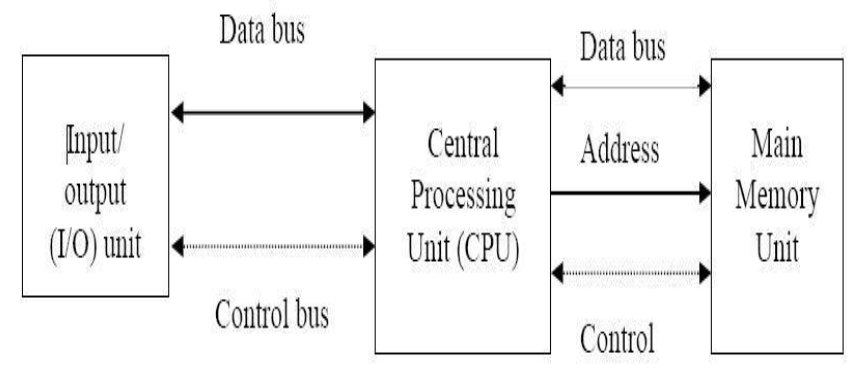
CENTRAL PROCESSING UNIT (CPU)

Arsitektur dasar mesin tipe Von Neumann menjadi kerangka referensi pada komputer digital umum (general-purpose) modern. 3 bagian fundamental tersebut adalah:

Mesin Von Neumann :



Program disimpan dalam unit memori utama yang berhadapan dengan piranti I/O melalui CPU. CPU membaca dari atau menulis ke memori, dengan mengirimkan alamat word ke unit memori melalui bus address kemudian menerima atau mengirimkan data melalui bus data. Data dipertukarkan antara CPU dan Unit I/O juga dengan menggunakan bus data. Operasi disinkronisasikan oleh dua bus control dengan sinyal kendali yang dikirimkan oleh CPU dan sinyal acknowledgment serta sinyal interupsi yang diterima oleh CPU.

Komponen CPU

Diagram blok sederhana sebuah CPU. Komponen CPU terbagi menjadi beberapa macam, yaitu sebagai berikut.

1.Unit kontrol

yang mampu mengatur jalannya program. Komponen ini sudah pasti terdapat dalam semua CPU.CPU bertugas mengontrol komputer sehingga terjadi sinkronisasi kerja antar komponen dalam menjalankan fungsi-fungsi operasinya. termasuk dalam tanggung jawab unit kontrol adalah mengambil intruksi-intruksi dari memori utama dan menentukan jenis instruksi tersebut. Bila ada instruksi untuk perhitungan aritmatika atau perbandingan logika, maka unit kendali akan mengirim instruksi tersebut ke ALU. Hasil dari pengolahan data dibawa oleh unit kendali ke memori utama lagi untuk disimpan, dan pada saatnya akan disajikan ke alat output. Dengan demikian tugas dari unit kendali ini adalah:

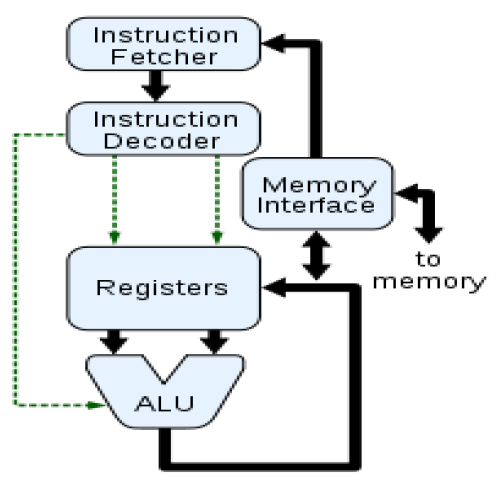
•Mengatur dan mengendalikan alat-alat input dan output.

•Mengambil instruksi-instruksi dari memori utama

•Mengambil data dari memori utama (jika diperlukan) untuk diproses.

