Set Instruksi & Mode Pengalamatan

Karakteristik Instruksi Mesin

- Set intruksi adalah kumpulan lengkap dari instruksi yang dapat dieksekusi oleh CPU
- Set instruksi adalah interface antara perancang komputer dan programmer

Element dari Instruksi

- Kode Operasi (Op code)
- Referensi Operand Sumber
- Referensi Operand Hasil
- Referensi Instruksi Selanjutnya

Elemen Instruksi (Siklus)

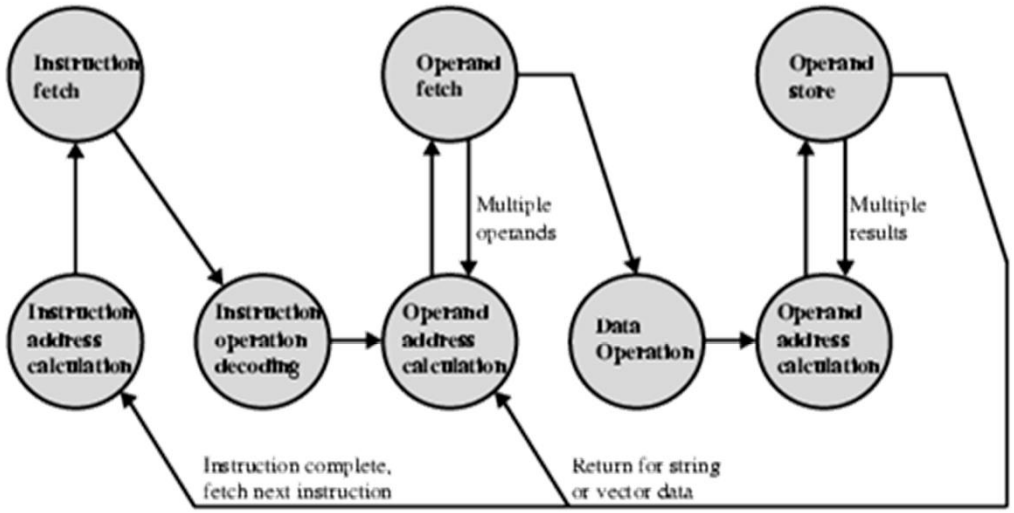


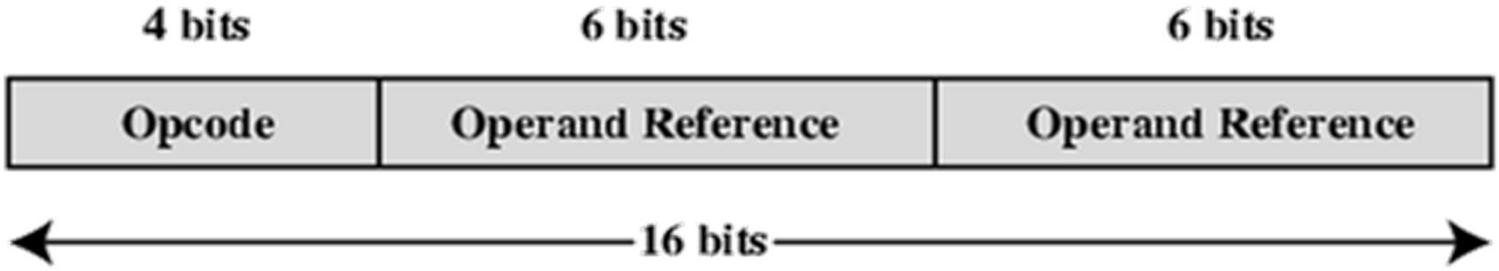
Figure 10.1 Instruction Cycle State Diagram

Operand di simpan di:

- Memori Utama (atau memori virtual atau cache)
- Register CPU
- Immediate
  - Operand berada pada instruksi yang sedang di eksekusi
- Divais I/O

Representasi Instruksi

- Dalam bahasa mesin setiap intruksi mempunyai pola kombinasi bit yang unik
- Representasi Simbolik digunakan untuk memudahkan
  - contoh. ADD, SUB, LOAD
- Operand dapat juga direpresentasikan sebagai:
  - ADD A,B



Jenis-jenis Instruksi

- Pengolahan data
- Penyimpanan data (memori utama)
- Perpidahan data (I/O)
- Control

Jumlah Alamat (a)

- 3 alamat
  - Operand 1, Operand 2, hasil
  - Tidak umum digunakan
  - Memerlukan word yang sangat panjang

  - Contoh : c = a + b
    - ADD c, a, b (c ← a + b)

Jumlah Alamat (b)

- 2 alamat
  - Alamat berfungsi sebagai operand dan hasil
  - Instuksi lebih pendek
  - Memerlukan bantuan
    - Penyimpanan sementara untuk menyimpan hasil

  - Contoh c = a + b
    - ADD a, b (a ← a + b)
    - MOV c, a (c ← a)

Jumlah Alamat (c)

- 1 alamat
  - Alamat kedua implisit
  - Biasanya register (accumulator)
  - Umum untuk mesin-mesin awal

  - Contoh c = a + b
    - LOAD a (AC ← a)
    - ADD b (AC ← AC + b)
    - STORE c (c ← AC)

Perbandingan Jumlah Operand

- KASUS : C = A + B

- ADD C,A,B

- ADD A,B
- MOV C,A

- LOAD A
- ADD B
- STOR C

Contoh Persamaan

- $Y = \frac{A-B}{[C+(DxE)]}$
- SK-37-03
  - $Y = \frac{A\ x\ C}{\left(\frac{D}{E}\right)-G} + F$

