

BUKU AJAR

MATA KULIAH ARSITEKTUR KOMPUTER

BAB 10

ORGANISASI CPU DAN SIKLUS INSTRUKSI

Berbasis Rencana Pembelajaran Semester (RPS)

Arsitektur Komputer IF221101 | 3 SKS | 2026

IDENTITAS BAB

Komponen	Keterangan
Mata Kuliah	Arsitektur Komputer
Kode	IF221101
Bobot	3 SKS
Bab	Bab 10
Judul	Organisasi CPU dan Siklus Instruksi
Fokus Pembelajaran	Memahami struktur internal CPU dan proses eksekusi instruksi dari pengambilan sampai penyelesaian
Level Pembelajaran	Pemahaman konsep, analisis aliran data, analisis register, dan simulasi siklus instruksi

CAPAIAN PEMBELAJARAN BAB

Setelah mempelajari bab ini, mahasiswa diharapkan mampu:

1. Menjelaskan fungsi CPU dalam sistem komputer.
2. Mengidentifikasi komponen utama CPU dan hubungan fungsionalnya.
3. Menjelaskan fungsi ALU dan Control Unit.
4. Membedakan beberapa jenis register utama dalam organisasi CPU.
5. Menjelaskan konsep datapath dan control path.
6. Menjelaskan struktur umum instruksi mesin secara konseptual.
7. Menganalisis tahapan fetch, decode, execute, dan tahapan lanjutan bila diperlukan.
8. Menjelaskan konsep micro-operation dan perpindahan data antar register.
9. Melakukan simulasi sederhana eksekusi instruksi.
10. Menganalisis hubungan antara siklus instruksi, clock, register, memori, dan jalur data.

PETA KONSEP

ALUR KONSEP BAB 10

Sistem Komputer → CPU → ALU + Control Unit + Register → Datapath + Control Path → Instruksi Mesin → Fetch → Decode → Operand Access → Execute → Write Back / Update State → Siklus Instruksi
Berikutnya.

10.1 Pendahuluan

CPU atau Central Processing Unit merupakan pusat pemrosesan yang menjalankan instruksi program. Agar sebuah program dapat dieksekusi, instruksi harus diambil dari memori, ditafsirkan, dan kemudian dijalankan melalui serangkaian perpindahan data serta operasi internal.

Pembahasan CPU tidak hanya berkaitan dengan satu komponen tunggal. CPU dapat dipahami sebagai organisasi sejumlah blok fungsional, seperti unit aritmetika dan logika, unit pengendali, register, serta jalur yang membawa data dan sinyal kendali.

PERTANYAAN PEMANTIK

Ketika sebuah program menjalankan instruksi sederhana seperti penjumlahan, langkah apa yang terjadi di dalam CPU? Dari mana instruksi diambil? Di mana instruksi disimpan sementara? Siapa yang menentukan operasi yang harus dilakukan?

10.2 Kedudukan CPU dalam Sistem Komputer

Dalam organisasi komputer, CPU berinteraksi dengan memori dan perangkat masukan/keluaran. Memori menyediakan instruksi dan data, sedangkan CPU menjalankan operasi sesuai instruksi. Perangkat I/O memungkinkan pertukaran informasi antara sistem komputer dan lingkungan luar.

Komponen	Peran Utama
CPU	Menjalankan instruksi dan mengendalikan pemrosesan
Memori	Menyimpan instruksi dan data
Input	Menyediakan data atau perintah dari lingkungan
Output	Menyajikan hasil pemrosesan
Interkoneksi	Menyediakan jalur pertukaran data, alamat, dan sinyal kendali

Interaksi tersebut merupakan dasar untuk memahami bagaimana program yang tersimpan dapat dieksekusi secara berurutan atau melalui perubahan aliran kontrol.

10.3 Struktur Utama CPU

Secara konseptual, CPU terdiri dari beberapa bagian yang bekerja sama. Pembagian rinci dapat berbeda antar arsitektur, tetapi komponen berikut membantu memahami organisasi dasar CPU.

