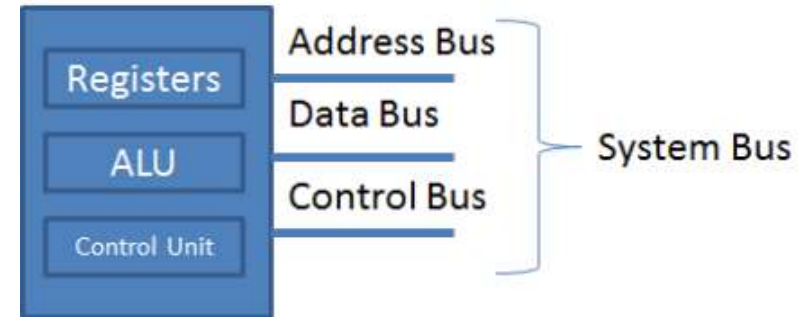
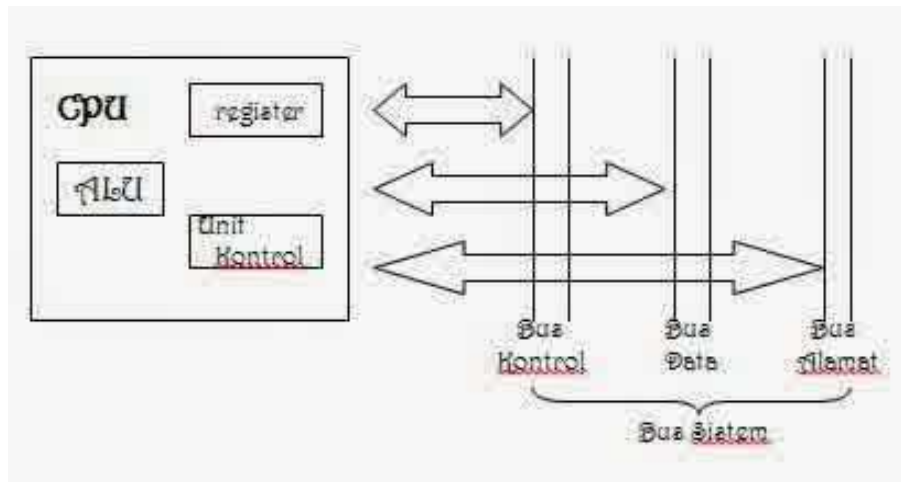


Organisasi & Fungsi Prosessor

Fungsi dasar yang dibentuk computer adalah eksekusi program. Program yang akan dieksekusi berisi sejumlah instruksi yang tersimpan di dalam memori. Central Processing Unit (CPU) melakukan tugas ini dengan cara mengeksekusi suatu program.

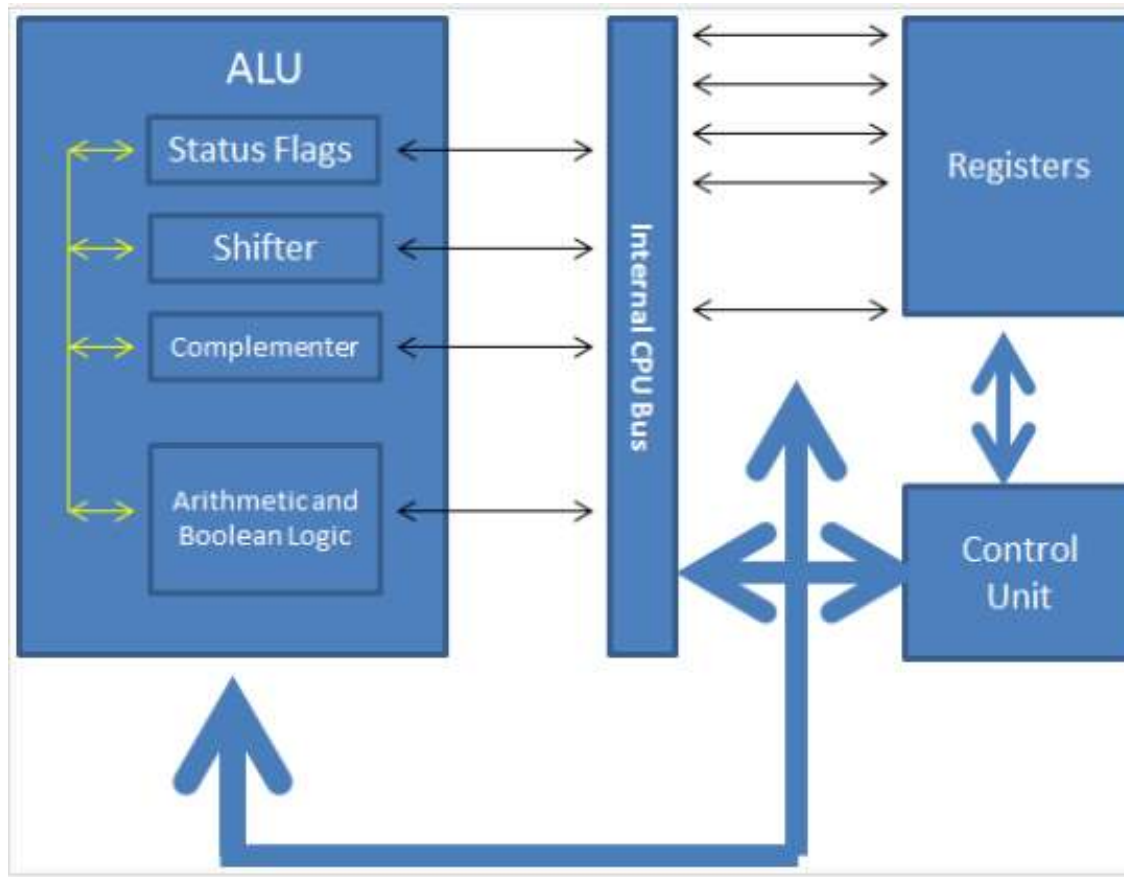
CPU adalah komponen pengolah data berdasarkan instruksi – instruksi yang diberikan.