SK-37-04

$$x = \frac{b + b^2 - 4ac}{2a}$$

Jumlah Alamat (d)

- 0 (zero) alamat
  - Semua alamat implisit
  - Menggunakan stack

  - contoh c = a + b
    - push a
    - push b
    - add
    - pop c

Karaktersitik Alamat

- Alamat lebih banyak
  - Instruksi lebih komplek
  - Lebih banyak register
    - Operasi antar register lebih cepat
  - Lebih sedikit instruksi per program
- Alamat lebih sedikit
  - Instruksi lebih sederhana
  - Lebih banyak instruksi per program
  - Fetch dan eksekusi lebih cepat

Perancangan Set Instruksi

- Keperluan operasi
  - Berapa banyak operasi?
  - Operasi apa yang akan disedikan?
  - Seberapa komplek?
- Jenis Data
- Format Instruksi
  - Panjang dari field op code
  - Banyaknya alamat
- Register
  - Jumlah register CPU yang tersedia
  - Operasi apa yang bisa dilakukan di register tertentu ?
- Pengalamatan

Jenis dari Operand

- Alamat
  - Dapat dianggap sebagai unsigned integer
- Angka
  - Integer/floating point/desimal
- Karakter
  - IRA (International Reference Alphabet)
  - ASCII (American Standard Code for Information Interchange)
- Data Logik
  - Bit-oriented

Jenis Operation

- Transfer Data
- Arithmetik
- Logik
- Konversi
- I/O
- Kendali Sistem
- Pengalihan Kendali (*Transfer of Control*)

Transfer Data

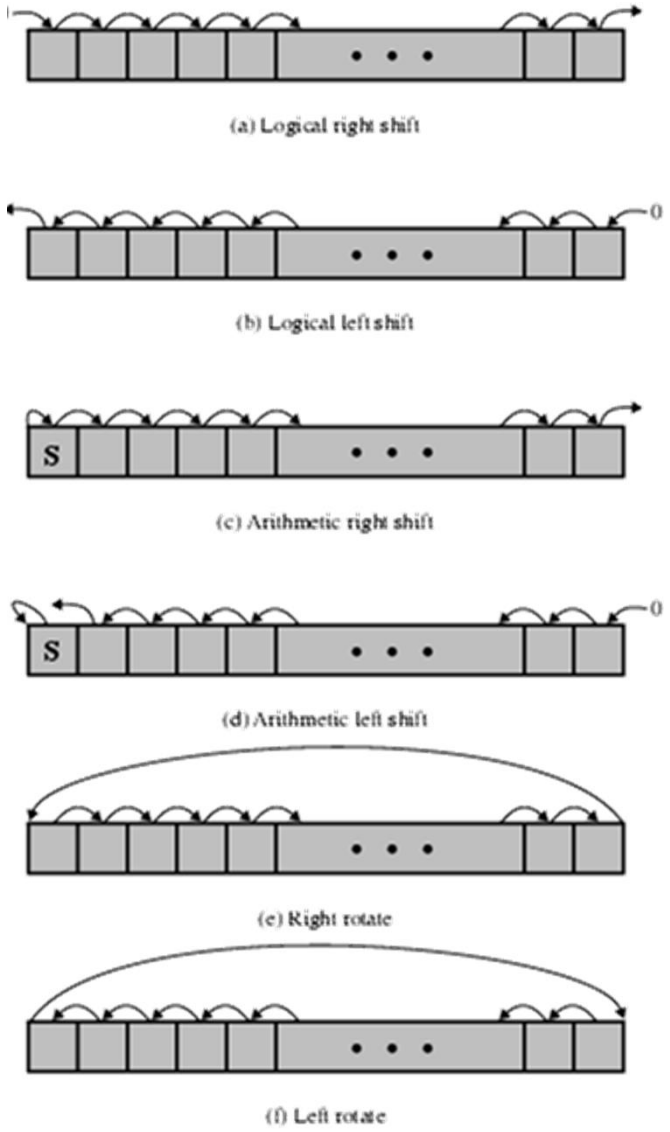
- Tentukan
  - Sumber dan tujuan (memory, register, atau stack)
  - Besarnya data
  - Mode pengalamatan
- Kegiatan CPU :
  - Menghitung address, periksa jika alamat berada di memori virtual
  - Periksa apakah sudah ada di cache
  - Memberi perintah ke memori

Arithmetik

- Add, Subtract, Multiply, Divide
  - Untuk signed integer, floating point, packed decimal
- Instruksi Operand Tunggal:
  - Increment
  - Decrement
  - Negate
  - Absolute
- Dapat melibatkan operasi transfer data

Logical

- Operasi level bit
- AND, OR, NOT, XOR
- Shift dan rotate
- Dapat melibatkan operasi transfer data



Konversi

- Mengubah format data
- Contoh biner ke desimal atau sebaliknya

Input/Output

- Dapat merupakan instruksi khusus
- Dapat dikerjakan dengan instruksi pemindah data (memory mapped)
- Dapat dikerjakan dengan pengendali khusus (DMA)