•Mengirim instruksi ke ALU bila ada perhitungan aritmatika atau perbandingan logika serta mengawasi kerja dari ALU.

•Menyimpan hasil proses ke memori utama.



2.Register

merupakan alat penyimpanan kecil yang mempunyai kecepatan akses cukup tinggi, yang digunakan untuk menyimpan data dan/atau instruksi yang sedang diproses. Memori ini bersifat sementara, biasanya di gunakan untuk menyimpan data saat di olah ataupun data untuk pengolahan selanjutnya. Secara analogi, register ini dapat diibaratkan sebagai ingatan di otak bila kita melakukan pengolahan data secara manual, sehingga otak dapat diibaratkan sebagai CPU, yang berisi ingatan-ingatan, satuan kendali yang mengatur seluruh kegiatan tubuh dan mempunyai tempat untuk melakukan perhitungan dan perbandingan logika.

3.ALU

unit yang bertugas untuk melakukan operasi aritmetika dan operasi logika berdasar instruksi yang ditentukan. ALU sering di sebut mesin bahasa karena bagian ini ALU terdiri dari dua bagian, yaitu unit arithmetika dan unit logika boolean yang masing-masing memiliki spesifikasi tugas tersendiri. Tugas utama dari ALU adalah melakukan semua perhitungan aritmatika (matematika) yang terjadi sesuai dengan instruksi program. ALU melakukan semua operasi aritmatika dengan dasar penjumlahan sehingga sirkuit elektronik yang digunakan disebut adder.

Tugas lain dari ALU adalah melakukan keputusan dari suatu operasi logika sesuai dengan instruksi program. Operasi logika meliputi perbandingan dua operand dengan menggunakan operator logika tertentu, yaitu sama dengan (=), tidak sama dengan (¹ ), kurang dari (<), kurang atau sama dengan (£ ), lebih besar dari (>), dan lebih besar atau sama dengan (³ ).

4). CPU Interconnections

adalah sistem koneksi dan bus yang menghubungkan komponen internal CPU, yaitu ALU, unit kontrol dan register-register dan juga dengan bus-bus eksternal CPU yang menghubungkan dengan sistem lainnya, seperti memori utama, piranti masukan /keluaran.

Cara Kerja CPU

Saat data dan/atau instruksi dimasukkan ke processing-devices, pertama sekali diletakkan di RAM (melalui Input-storage); apabila berbentuk instruksi ditampung oleh Control Unit di Program-storage, namun apabila berbentuk data ditampung di Working-storage). Jika register siap untuk menerima pengerjaan eksekusi, maka Control Unit akan mengambil instruksi dari Program-storage untuk ditampungkan ke Instruction Register, sedangkan alamat memori yang berisikan instruksi tersebut ditampung di Program Counter. Sedangkan data diambil oleh Control Unit dari Working-storage untuk ditampung di General-purpose register (dalam hal ini di Operand-register). Jika berdasar instruksi pengerjaan yang dilakukan adalah arithmatika dan logika, maka ALU akan mengambil alih operasi untuk mengerjakan berdasar instruksi yang ditetapkan. Hasilnya ditampung di Accumulator. Apabila hasil pengolahan telah selesai, maka Control Unit akan mengambil hasil pengolahan di Accumulator untuk ditampung kembali ke Working-storage. Jika pengerjaan keseluruhan telah selesai, maka Control Unit akan menjemput hasil pengolahan dari Working-storage untuk ditampung ke Output-storage. Lalu selanjutnya dari Output-storage, hasil pengolahan akan ditampilkan ke output-devices.