Komponen	Fungsi Konseptual
ALU	Melakukan operasi aritmetika dan logika
Control Unit	Menghasilkan dan mengoordinasikan sinyal kendali
Register	Menyimpan data, alamat, instruksi, atau keadaan secara sementara
Datapath	Jalur dan unit yang digunakan untuk perpindahan serta pemrosesan data
Control Path	Jalur logis yang mengatur kapan dan bagaimana operasi dilakukan

10.4 Arithmetic Logic Unit (ALU)

ALU adalah bagian CPU yang melakukan operasi aritmetika dan logika. Contoh operasi meliputi penjumlahan, pengurangan, AND, OR, XOR, pergeseran, dan perbandingan, tergantung rancangan prosesor.

ALU menerima operand, menerima informasi mengenai jenis operasi yang harus dilakukan, kemudian menghasilkan keluaran dan dalam banyak rancangan memperbarui informasi status.

Kelompok Operasi	Contoh
Aritmetika	Tambah, kurang, increment
Logika	AND, OR, XOR, NOT
Perbandingan	Sama dengan, lebih besar, lebih kecil
Pergeseran	Shift kiri, shift kanan sesuai rancangan

HUBUNGAN DENGAN BAB SEBELUMNYA

ALU dibangun berdasarkan prinsip rangkaian kombinasional, penjumlah biner, representasi integer, dan operasi logika yang telah dipelajari pada bab sebelumnya.

10.5 Control Unit

Control Unit mengoordinasikan urutan operasi dalam CPU. Unit ini menentukan kapan register membaca atau menulis data, kapan ALU melakukan operasi tertentu, dan bagaimana proses eksekusi instruksi bergerak dari satu tahap ke tahap berikutnya.

Secara konseptual, Control Unit dapat dipandang sebagai bagian yang menghasilkan sinyal kendali berdasarkan instruksi, keadaan sistem, dan tahapan eksekusi.

Contoh Aktivitas Kendali	Tujuan
Load register	Memindahkan data ke register
Select ALU operation	Menentukan fungsi ALU
Read memory	Mengambil instruksi atau data
Write memory	Menyimpan data ke memori
Update PC	Mengubah alamat instruksi berikutnya

10.6 Register dalam CPU

Register adalah penyimpanan berkecepatan tinggi yang digunakan untuk menahan informasi yang sedang dibutuhkan CPU. Fungsi dan nama register dapat berbeda antar arsitektur.

Register	Fungsi Konseptual
Program Counter / PC	Menyimpan alamat instruksi berikutnya
Instruction Register / IR	Menyimpan instruksi yang sedang diproses

Memory Address Register / MAR	Menyimpan alamat lokasi memori yang akan diakses
Memory Data Register / MDR atau MBR	Menyimpan data yang dibaca dari atau akan ditulis ke memori
Accumulator	Menyimpan hasil antara pada organisasi tertentu
General Purpose Register	Menyimpan operand dan hasil sementara
Status/Flag Register	Menyimpan informasi kondisi hasil operasi

Nama dan implementasi register dapat berbeda, tetapi pemahaman fungsi konseptualnya penting untuk menganalisis siklus instruksi.

10.7 Program Counter dan Instruction Register

10.7.1 Program Counter

Program Counter menyimpan alamat instruksi yang secara konseptual akan diambil berikutnya. Setelah pengambilan instruksi, nilai PC dapat diperbarui sesuai panjang instruksi dan aliran kontrol yang digunakan oleh arsitektur.

10.7.2 Instruction Register

Instruction Register menyimpan instruksi yang sedang didekode dan dieksekusi. Penyimpanan sementara ini memungkinkan bagian pengendali menganalisis informasi operasi tanpa harus terus-menerus mengambil instruksi yang sama dari memori.

10.8 MAR dan MDR/MBR

Akses memori secara konseptual dapat dijelaskan melalui dua jenis informasi: alamat lokasi dan data yang dipertukarkan. MAR menyimpan alamat, sedangkan MDR atau MBR digunakan untuk menyimpan data yang sedang ditransfer.