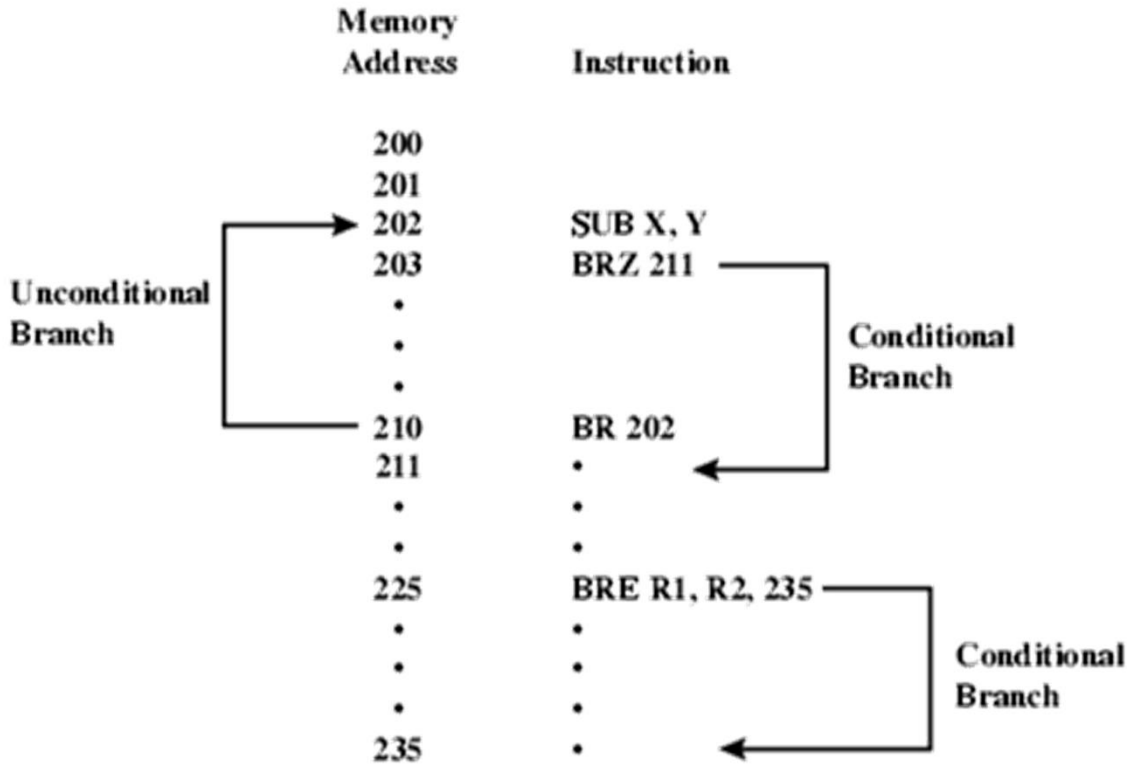
Kendali Sistem

- Instruksi khusus
- CPU harus ada pada state tertentu
- Disediakan untuk sistem operasi

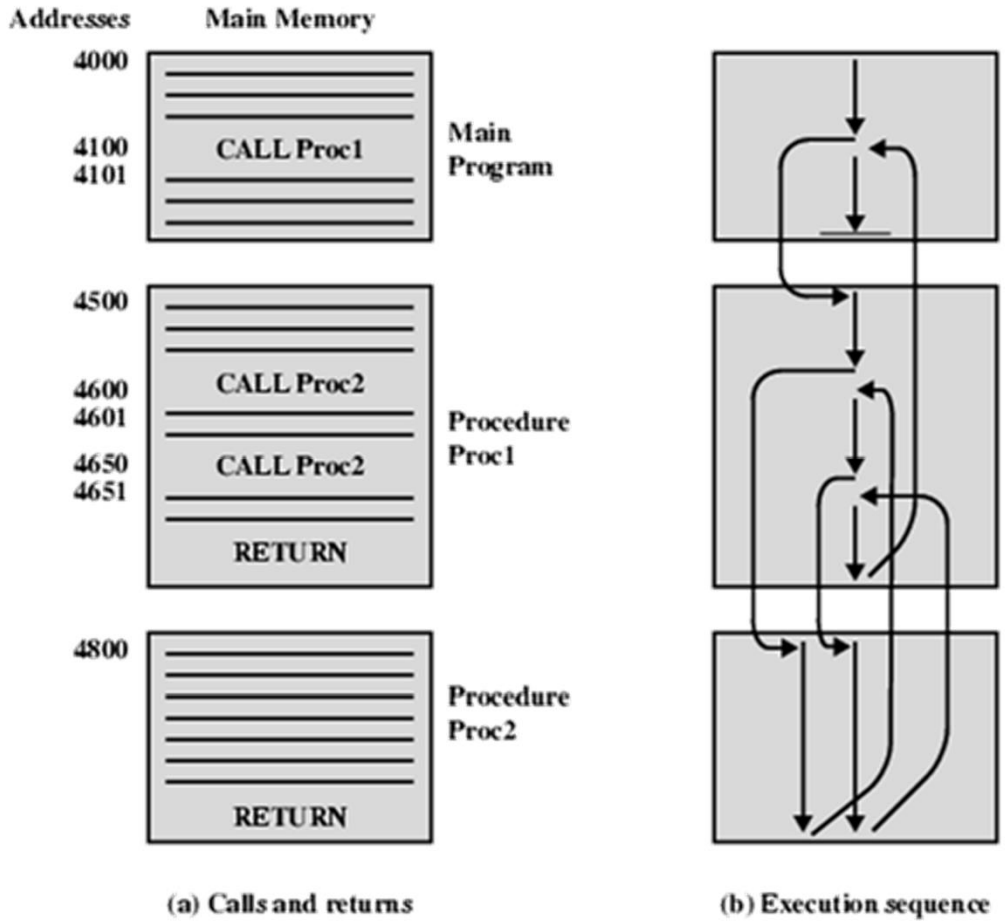
Pengalihan Kendali

- Pencabangan (*Branch*)
  - Loncat ke intruksi di lokasi lain selain instruksi berikutnya
- Skip
  - Skip instruksi selanjutnya
- Procedure call
  - Memanggil program lain

