Set Instruksi

Pertemuan 5

Oleh :

Riyanto Sigit, S.T, M.Kom

Nur Rosyid Mubtada'i S.Kom

Setiawardhana , S.T

Hero Yudo Martono, S.T

Politeknik Elektronika Negeri Surabaya - ITS

2005

Tujuan

- Memahami representasi set instruksi, dan jenis-jenis format instruksi
- Mengetahui jenis-jenis type operand digunakan
- Macam-macam Mode pengalamatan
- Format Instruksi
- Format Instruksi pada Pentium
- Memahami Implementasi Set Instruksi pada Pentium II

Sasaran

- Pengetahuan mengenai set instruksi sangat dirasakan manfaatnya oleh programer bahasa tinggat rendah, seperti bahasa asembler.
- Bagi programer bahasa tingkat rendah sangat memerlukan informasi tentang penggunaan register dan spesifikasinya, struktur memori, maupun format instruksinya.
- Bab ini akan mengupas tentang karakteristik mesin instruksi, tipe – tipe operasi, mode pengalamatan dan format instruksi

3.1. Karakteristik Mesin Instruksi

- Instruksi mesin (machine intruction) yang dieksekusi membentuk suatu operasi dan berbagai macam fungsi CPU.
- Kumpulan fungsi yang dapat dieksekusi CPU disebut set instruksi (instruction set) CPU.
- Mempelajari karakteristik instruksi mesin, meliputi
  - Elemen – elemen intruksi mesin
  - Representasi instruksinya
  - Jenis – jenis instruksi
  - Penggunaan alamat
  - Rancangan set instruksi

Arithmatika

- Komputer menyediakan operasi aritmetika seperti penjumlahan, pengurangan, perkalian, dan pembagian.
  - Operasi tersedia dalam bentuk fixed point maupun floating point.
  - Operasi lain, terutama untuk operand tunggal seperti: absolute, negate, increment, dan decrement disediakan untuk keperluan pembentukan fungsi operasi