CONTOH KONSEPTUAL FETCH

PC → MAR; Memori dibaca; data instruksi dari memori → MDR; MDR → IR; kemudian PC diperbarui sesuai mekanisme arsitektur.

10.9 Datapath

Datapath adalah kumpulan jalur dan unit yang memungkinkan data berpindah serta diproses. Datapath dapat mencakup register, ALU, bus internal, multiplexer, dan koneksi ke antarmuka memori.

Ketika sebuah instruksi meminta operasi penjumlahan, datapath menyediakan jalur agar operand dapat dipilih dari register atau sumber lain, diteruskan ke ALU, dan hasilnya disimpan kembali ke lokasi yang sesuai.

10.10 Control Path

Control path menghasilkan sinyal yang mengatur operasi datapath. Jika datapath menjawab pertanyaan “data berpindah ke mana?”, control path membantu menjawab “kapan perpindahan terjadi dan operasi apa yang dilakukan?”.

Sinyal Kendali Konseptual	Efek
Register Write	Mengizinkan penulisan ke register
ALU Control	Memilih operasi ALU
Memory Read	Meminta pembacaan memori
Memory Write	Meminta penulisan memori
PC Control	Menentukan mekanisme pembaruan PC

10.11 Instruksi Mesin

Program tingkat tinggi pada akhirnya diterjemahkan menjadi instruksi yang dapat dikenali oleh arsitektur prosesor. Secara umum, sebuah instruksi dapat memuat informasi mengenai operasi yang harus dilakukan dan informasi operand.

Bagian Instruksi	Makna Umum
Opcode	Menentukan jenis operasi
Operand/Address Field	Menentukan data, register, atau alamat yang digunakan
Mode atau Field Tambahan	Memberikan informasi tambahan sesuai format arsitektur

Format instruksi berbeda antar arsitektur. Oleh karena itu, pembahasan dalam bab ini menggunakan model konseptual untuk menjelaskan aliran eksekusi.

10.12 Siklus Instruksi

Siklus instruksi adalah urutan aktivitas yang dilakukan CPU untuk mengambil dan menjalankan instruksi. Model dasar sering dijelaskan melalui tahap fetch, decode, dan execute. Dalam implementasi tertentu, tahap tersebut dapat diperinci lebih jauh.

Tahap	Aktivitas Utama
Fetch	Mengambil instruksi dari memori
Decode	Menafsirkan opcode dan informasi instruksi
Operand Access	Mengambil operand bila diperlukan
Execute	Melakukan operasi
Write Back	Menyimpan hasil bila diperlukan
Next Instruction	Menentukan instruksi berikutnya

10.13 Tahap Fetch

Pada tahap fetch, CPU menggunakan alamat instruksi yang tersedia untuk melakukan akses ke memori. Instruksi kemudian dipindahkan ke register instruksi.

Urutan mikro-operasi konseptual dapat ditulis sebagai:

- $MAR \leftarrow PC$
- $MDR \leftarrow \text{Memory}[MAR]$
- $IR \leftarrow MDR$
- $PC \leftarrow \text{alamat instruksi berikutnya sesuai mekanisme arsitektur}$

Notasi panah menunjukkan perpindahan atau penyalinan nilai secara konseptual. Implementasi nyata dapat memiliki organisasi internal yang berbeda.

10.14 Tahap Decode

Pada tahap decode, bagian pengendali menganalisis isi IR. Opcode digunakan untuk menentukan operasi, sedangkan field lain dapat menunjukkan register, operand, alamat, atau mode pengalamatan.

Hasil decode menjadi dasar pembentukan sinyal kendali untuk tahap berikutnya.

10.15 Operand Access

Tidak semua instruksi membutuhkan akses operand dengan cara yang sama. Operand dapat berasal dari register, nilai yang terdapat dalam instruksi, atau memori sesuai mekanisme arsitektur.

Tahap operand access membantu memisahkan proses pengambilan instruksi dari proses memperoleh data yang akan digunakan dalam operasi.