Instruksi Pencabangan (Branch)

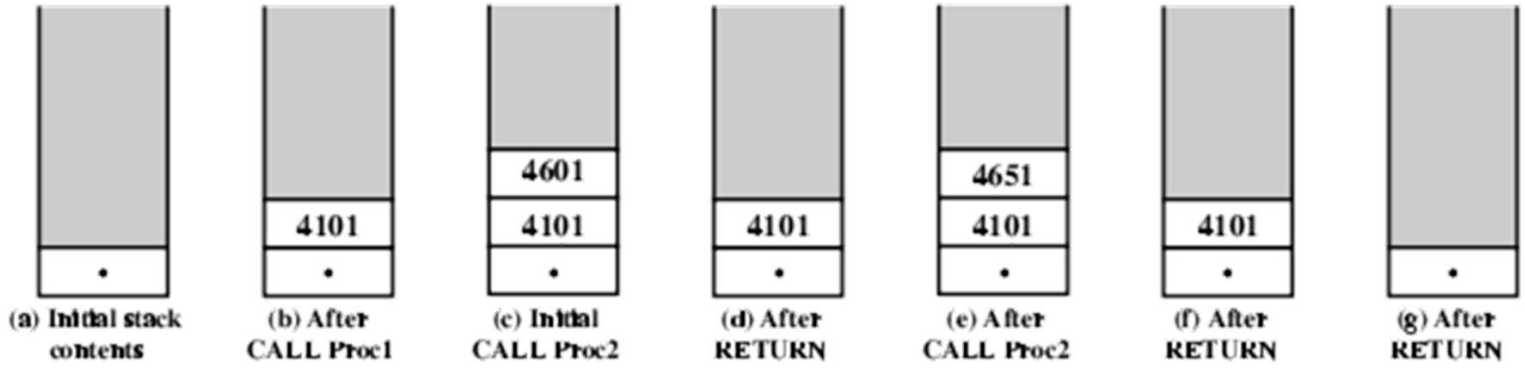


Nested Procedure Calls



Alamat Kembali dari suatu Procedure Call

- Register
- Permulaan dari prosedur yang di panggil (*called procedure)*
- Top dari stack



Mode Pengalamatan

- Umumnya terdapat beberapa mode pengalamatan
  - Setiap mode menggunakan opcode yang berbeda
  - Field mode di intruksi menunjukkan mode pengalamatan

Teknik pengalamatan umum:
- Immediate
- Langsung (Direct)
- Tak langsung (Indirect)
- Register
- Register Indirect
- Displacement (Indexed)
- Stack

Pengalamatan Immediate

- Operand merupakan bagian dari instruksi, sehingga tidak bisa mengganti operand tanpa mengubah instruksi
- Misal ADD 5 (INTEL : ADD AX,5)
  - Tambahkan 5 ke isi accumulator
  - 5 adalah operand
- Tidak ada pengambilan data dari memori,
- Cepat

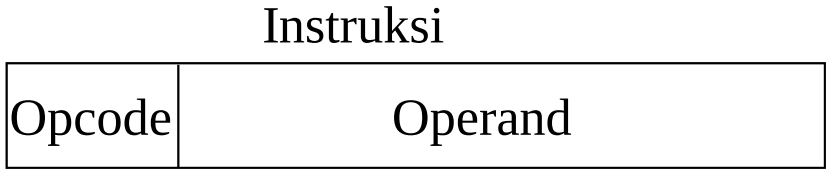
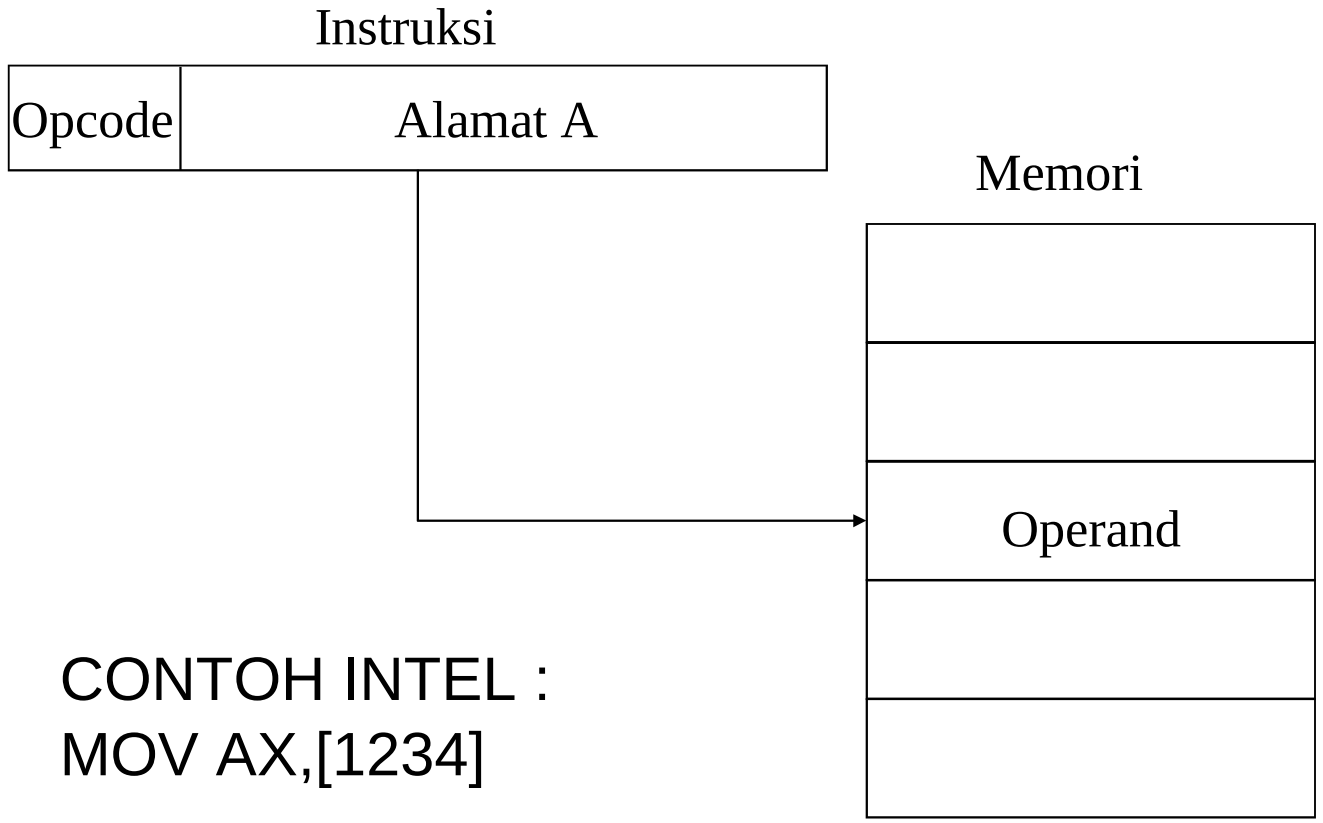