Proses yang dilakukan CPU adalah :

1. Fetch Instruction / mengambil instruksi : CPU harus membaca instruksi dari memori.
2. Interpret Instruction / menerjemahkan instruksi : instruksi harus didekode untuk menentukan aksi apa yang diperlukan.
3. Fetch Data / mengambil data : eksekusi suatu instruksi mungkin memerlukan pembacaan data dari memori atau modul I/O.
4. Process Data / mengolah data : eksekusi suatu instruksi mungkin memerlukan operasi aritmetika atau logika terhadap data.
5. Write Data / menulis data : hasil eksekusi mungkin memerlukan penulisan data ke memori atau modul I/O.

Representasi sederhana untuk menyimpan instruksi dan data sementara dari sebuah prosesor dapat ditampilkan sebagai berikut ,



Gambar 1. Organisasi Register

Pengertian

☐ Status flag

Register untuk menyimpan tanda-tanda (Flags) seperti tanda minus, tanda plus tanda pinjam (Borrow) saat pengurangan, dll. Register Flag merupakan pembantu dari Register Akumulator.

☐ Shifler :

Rangkaian pelaksana untuk Instruksi Geser (Shift) , yang terdiri dari SHL (Shift Left) ,SHR (Shift Right) , dan RTA (Rotation) .

☐ Complementer :

Rangkaian untuk menentukan Complement dari setiap Sistem bilangan, misalnya : Complement -1 dari System bilangan Biner

☐ Aritmatik dan boolean logic :

Rangkaian untuk pembandingan (logika) melalui Rangkaian Logic untuk menentukan apakah sama ($=$) , lebih besar ($>$) , lebih kecil ($<$) , Tidak sama (\neq) , Lebih besar sama dengan (\geq) , Lebih kecil sama dengan (\leq) . Logika-logika diatas dirangkai dari Gate-Gate seperti AND, OR , dan NOT(Inverter)

Organisasi Register

Dalam prosesor ada satu set register yang berfungsi sebagai memori utama dan cache dalam hirarki / register tersebut dalam prosesor melakukan dua peran, yaitu :

- ❑ **User Visible register** : register ini memungkinkan pemrogram bahasa mesin dan bahasa assembler meminimalkan referensi main memori dengan cara mengoptimasi penggunaan register. Register ini adalah register yang dapat direfensikan dengan menggunakan bahasa mesin yang dieksekusi CPU.