Logika

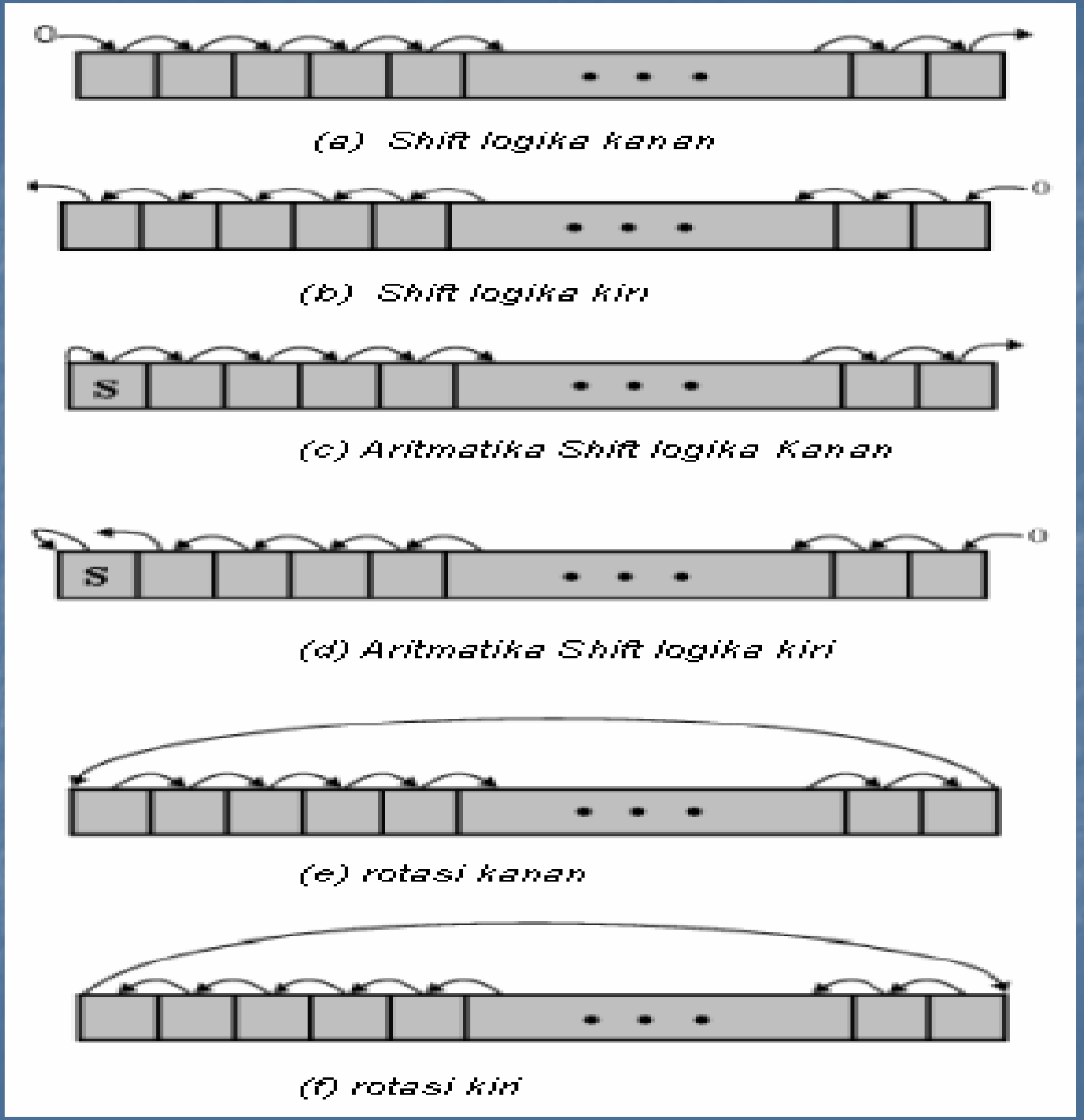
- Arsitektur CPU menyediakan operasi Boolean.
- Operasi ini untuk memanipulasi bit – bit word maupun alamat dalam membentuk operasi fungsi yang diinginkan
- Contoh :

(R1) = 1001 1101

(R2) = 0011 1001

(R1) AND (R2) = 0001 1001

Logika


0
(a) Shift logika kanan
0
(b) Shift logika kiri
S
(c) Aritmatika Shift logika Kanan
S
0
(d) Aritmatika Shift logika kiri
(e) rotasi kanan
(f) rotasi kiri

Apa Gunanya ?

- Operasi pergeseran logika sangat berguna dalam perpindahan data serial maupun pemecahan word menjadi beberapa bagian, misal 16 bit menjadi 8 bit.
- Dalam operasi penggeseran aritmetika, data diperlakukan sebagai integer bertanda dan tidak akan menggeser bit tandanya.
- Pergeseran ini biasanya untuk menangani overflow maupun underflow.
- Operasi rotasi berfungsi untuk membawa seluruh bit secara berurutan ke bit yang paling kiri.
- Bit – bit tersebut dapat diidentifikasikan dengan pemeriksaan tanda dari datanya (diperlakukan sebagai bilangan).

Konversi

- Instruksi konversi adalah instruksi yang mengubah format data.
- Contoh
  - Pengubahan bilangan desimal menjadi bilangan biner atau sebaliknya.
  - Pengubahan kode 8 bit menjadi kode lainnya

Input/Output

- Instruksi input/output telah dibahas pada matakuliah Organisasi Komputer.
- Instruksi ini berhubungan dengan kerja modul I/O terhadap CPU