Diagram Pengalamatan Langsung

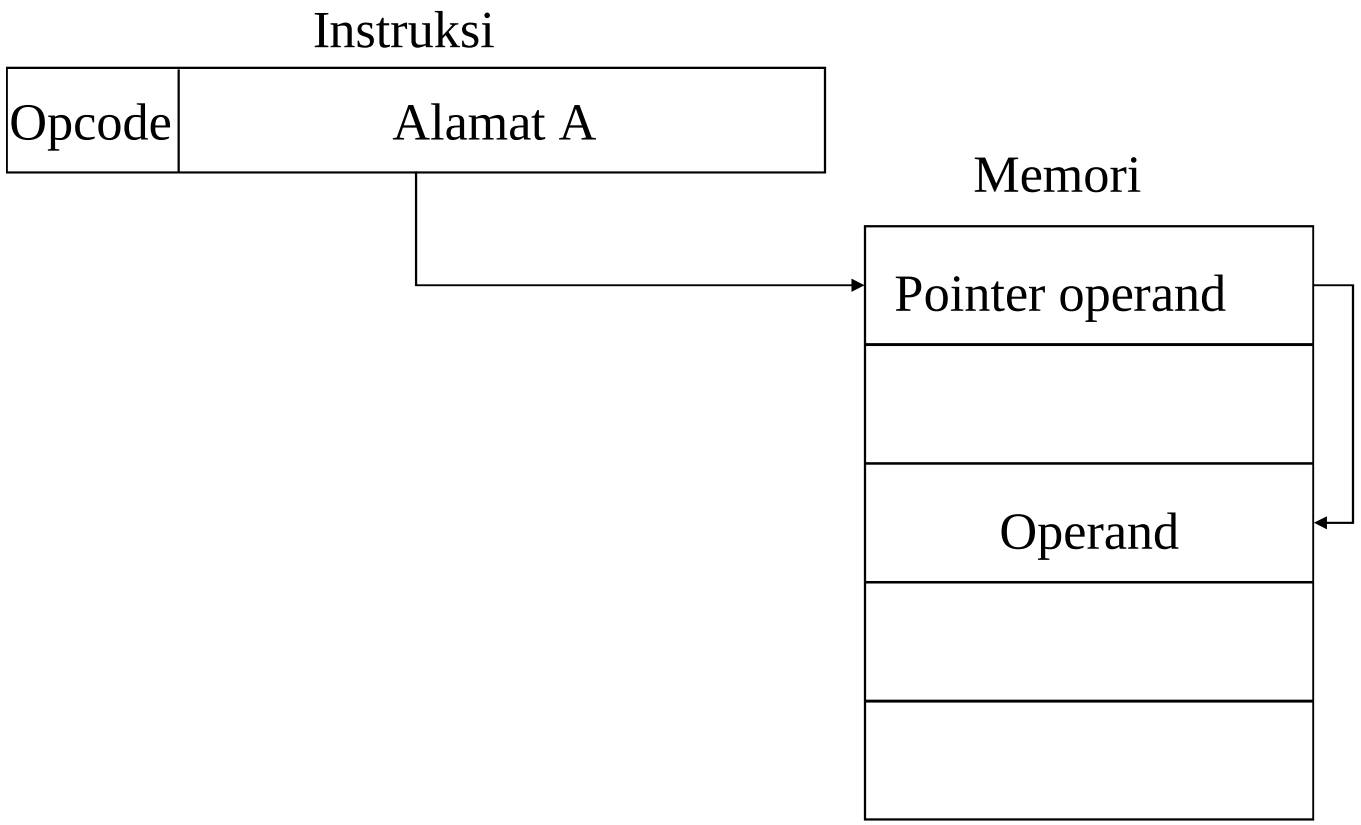


CONTOH INTEL :
MOV AX,[1234]