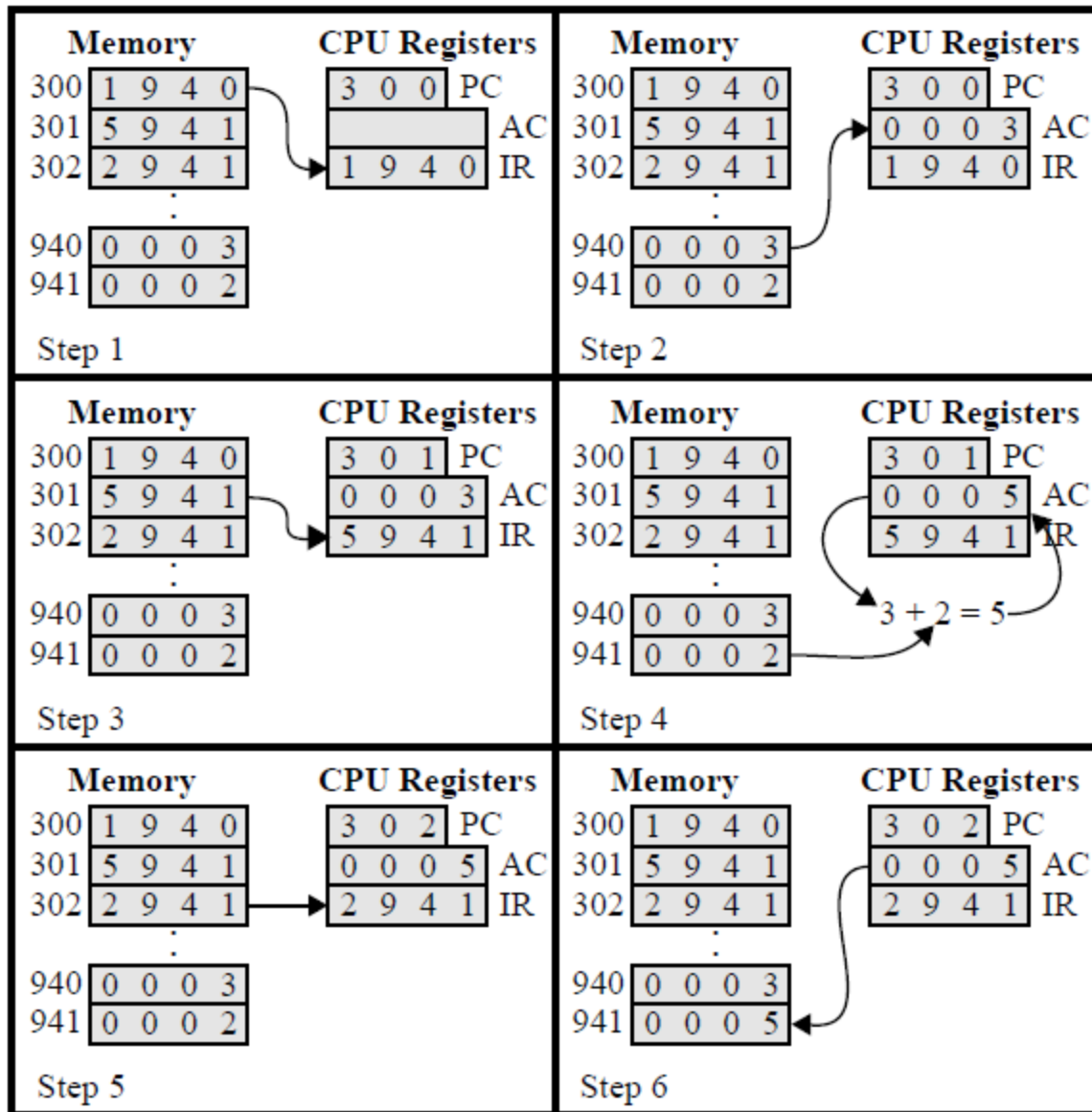
Secara virtual semua rancangan CPU modem memiliki sejumlah user-visible register, yang merupakan kebalikan akumulator tunggal. Kita dapat membedakannya dengan kategori-kategori berikut ini :

- ✓ Register General Purpose
- ✓ Register Data
- ✓ Register Alamat
- ✓ Register Kode-kode Kondisi

Control and Status register : register yang digunakan oleh CU, kontrol operasi CPU dan oleh sistem operasi untuk kontrol eksekusi program. Terdapat 4 macam register yang penting bagi eksekusi instruksi :

1. Program Counter (PC) atau pencacah program : berisi alamat instruksi yang akan diambil.
2. Instruction Register (IR) : berisi instruksi yang terakhir diambil.
3. Memori Address Register (MAR) : berisi alamat sebuah lokasi di dalam memori.
4. Memori Buffer Register (MBR) : berisi sebuah word data yang akan dituliskan ke dalam memori atau word yang terakhir dibaca.

Eksekusi Program



Penjelasan Gambar

Step 1 :

CPU menyiapkan instruksi yang akan dieksekusi, instruksi berada pada memori sedangkan instruksi yang akan dieksekusi berada di Instruction Register (IR) pada CPU Register. Instruksi yang pertama yang akan dieksekusi adalah **1940**.