10.16 Tahap Execute

Tahap execute melakukan operasi yang telah ditentukan. Untuk instruksi aritmetika, operand dapat dikirim ke ALU. Untuk instruksi logika, ALU menjalankan fungsi logika. Untuk instruksi kontrol, tahap ini dapat mengubah aliran eksekusi.

CONTOH KONSEPTUAL

Jika instruksi adalah `ADD R1, R2`, maka CPU secara konseptual mengambil isi R1 dan R2, memilih operasi penjumlahan pada ALU, lalu menempatkan hasil ke tujuan yang ditentukan oleh format instruksi.

10.17 Write Back dan Pembaruan State

Setelah operasi selesai, hasil dapat ditulis ke register, memori, atau digunakan untuk memperbarui state internal. Beberapa instruksi tidak menghasilkan nilai yang harus ditulis kembali, tetapi dapat mengubah PC atau status sistem.

Pembaruan state menunjukkan bahwa eksekusi instruksi dapat memengaruhi kondisi yang akan digunakan oleh instruksi berikutnya.

10.18 Micro-operation

Micro-operation adalah operasi dasar yang dilakukan terhadap data yang disimpan dalam register atau bagian internal CPU. Contoh sederhana adalah transfer register, operasi aritmetika, operasi logika, dan pergeseran.

Jenis	Contoh Notasi Konseptual
Transfer	$R2 \leftarrow R1$
Aritmetika	$R3 \leftarrow R1 + R2$
Logika	$R3 \leftarrow R1 \text{ AND } R2$
Shift	$R1 \leftarrow \text{shift}(R1)$

Urutan micro-operation membantu menjelaskan bagaimana satu instruksi tingkat mesin dapat dipecah menjadi langkah-langkah internal.

10.19 Clock dan Urutan Operasi CPU

Pada CPU sinkron, clock menyediakan acuan waktu untuk koordinasi perubahan keadaan. Perpindahan data dan pembaruan register dilakukan sesuai rancangan waktu sistem.

Satu instruksi tidak selalu identik dengan satu siklus clock. Jumlah tahap dan cara pelaksanaan bergantung pada organisasi CPU, desain kontrol, dan pendekatan eksekusi yang digunakan.

10.20 Flag dan Status Register

Operasi ALU dapat menghasilkan informasi kondisi yang disimpan dalam status register. Contoh flag yang umum secara konseptual mencakup zero, carry, negative/sign, dan overflow, meskipun detailnya berbeda antar arsitektur.

Flag Konseptual	Makna Umum
Zero	Hasil memenuhi kondisi nol
Carry	Terjadi carry pada operasi tertentu
Negative/Sign	Menunjukkan sifat tanda hasil dalam representasi tertentu
Overflow	Hasil signed berada di luar rentang representasi

Flag dapat digunakan oleh instruksi berikutnya, terutama instruksi percabangan bersyarat.

10.21 Aliran Kontrol dan Percabangan

Tidak semua instruksi menyebabkan PC bergerak secara linear. Instruksi branch atau jump dapat mengubah alamat instruksi berikutnya. Percabangan bersyarat menggunakan informasi kondisi untuk menentukan apakah perpindahan aliran kontrol dilakukan.

Konsep ini memungkinkan program melakukan pengulangan, pengambilan keputusan, dan struktur kontrol lainnya.

10.22 Studi Kasus 1: Simulasi Fetch Instruksi

Misalkan PC berisi alamat 100 dan memori pada alamat tersebut menyimpan instruksi. Proses konseptual dapat digambarkan sebagai berikut:

Langkah	Keadaan
1	PC berisi alamat 100
2	Nilai PC dipindahkan ke MAR
3	Memori dibaca menggunakan alamat pada MAR
4	Instruksi diterima ke MDR
5	Isi MDR dipindahkan ke IR
6	PC diperbarui untuk menentukan alamat instruksi berikutnya

Mahasiswa dapat mensimulasikan perubahan nilai register menggunakan tabel keadaan sebelum dan sesudah setiap micro-operation.