Pengalamatan Langsung

- Field alamat pada instruksi berisi alamat dari operand
- Alamat effektif (EA) = field alamat (A)
- Misal ADD A
  - Jumlahkan **isi** memori pada alamat A ke accumulator
- Referensi memori tunggal untuk mengakses data
- Tidak perlu ada perhitungan alamat
- Ruang alamat memori terbatas tidak bisa melebihi field alamat

Diagram Pengalamatan Tak Langsung



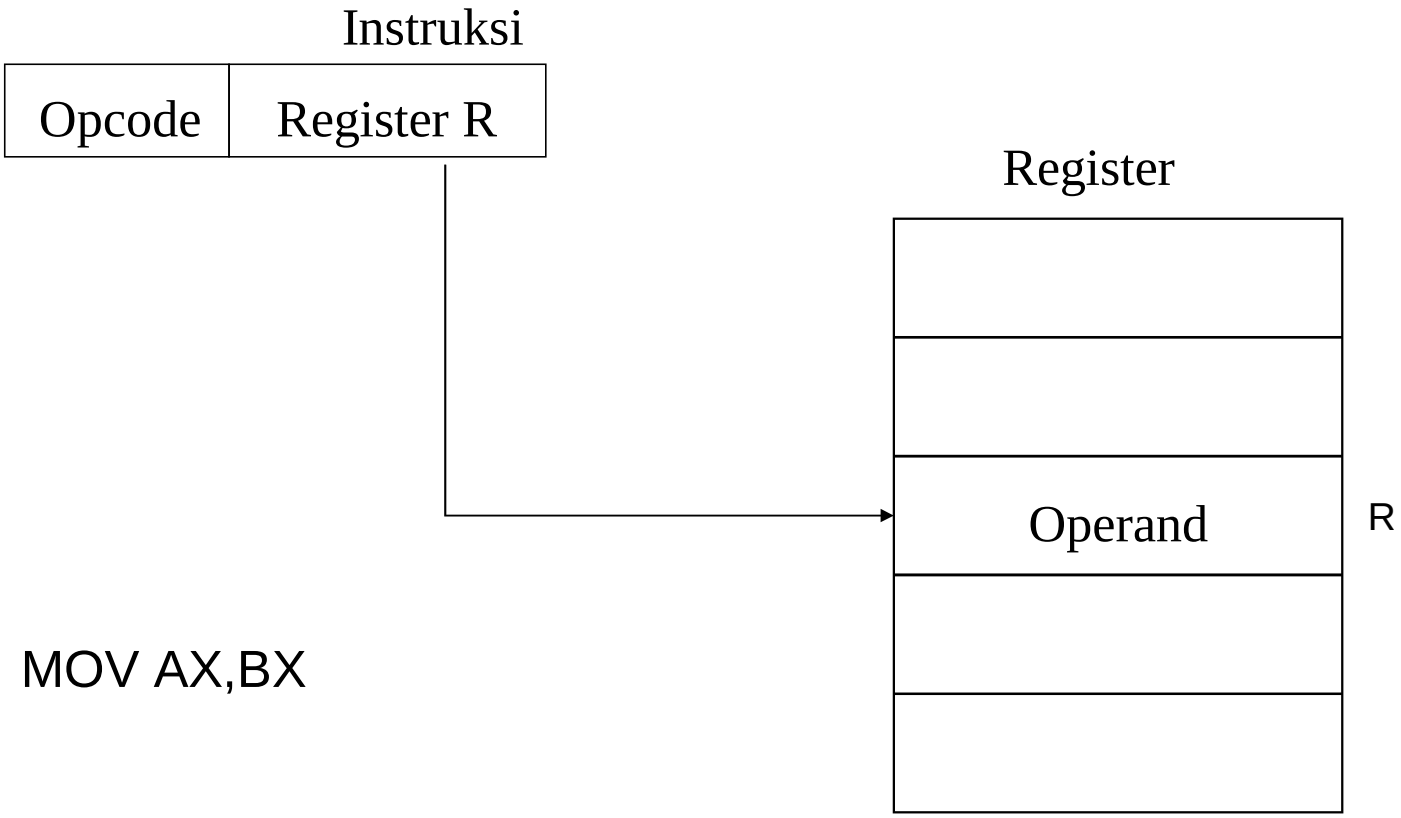
Pengalamatan Tak Langsung (1)

- Lokasi memori ditunjukkan oleh field alamat yang mengandung alamat dari operand
- EA = (A)
  - Operand terletak di memori dengan alamat yang terletak di A
- Contoh ADD (A)
  - Jumlahkan isi memori yang ditunjukkan oleh isi dari alamat A ke accumulator

Pengalamatan Tak Langsung(2)

- Ruang alamat besar
- Minimal $2^n * 2^m$;dengan n = field alamat di instruksi dan m = lebar memori
- Bisa nested, multilevel, cascade
  - contoh EA = (((A)))
- Pengaksesan memori berkali-kali untuk mendapatkan operand
- Lebih lambat dari pengalamatan langsung