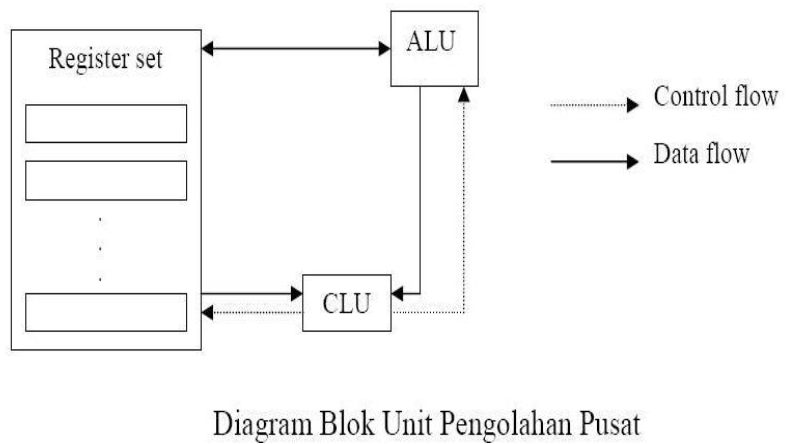
Fungsi CPU

CPU berfungsi seperti kalkulator, hanya saja CPU jauh lebih kuat daya pemrosesannya. Fungsi utama dari CPU adalah melakukan operasi aritmatika dan logika terhadap data yang diambil dari memori atau dari informasi yang dimasukkan melalui beberapa perangkat keras, seperti papan ketik, pemindai, tuas kontrol, maupun tetikus. CPU dikontrol menggunakan sekumpulan instruksi perangkat lunak komputer. Perangkat lunak tersebut dapat dijalankan oleh CPU dengan membacanya dari media penyimpan, seperti cakram keras, disket, cakram padat, maupun pita perekam. Instruksi-instruksi tersebut kemudian disimpan terlebih dahulu pada memori fisik (RAM), yang mana setiap instruksi akan diberi alamat unik yang disebut alamat memori. Selanjutnya, CPU dapat mengakses data-data pada RAM dengan menentukan alamat data yang dikehendaki.

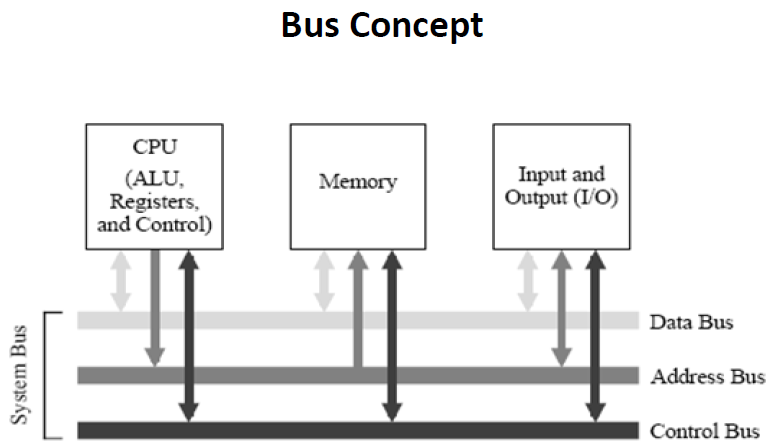
Saat sebuah program dieksekusi, data mengalir dari RAM ke sebuah unit yang disebut dengan bus, yang menghubungkan antara CPU dengan RAM. Data kemudian didekode dengan menggunakan unit proses yang disebut sebagai pendekoder instruksi yang sanggup menerjemahkan instruksi. Data kemudian berjalan ke unit aritmatika dan logika (ALU) yang melakukan kalkulasi dan perbandingan. Data bisa jadi disimpan sementara oleh ALU dalam sebuah lokasi memori yang disebut dengan register supaya dapat diambil kembali dengan cepat untuk diolah. ALU dapat melakukan operasi-operasi tertentu, meliputi penjumlahan, perkalian, pengurangan, pengujian kondisi terhadap data dalam register, hingga mengirimkan hasil pemrosesannya kembali ke memori fisik, media penyimpan, atau register apabila akan mengolah hasil pemrosesan lagi. Selama proses ini terjadi, sebuah unit dalam CPU yang disebut dengan penghitung program akan memantau instruksi yang sukses dijalankan supaya instruksi tersebut dapat dieksekusi dengan urutan yang benar dan sesuai.