Step 2 :

Instruction Register pada CPU Register berisi 1940, *artinya*

Angka 1 diasumsikan adalah memindahkan data dari memori yang beralamat **940** ke Accumulator (AC). → diperhatikan bahwa data dari memori beralamat 940 telah berada di AC (0003)

Step 3 :

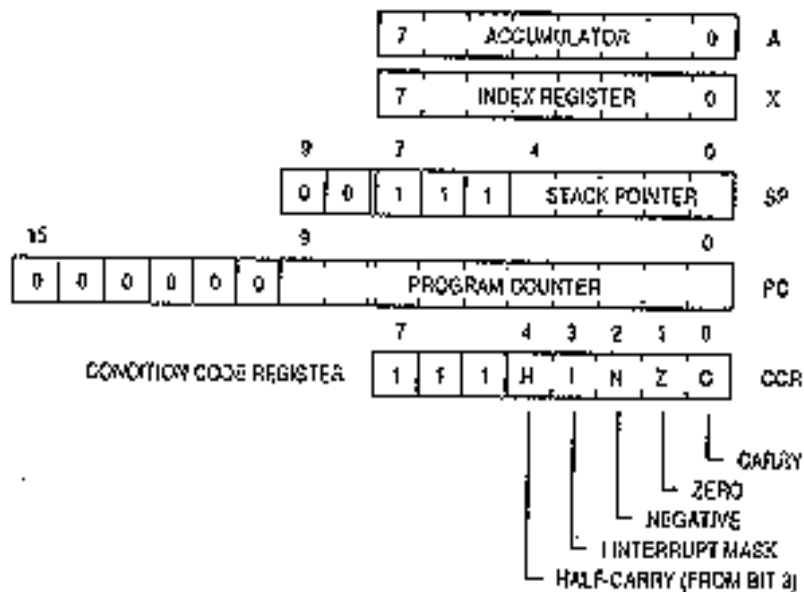
IR bernilai 5941 → diasumsikan bawah 5 melakukan pemindahan data dari memori yang beralamat 941 kemudian data tersebut dijumlahkan dengan data yang berada pada AC, selanjutnya hasil penjumlahan tersebut disimpan kembali ke AC.

Step 4- 6 silahkan diteruskan dipahami maksudnya.

Organisasi register meliputi jenis-jenis register di bawah ini :

1. General : terdapat 8 buah register general-purpose 32-bit. Register-register ini digunakan untuk semua jenis instruksi Pentium; dapat menampung operand-operand untuk keperluan kalkulasi alamat.
2. Segment : keenam register segmen 16-bit berisi pemilih segmen, yang diindex ke dalam tabel segmen. Register code segment (CS) mereferensi segmen yang berisi instruksi yang sedang dieksekusi. Register stack segment (SS) mereferensi segmen yang berisi user-visible stack (sebuah register yang diperuntukkan untuk menunjukkan bagian atas stack). Register-register segmen lainnya (DS, ES, FS, GS) mengizinkan pengguna untuk mereferensi hingga empat buah segmen data yang berlainan sekaligus.
3. [Flags](#) : register EFLAG berisi kode kondisi (persyaratan) dan bermacam-macam bit mode.
4. Instruction Pointer : berisi alamat instruksi saat itu.

1. Flag H (half carry) digunakan untuk operasi aritmatika BCD (Binary Coded Decimal) dan dipengaruhi oleh kerja instruksi ADD dan ADC. Bit H ini akan set jika ada carry yang timbul dari digit heksadesimal 0-3 (low order) dan digit desimal 4-7 (high order).
2. Bit I (interrupt mask) bukanlah status flag tetapi merupakan bit yang akan menonaktifkan semua sumber interrupt yang maskable saat bit ini diset. Interrupt baru akan aktif jika bit ini nol.
3. Flag N (negative) akan diset jika hasil dari operasi aritmatika, logika, maupun manipulasi data yang terakhir adalah negatif.
4. Flag Z (zero) diset jika hasil dari operasi aritmatika, logika, maupun manipulasi data terakhir adalah nol.
5. Flag C (carry/borrow) digunakan untuk menandai apakah ada carry dari hasil operasi tambah atau ada borrow.



Siklus Instruksi (Instruction Cycle)

Sebuah siklus instruksi meliputi subsiklus-subsiklus berikut ini :

- ☐ Fetch : membaca instruksi berikutnya dari memori ke dalam CPU.
- ☐ Execute : menginterpretasikan opcode dan melakukan operasi yang diindikasikan.
- ☐ Interrupt : apabila interrupt diaktifkan dan interrupt telah terjadi, simpan status proses saat itu dan layani interrupt.