10.23 Studi Kasus 2: Instruksi Penjumlahan

Pertimbangkan instruksi konseptual ADD R1, R2. Asumsikan isi R1=5 dan R2=7. Setelah decode menentukan operasi ADD, datapath memilih kedua register sebagai operand ALU. ALU menghasilkan 12, kemudian hasil ditulis ke lokasi tujuan sesuai format instruksi.

ANALISIS

Mahasiswa diminta menuliskan urutan register transfer yang mungkin terjadi dan menjelaskan sinyal kendali konseptual yang diperlukan.

10.24 Studi Kasus 3: Percabangan Bersyarat

Sebuah instruksi branch dapat menggunakan flag zero. Jika hasil operasi sebelumnya bernilai nol, alamat target dimuat sebagai alamat instruksi berikutnya. Jika tidak, PC melanjutkan ke instruksi berikutnya sesuai urutan normal.

Kasus ini memperlihatkan hubungan antara ALU, status register, Control Unit, dan Program Counter.

10.25 Aktivitas Pembelajaran Mahasiswa

Aktivitas 1: Peta CPU

Buat diagram konseptual hubungan ALU, Control Unit, register, memori, dan I/O.

Aktivitas 2: Register Transfer

Tuliskan urutan transfer register untuk proses fetch sederhana.

Aktivitas 3: Siklus Instruksi

Simulasikan fetch–decode–execute untuk instruksi penjumlahan.

Aktivitas 4: Datapath

Tentukan jalur yang dilalui operand dari register menuju ALU dan kembali ke register tujuan.

Aktivitas 5: Control Signal

Identifikasi sinyal kendali konseptual yang diperlukan untuk membaca memori dan menulis register.

Aktivitas 6: Branch

Buat simulasi perubahan PC untuk percabangan bersyarat berdasarkan nilai zero flag.

Aktivitas 7: Analisis

Bandingkan peran PC, IR, MAR, dan MDR dalam eksekusi instruksi.

10.26 Rangkuman

- CPU merupakan pusat pemrosesan yang menjalankan instruksi program.
- Komponen konseptual utama CPU meliputi ALU, Control Unit, register, datapath, dan control path.
- ALU melakukan operasi aritmetika dan logika.
- Control Unit mengoordinasikan urutan operasi melalui sinyal kendali.
- Register menyimpan data, instruksi, alamat, dan state secara sementara.
- PC menyimpan informasi alamat instruksi berikutnya, sedangkan IR menyimpan instruksi yang sedang diproses.
- MAR dan MDR/MBR membantu menjelaskan akses alamat dan pertukaran data dengan memori.
- Datapath menyediakan jalur perpindahan dan pemrosesan data.
- Control path mengatur operasi datapath.
- Siklus instruksi dapat dijelaskan melalui fetch, decode, operand access, execute, dan write back.
- Micro-operation memecah eksekusi instruksi menjadi langkah internal yang lebih kecil.
- Clock membantu mengoordinasikan perubahan state pada CPU sinkron.
- Flag dan status register menyediakan informasi kondisi untuk operasi dan percabangan.

10.27 Glosarium Bab

Istilah	Pengertian
ALU	Unit yang melakukan operasi aritmetika dan logika.
Branch	Perubahan aliran instruksi berdasarkan alamat atau kondisi tertentu.
Clock	Sinyal acuan waktu untuk sinkronisasi operasi sistem.
Control Path	Bagian konseptual yang menghasilkan sinyal pengendali.
Control Unit	Unit yang mengatur urutan pelaksanaan operasi CPU.
Datapath	Jalur dan unit yang digunakan untuk perpindahan serta pemrosesan data.