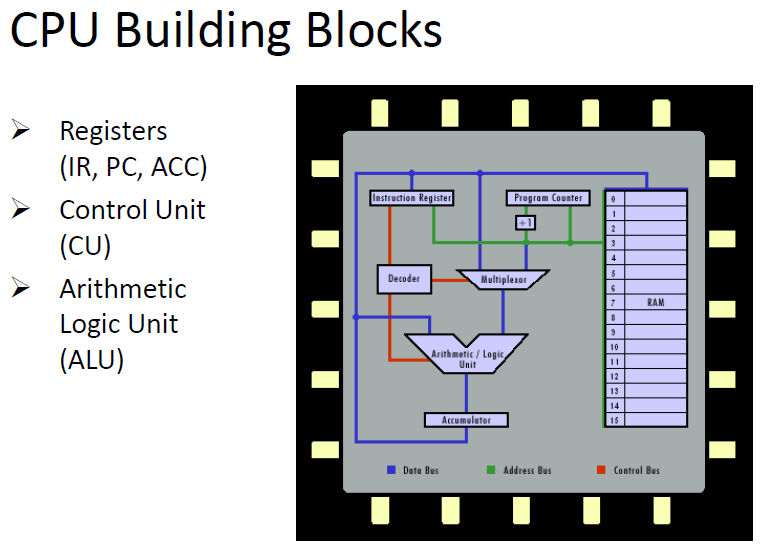
Kumpulan Register

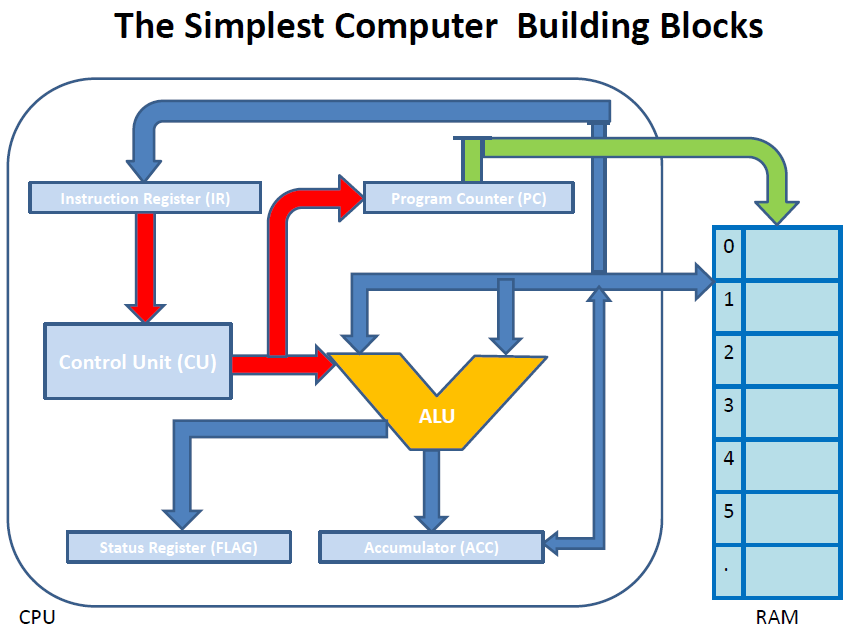
Register dari sebuah computer secara koektif disebur sebagai kumpulan register (register set).