Kontrol Sistem

- Instruksi ini merupakan instruksi khusus (priveleged instruction) yang hanya dapat dieksekusi ketika prosesor berada dalam kondisi khusus atau sedang mengeksekusi program yang berada pada areal khusus.
- Instruksi ini digunakan dalam sistem operasi.
- Contoh
  - Instruksi untuk membaca atau mengubah kunci proteksi penyimpanan (S/370),
  - Instruksi untuk akses blok kontrol proses pada sistem multiprogramming, mode kernel, dan lain - lain

Pemindahan Kontrol

- Instruksi kali ini akan sangat berbeda karena instruksi – instruksi sebelumnya adalah operasi bagi operand, sedangkan kali ini adalah instruksi yang dilakukan oleh instruksi berikutnya

Operasi transfer kontrol untuk apa ?

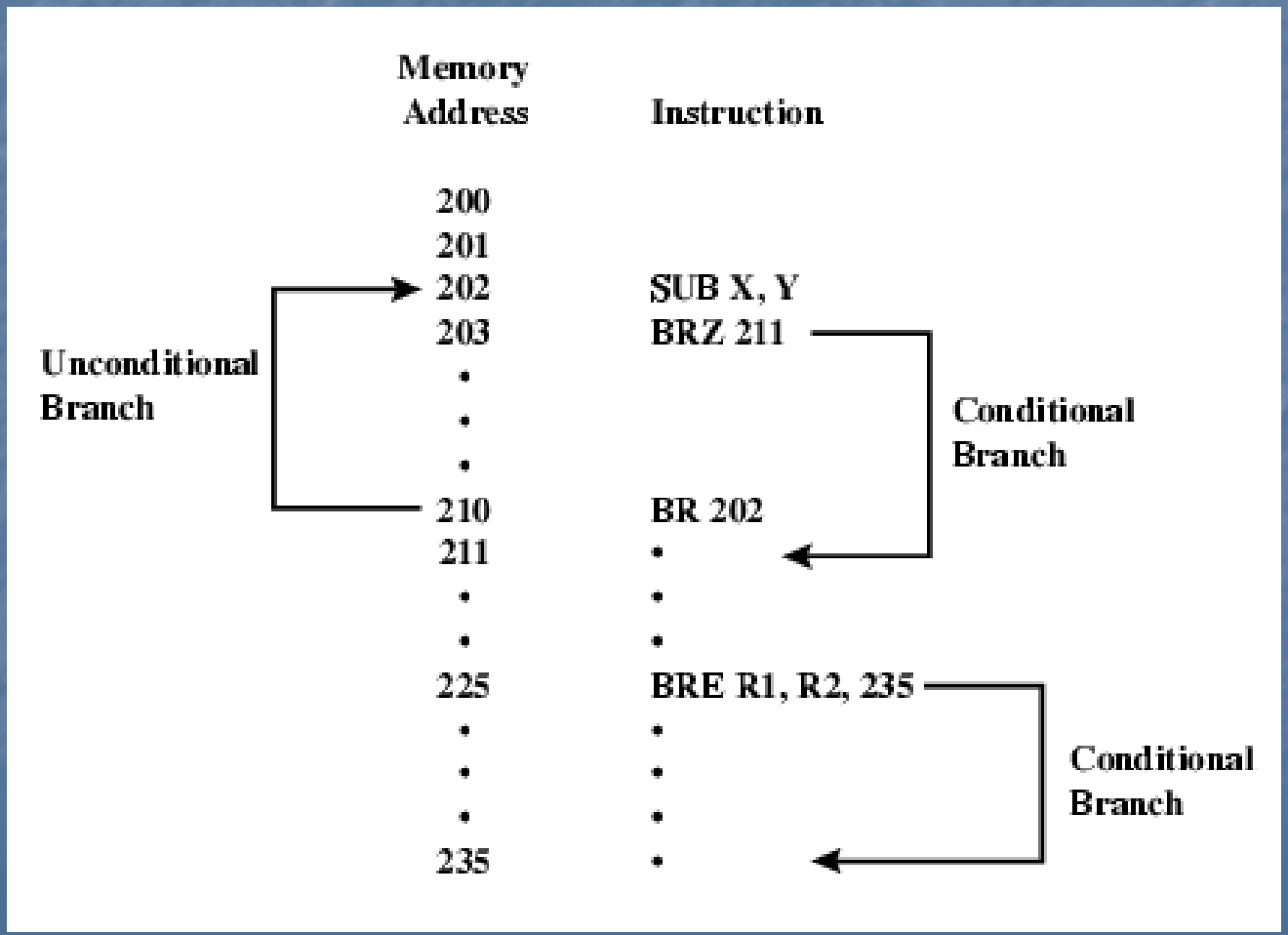
- Adanya aplikasi yang berulang dilakukan
- Adanya fungsi – fungsi persyaratan, yaitu operasi akan dilakukan apabila kondisi syaratnya terpenuhi
- Untuk kemudahan dalam pemrograman, yaitu program besar dapat dipecah – pecah menjadi program kecil yang modular