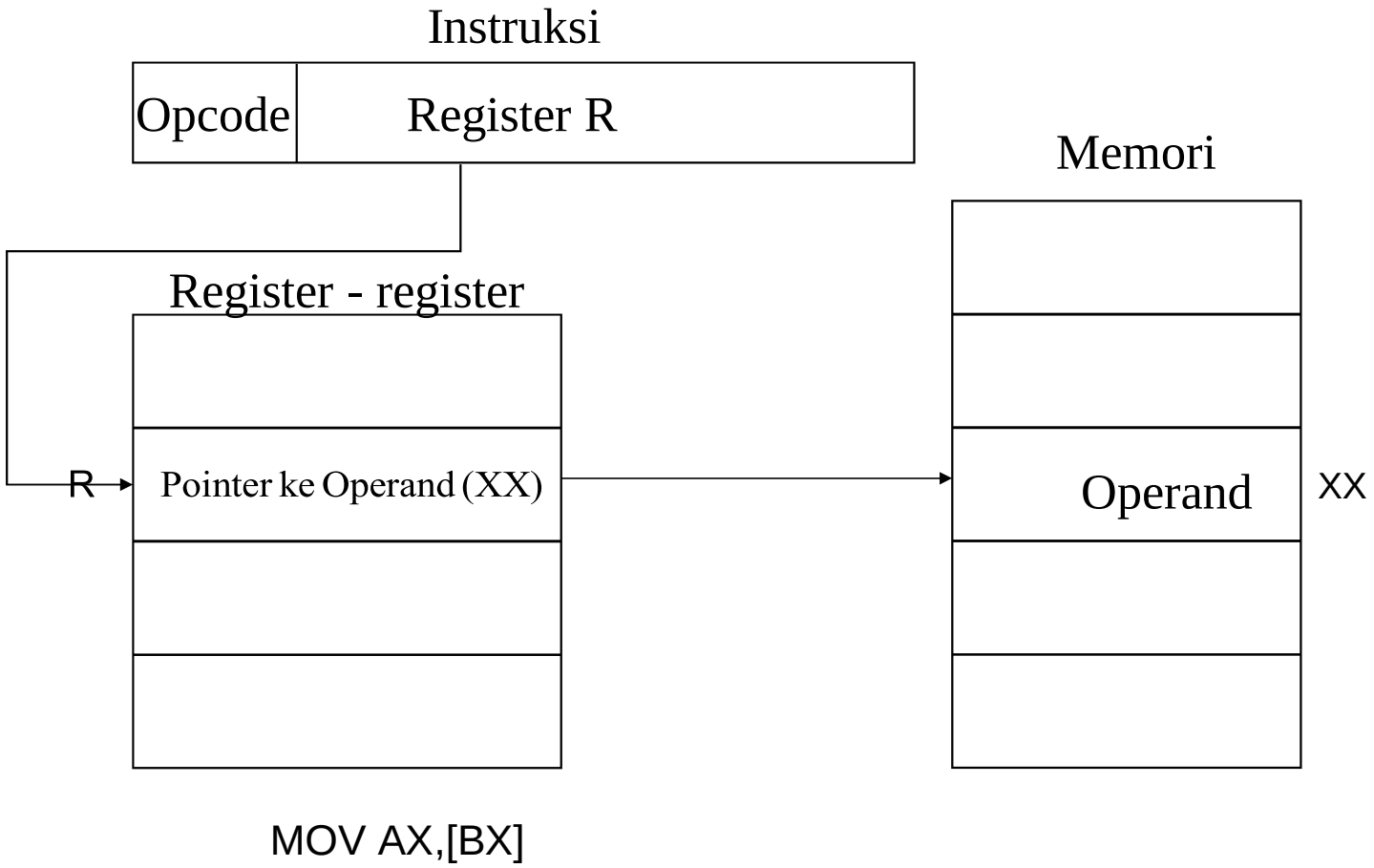
Diagram Pengalamatan Register



Pengalamatan Register (1)

- Operand ada di register
- EA = R
- Jumlah register terbatas
- Field alamat sangat kecil (jumlah bit pengkodean register lebih sedikit)
  - Instruksi lebih ringkas
  - Fetch lebih cepat
- Tanpa akses memori
- Eksekusi sangat cepat
- Ruang alamat sangat terbatas