Decode	Proses menafsirkan informasi instruksi.
Execute	Tahap menjalankan operasi yang ditentukan instruksi.
Fetch	Tahap mengambil instruksi dari memori.
Flag	Informasi status hasil operasi.
Instruction Register	Register penyimpan instruksi yang sedang diproses.
MAR	Register konseptual penyimpan alamat memori yang diakses.
MDR/MBR	Register konseptual penyimpan data yang ditransfer dari atau ke memori.
Micro-operation	Operasi dasar terhadap register atau elemen internal CPU.
Opcode	Bagian instruksi yang menunjukkan jenis operasi.
Program Counter	Register penyimpan alamat instruksi berikutnya.
Write Back	Tahap penyimpanan hasil operasi ke tujuan yang sesuai.

10.28 Soal Latihan

A. Soal Pemahaman

11. Jelaskan fungsi utama CPU dalam sistem komputer.
12. Bandingkan fungsi ALU dan Control Unit.
13. Jelaskan perbedaan konseptual antara datapath dan control path.
14. Apa fungsi Program Counter?
15. Apa fungsi Instruction Register?
16. Jelaskan peran MAR dan MDR dalam akses memori.
17. Jelaskan tahapan utama dalam siklus instruksi.
18. Apa yang dimaksud micro-operation?
19. Jelaskan hubungan antara clock dan pembaruan state CPU.
20. Mengapa status register penting untuk instruksi percabangan?

B. Soal Analisis

21. Tuliskan urutan micro-operation konseptual untuk mengambil instruksi dari memori ke IR.
22. Buat tabel perubahan register untuk proses fetch ketika PC mula-mula bernilai 200.
23. Analisis jalur data untuk instruksi konseptual ADD R1, R2.
24. Jika R1=15 dan R2=9, jelaskan perubahan state yang terjadi ketika hasil penjumlahan ditulis ke R1.
25. Buat urutan konseptual eksekusi instruksi LOAD dari memori ke register.
26. Analisis bagaimana PC berubah ketika instruksi branch bersyarat dijalankan dengan kondisi benar dan kondisi salah.

C. Soal HOTS dan Studi Kasus

27. Rancang model datapath sederhana yang memiliki dua register sumber, satu ALU, dan satu register tujuan. Jelaskan jalur data dan sinyal kontrolnya.
28. Analisis akibat yang mungkin terjadi jika Program Counter tidak diperbarui secara benar.

29. Jelaskan bagaimana perubahan jumlah register dapat memengaruhi kebutuhan instruksi dan perpindahan data.
30. Bandingkan secara konseptual proses eksekusi instruksi aritmetika, instruksi load, dan instruksi branch.
31. Buat simulasi state CPU untuk menjalankan tiga instruksi sederhana secara berurutan dan identifikasi perubahan register pada setiap tahap.

10.29 Proyek Mini

Judul proyek: CPU Instruction Cycle Simulator.

Mahasiswa membuat simulasi, program, spreadsheet interaktif, atau visualisasi yang menunjukkan proses kerja CPU sederhana.

- Menyediakan register minimal PC, IR, dua general purpose register, dan satu register hasil.
- Menyediakan model memori sederhana untuk instruksi dan data.
- Menyimulasikan tahap fetch.
- Menyimulasikan tahap decode.
- Menyimulasikan minimal tiga instruksi: LOAD, ADD, dan STORE atau operasi setara.
- Menampilkan perubahan register pada setiap langkah.
- Menampilkan sinyal kontrol atau aktivitas datapath secara konseptual.
- Menyediakan minimal lima skenario pengujian.

OUTPUT YANG DIHARAPKAN

Dokumentasi arsitektur simulator, tabel state register, alur fetch–decode–execute, hasil pengujian, serta analisis hubungan antara register, ALU, Control Unit, dan memori.

10.30 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.
- Mano, M. M. Computer System Architecture. Pearson.
- Harris, D. M., & Harris, S. L. Digital Design and Computer Architecture. Morgan Kaufmann.
- 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 menjadi penghubung antara konsep rangkaian digital dan pembahasan organisasi prosesor. Setelah memahami bagaimana data direpresentasikan dan bagaimana rangkaian menyimpan serta memproses state, mahasiswa dapat melihat bagaimana CPU menggabungkan register, ALU, kendali, dan akses memori untuk menjalankan instruksi program.