Siklus Tak Langsung

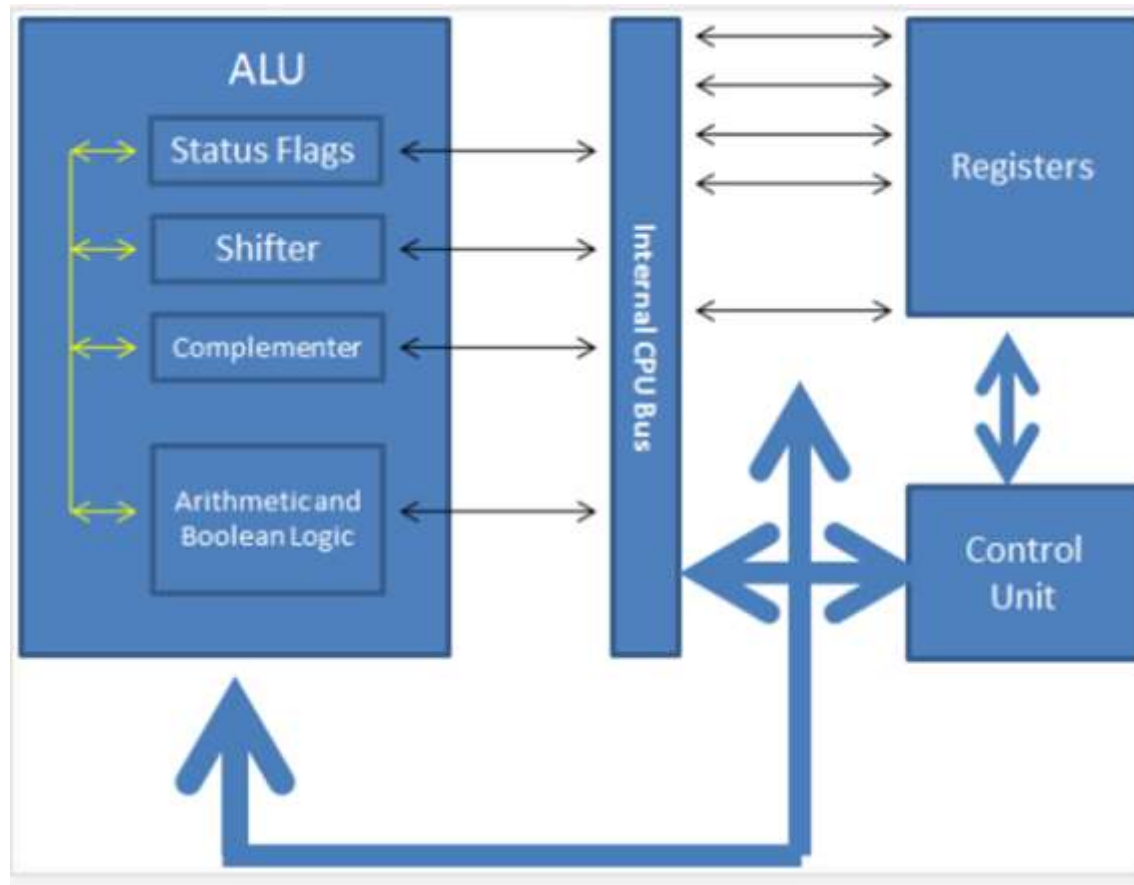
Eksekusi sebuah instruksi melibatkan sebuah operand atau lebih di dalam memori, yang masing-masing operand memerlukan akses memori. Kemudian, apabila digunakan pengalamatan tak langsung, maka diperlukan akses memori tambahan.



Pipelining Instruksi

1. Fetch instruction (FI) : membaca instruksi berikutnya yang diharapkan ke dalam buffer.
2. Decode instruction (DI) : menentukan opcode dan operand specifier.
3. Calculate operand (CO) : menghitung alamat efektif seluruh operand sumber. Hal ini mungkin melibatkan displacement, register indirect, indirect atau bentuk kalkulasi alamat lainnya.
4. Fetch operand (FO) : mengambil semua operand dari memori. Operand-operand yang berada di dalam register tidak perlu diambil.
5. Execute instruction (EI) : melakukan operasi yang diindikasikan dan menyimpan hasilnya, bila ada, di lokasi operand tujuan yang telah ditentukan.
6. Write operand (WO) : menyimpan hasilnya di dalam memori.

Tugas I



Berdasarkan skema gambar diatas, jelaskan bagaimana proses kerja dari organisasi register diatas.

Selesai