancang mekanisme I/O konseptual untuk sistem yang menerima data sensor secara berkala dan juga melakukan transfer file besar. Tentukan metode transfer yang sesuai untuk masing-masing.
29. Sebuah aplikasi mengalami penurunan respons karena CPU banyak digunakan untuk polling. Analisis alternatif desain dan trade-off yang mungkin muncul.
30. Buat skenario nested interrupt dengan tiga perangkat yang memiliki prioritas berbeda. Jelaskan urutan pelayanan yang Anda pilih.
31. Analisis bottleneck pada sistem yang memiliki CPU cepat, memori cepat, tetapi bandwidth interkoneksi I/O terbatas.
32. Rancang diagram dan pseudocode konseptual untuk DMA transfer yang memberikan interrupt ketika seluruh blok selesai dipindahkan.

13.27 Proyek Mini

Judul proyek: I/O, Interrupt, dan DMA Simulator.

Mahasiswa membuat program, aplikasi sederhana, spreadsheet interaktif, atau simulator yang menunjukkan perbedaan alur transfer data.

- Menyediakan minimal satu CPU konseptual, satu memori, dan dua perangkat I/O.
- Menyediakan register status dan data untuk perangkat.
- Menyimulasikan programmed I/O dengan polling.
- Menyimulasikan interrupt-driven I/O.
- Menyimulasikan transfer blok menggunakan DMA.
- Menampilkan perubahan state CPU, perangkat, dan memori.
- Menyediakan statistik jumlah pemeriksaan, interrupt, dan transfer.
- Membandingkan penggunaan CPU pada setiap metode.

- Menyediakan minimal lima skenario pengujian.

OUTPUT YANG DIHARAPKAN

Diagram arsitektur, tabel state, alur transfer data, hasil simulasi programmed I/O, interrupt, dan DMA, serta analisis kelebihan dan keterbatasan masing-masing metode.

13.28 Daftar Pustaka

- Patterson, D. A., & Hennessy, J. L. Computer Organization and Design: The Hardware/Software Interface. Morgan Kaufmann.
- Stallings, W. Computer Organization and Architecture: Designing for Performance. Pearson.
- Hennessy, J. L., & Patterson, D. A. Computer Architecture: A Quantitative Approach. Morgan Kaufmann.
- Mano, M. M. Computer System Architecture. Pearson.
- Null, L., & Lobur, J. The Essentials of Computer Organization and Architecture. Jones & Bartlett Learning.
- Dokumen Rencana Pembelajaran Semester Mata Kuliah Arsitektur Komputer IF221101, Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jawa Timur.

CATATAN PENYUSUNAN BAB

Bab ini melanjutkan pembahasan organisasi memori dengan memperluas perhatian pada jalur komunikasi antara CPU, memori, dan perangkat eksternal. Pemahaman mengenai bus, controller I/O, interrupt, dan DMA menjadi dasar untuk menganalisis bagaimana sistem komputer mengoordinasikan transfer data dan menangani kejadian eksternal secara efisien.