Arithmetic Logic Unit ALU

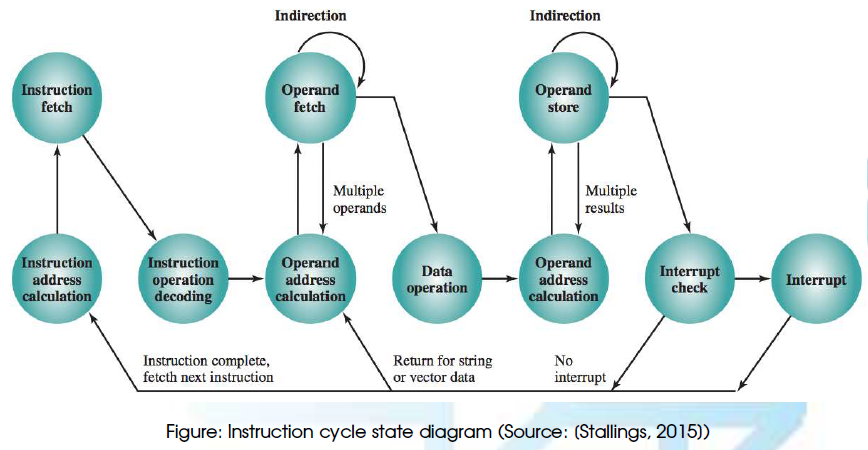






**Processor Structure and Function**

Processor Organization.



Requirements placed on the processor:

• Fetch instruction: reads an instruction from memory;

• Interpret instruction: determines what action to perform;

• Fetch data: if necessary read data from memory or an I/O module.

• Process data: If necessary perform arithmetic / logical operation on data.

• Write data: If necessary write data to memory or an I/O module.

To do these things the processor needs to:

• store some data temporarily

• remember the location of the next instruction;

while an instruction is being executed:

• In other words, the processor needs a small internal memory.

To do these things the processor needs to:

• store some data temporarily

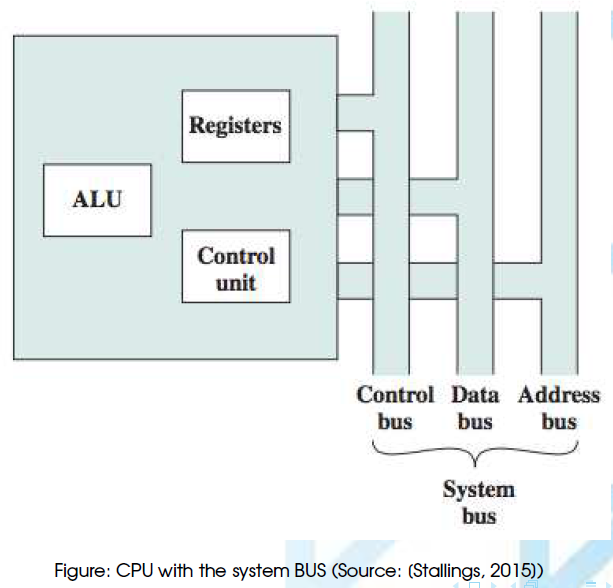
• remember the location of the next instruction;

while an instruction is being executed:

• In other words, the processor needs a small internal memory.

• Registers =)

We also need other components:



Major components of the processor:

• Arithmetic and Logic Unit (ALU):

• Performs computation or processing of data

• Control Unit:

• Moves data and instructions in and out of the processor;

• Also controls the operation of the ALU;

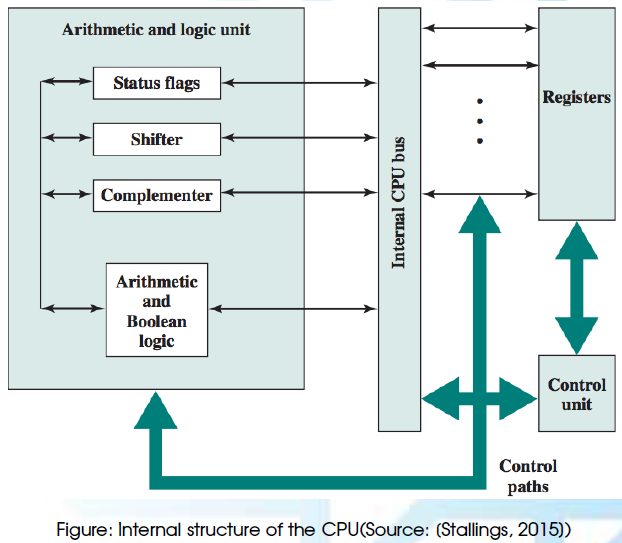
• Registers:

• Used as internal memory;

• System Bus:

• Acting as a pathway between processor, memory and I/O module(s);

A more detailed view:



Besides the usual elements the previous figure also includes:

• Internal CPU bus:

• Needed to transfer data between the various registers and the ALU;

• Logic control paths;

• Needed to specify which operations to perform;

Register Organization

Registers in the processor perform two roles:

• User-visible registers:

• Used as internal memory by the assembly language programmer;

• Control and status registers:

• Used to control the operation of the processor;

• Used to check the status of the processor / ALU;

Lets have a look at each one of these =)

User-visible Registers

May be referenced by the programmer, categorized into:

• General purpose

• Data

• Address

• Condition codes

General-purpose registers

Can be assigned to a variety of functions by the programmer:

• Memory reference & backup;

• Register reference & backup;

• Data reference & backup;

• These are the ones you use in the laboratory =)

Data registers

May be used only to hold data and cannot hold addresses:

• Must be able to hold values of most data types;

• Some machines allow two contiguous registers to be used:

• For holding double-length values.

Address Registers

Used to hold addresses, e.g.:

• Stack Pointer

• Program Counter

• Index Registers

Must be at least long enough to hold the largest address.

Condition Codes

Hold condition codes (a.k.a. flags):

• Flags are bits set by processor as the result of operations

• E.g.: an arithmetic operation may produce:

• a positive result;

• a negative result;

• a zero result;

• an overflow result.

Condition code bits are collected into one or more control registers:

In some machines:

• Interruption results in all user-visible registers being saved;

• These are then restored on return;

• Allows each subroutine to use the user-visible registers independently;

On other machines:

• responsibility of the programmer to:

• save the contents of user- visible registers prior to a subroutine call;

Control and Status Registers

Employed to control the operation of the processor:

• Mostly not visible to the user;

• Program counter (PC): Contains instruction address to be fetched;

• Instruction Register (IR): Contains last instruction fetched;

• Memory address register (MAR): Contains memory location address;

• Memory buffer register (MBR): Contains:

• a word of data to be written to memory;

• a word of data read from memory.

In general terms:

• Processor updates PC after each instruction fetch;

• A branch or skip instruction will also modify the contents of the PC;

• The fetched instruction is loaded into an IR

• Data are exchanged with memory using the MAR and MBR, e.g.:

• MAR connects directly to the address bus

• MBR connects directly to the data bus

The four registers just mentioned are used for:

• Data movement between processor and memory;

• Within the processor, data must be presented to the ALU for processing:

• ALU may have direct access to the MBR and user-visible registers;

• Alternatively:

• There may be additional buffering registers within ALU;

• These registers serve as input and output registers for the ALU;

• These registers exchange data with the MBR and user-visible registers.

Many processors include a program status word (PSW) register:

• Contains condition codes plus other status information

• Common fields or flags include the following:

• Sign: Sign bit of the result of the last arithmetic operation;

• Zero: when the result is 0;

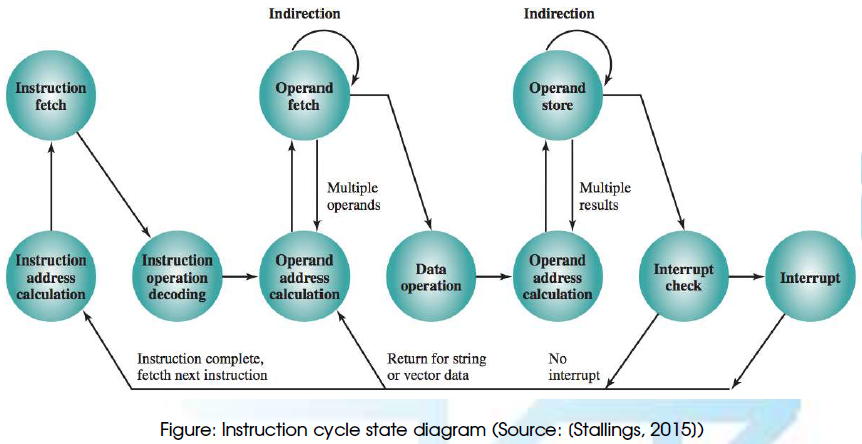
• Carry: Set if an operation resulted in a carry/borrow bit;