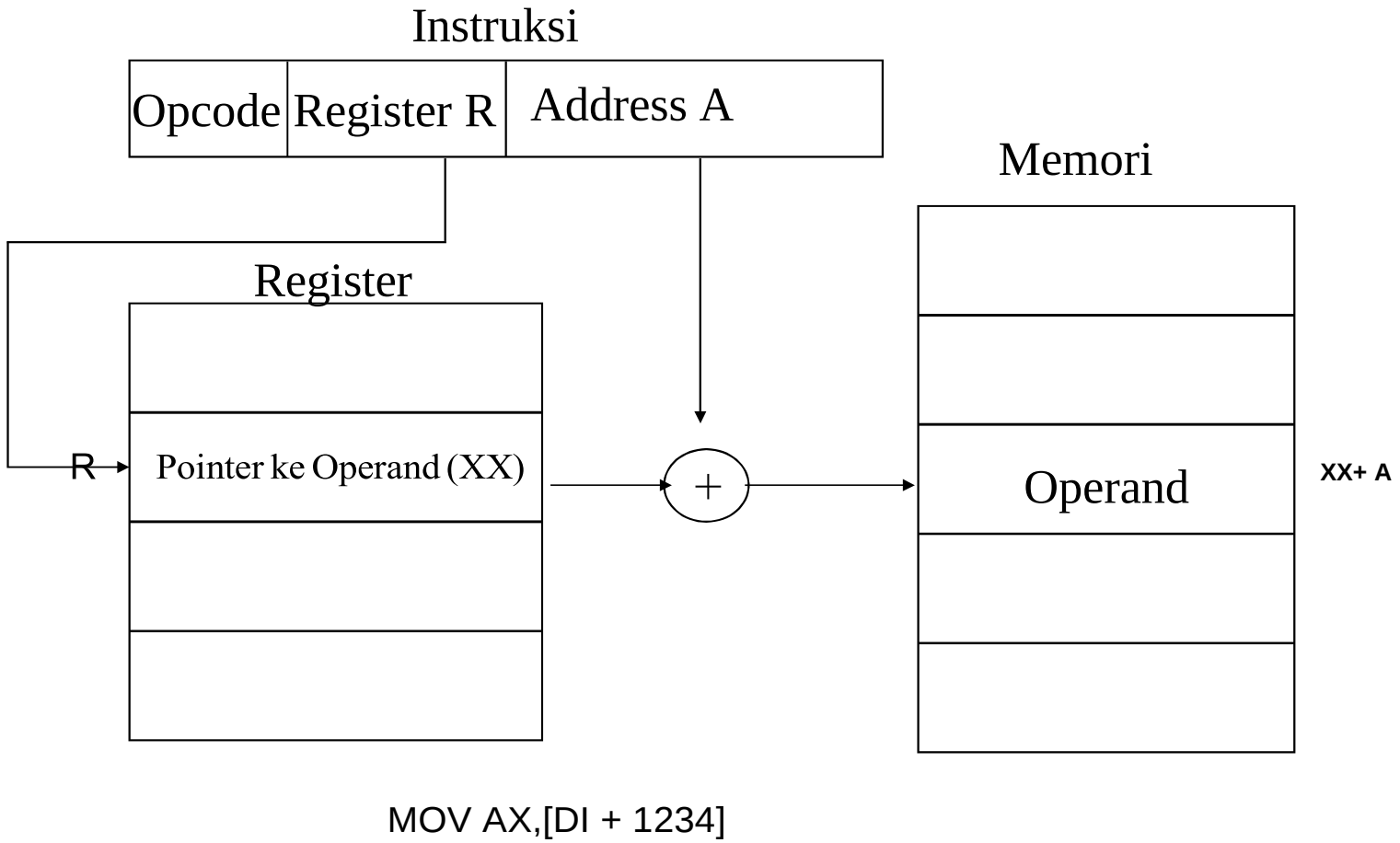
Diagram Pengalamatan Register Tak Langsung



Pengalamatan Register Tak Langsung

- EA = (R)
- Operand di memori yang ditunjukkan oleh isi register R
- Ruang alamat besar (2^n); n = lebar register
- Lebih sedikit satu referensi memori daripada pengalamatan tak langsung (baca memori hanya satu kali)

Diagram Pengalamatan Displacement



MOV AX,[DI + 1234]

Pengalamatan Displacement

- Menggabungkan kemampuan pengalamatan langsung dan pengalamatan register tak langsung
- EA = A + (R)
- Field alamat menyimpan 2 nilai
  - A = nilai dasar
  - R = register yang menyimpan nilai displacement
  - atau sebaliknya

Pengalamatan Relatif

- Suatu variasi dari pengalamatan displacement
- R = Program counter, PC
- EA = A + (PC)
- Alamat efektif adalah displacement relatif ke alamat dari instruksi
- Sangat baik untuk pengaksesan memori yang relatif *dekat* dengan instruksi yang sedang di eksekusi

Pengalamatan Register-Dasar

- Field alamat menyimpan displacement
- Register referensi menyimpan sebuah alamat memori
- Register referensi bisa eksplisit atau implisit
- Misal : register segment
- PA = S + EA

Pengalamatan ber-Indeks

- Field alamat mengandung sebuah alamat memori utama
- Register referensi mengandung displasment positif dari alamat tersebut
- EA = A + R
- Baik untuk mengakses array
  - EA = A + R
  - R++

Kombinasi

- Menggabungkan pengalamatan tak langsung dengan pengalamatan ber-indeks

- Post-index
- EA = (A) + (R)

- Pre-index
- EA = (A+(R))

Pengalamatan Stack

- Operand adalah puncak stack
- misal.
  - ADD Pop dua isi stack dan jumlahkan

Dasar Perancangan Set Intruksi

- Format instruksi
- Panjang instruksi

Format Instruksi

- Layout dari bit-bit sebuah instruksi
- Termasuk opcode
- Termasuk (implisit atau eksplisit) operand
- Terdapat lebih dari satu format intruksi pada set instruksi

Alokasi dari Bit-bit

- Jumlah mode pengalamatan
  - Satu atau dua bit tambahan mungkin diperlukan untuk menyatakan mode pengalamatan
- Jumlah operand
  - Biasanya 2 operand
  - Setiap alamat operand mungkin memerlukan indikator modenya
- Register vs memori
  - Referensi ke register memerlukan lebih sedikit bit daripada memori
- Jumlah dari set register
  - Satu set dari register serba guna
  - Dua atau lebih set khusus
- Jangkauan alamat
  - Pengalamatan langsung membatasi ruang alamat
  - Pengalamatan displacement memungkinkan ruang alamat lebih besar

Panjang Instruksi

- Terpengaruh dan mempengaruhi:
  - Ukuran Memori
  - Organisasi Memori
  - Struktur Bus
  - Kompleksitas CPU
  - Kecepatan CPU
- Trade off antara kemampuan instruksi dan pengiritan pemakaian memori

Soal :

- Jelaskan Hubungan antara jumlah bit pada opcode dengan jumlah instruksi yang ada ?
- Jelaskan hubungan antara jumlah bit pada Alamat yang ada di set instruksi dengan jumlah alamat yang bisa di jangkau
- Bagaimana cara agar set instruksi jumlah dapat menambahkan jangkauan pada memori ?
- Jelaskan kapan dan pada saat apa mode pengalamatan digunakan