Set instruksi Operasi transfer kontrol

- Branch (percabangan)
- Skip (lompat/lewati)
- Subroutine call

Instruksi percabangan

- BRE R1, R2, X ; bercabang ke X bila isi R1 = R2



Instruksi ‘skip’

- Merupakan bentuk pemindahan kontrol lainnya.
- Menyatakan bahwa sebuah instruksi dapat dilewati.
- Tidak memerlukan field alamat tujuan

ISZ R1 ; Increment and Skip if Zero (tambahkan program counter (PC) dengan nilai 1 dan melompat ke instruksi berikutnya bila nilai register 1 adalah zero (0)).

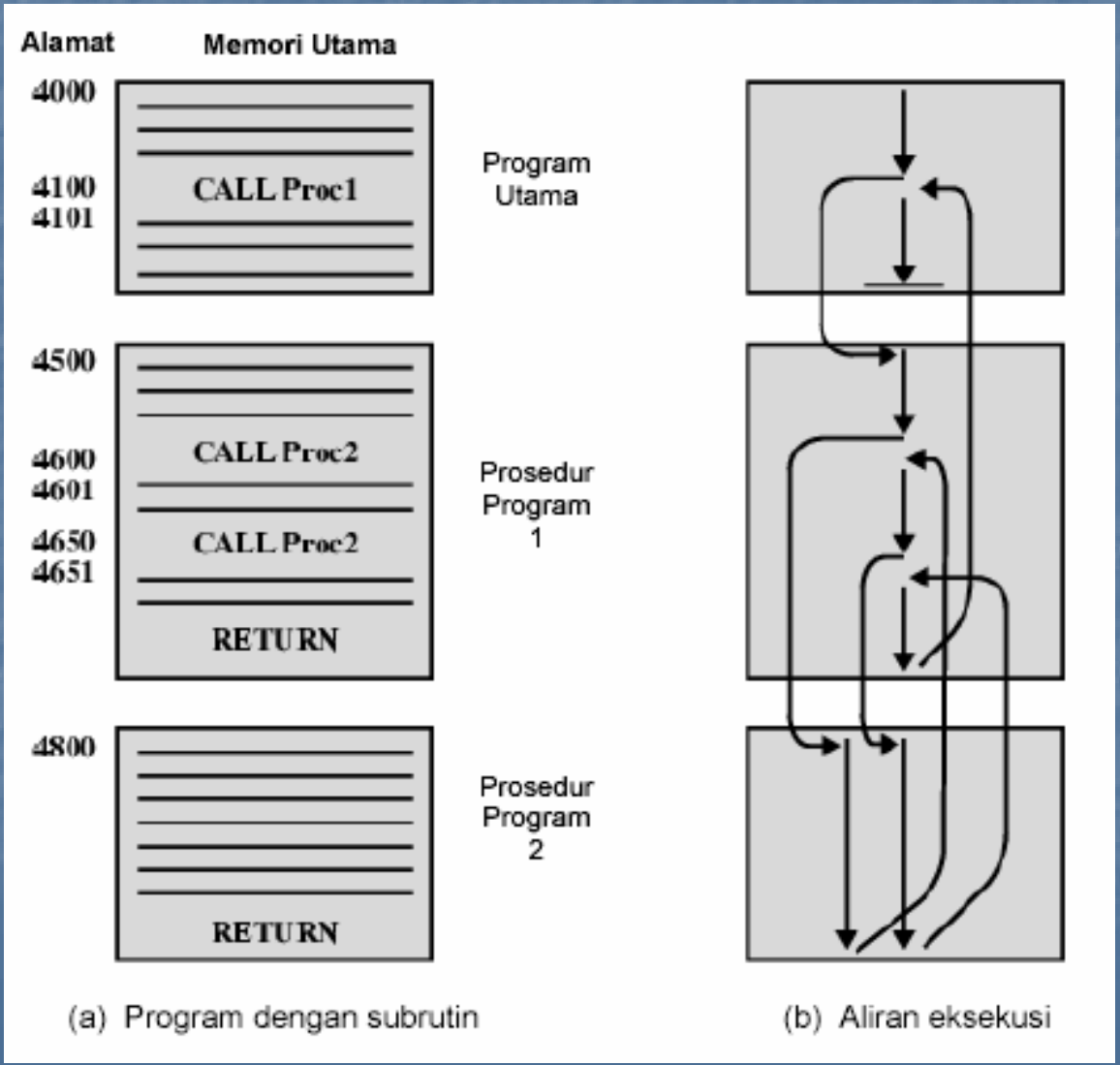
Subrutin

- adalah bentuk pemindahan kontrol lainnya
- adalah program modular yang merupakan bagian dari program komputer yang lebih besar.
- berguna bila potongan program tersebut digunakan berulang kali sehingga akan memudahkan dalam pemrograman

Mekanismenya bagaimana ?

- Melibatkan dua instruksi dasar
  - Instruksi pemanggilan (call instruction) yang bercabang ke lokasi subrutin
  - Instruksi kembali (return instruction) yang menggembalikan ke program pemanggilnya

Urutan eksekusi bersarang



Mengapa menyimpan informasi ?

- Karena pemanggilan subrutin dari sembarang tempat
- Untuk mekanisme kembali dari subrutin ke program utamanya.
  - Tempat yang digunakan untuk penyimpanan adalah
    - Register
    - Awal Subrutin (Start of Subroutine)
    - Puncak Stack (Top of Stack)

Ada Contoh ?

- Operasi "CALL X", apa yang terjadi ?

  RN PC +

  PC X
- RN adalah register yang digunakan untuk **menyimpan** alamat kembali.
- X adalah lokasi subrutin.
- Proses ini X sebagai alamat subrutin dimasukkan ke dalam program counter (PC) untuk dieksekusi.
- Subrutin yang dipanggil dapat menyimpan RN untuk digunakan dalam return nantinya.
- Sedangkan dalam metode **penyimpanan awal subrutin**, operasi CALL X ditandai dengan tindakan :

  X PC +

  PC X + 1

  Dengan operasi tersebut, maka subrutin secara otomatis menyimpan alamat untuk mekanisme kembali ke program utama

Ada Contoh ?

- Sistem stack sebagai penyimpanan
- Saat CPU mengeksekusi perintah CALL
  - CPU akan menaruh alamat pengembalian di atas stack.
- Metode ini lebih umum dan baik , Why ?
  - Pendekatan ini mampu mengakomodasikan subrutin reentrant.
  - Subrutin reentrant adalah subrutin yang memungkinkan pembukaan beberapa call dalam waktu bersamaan.
  - Instruksi CALL mengharuskan adanya pelewatan paramater (parameter passing) pada register.
  - Parameter tersebut sekaligus dapat disimpan pada stack

Penggunakan stack pada instruksi pemanggilan