• Equal: Set if a logical compare result is equality.

• Overflow: Used to indicate arithmetic overflow.

• Interrupt Enable/Disable: Used to enable or disable interrupts.

Instruction Cycle



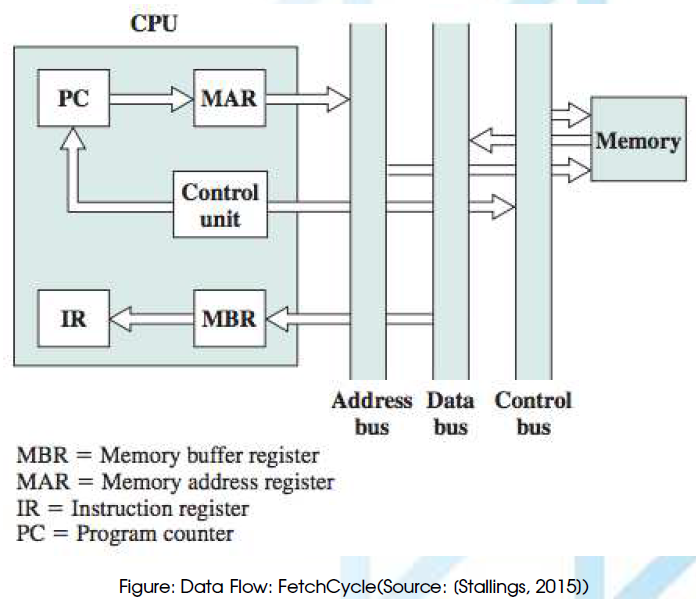
Now that we know more about the inner workings of:

• CPU;

• Registers;

• Bus

The flow of data during the instruction **fetch** cycle



The flow of data during the instruction fetch cycle:

1 PC contains the address of the next instruction to be fetched;

2 Address is moved to the MAR and placed on the address bus;

3 Control unit requests a memory read;

4 Result is:

• placed on the data bus;

• copied into the MBR;

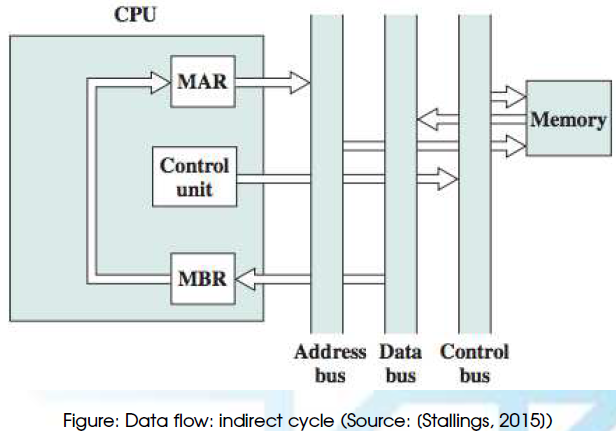
• then moved to the IR.

5 Meanwhile, the PC is incremented by 1;

Once the fetch cycle is over, control unit examines IR:

1 to determine if it contains an operand specifier using indirect addressing;

2 If so, an indirect cycle is performed:



The indirect addressing cycle:

• Bits of the MBR containing the address are transferred to the MAR;

• Control unit then requests a memory read:

• to get the desired address of the operand into the MBR.

The execute cycle takes many forms:

• Depending on the operation to be performed...

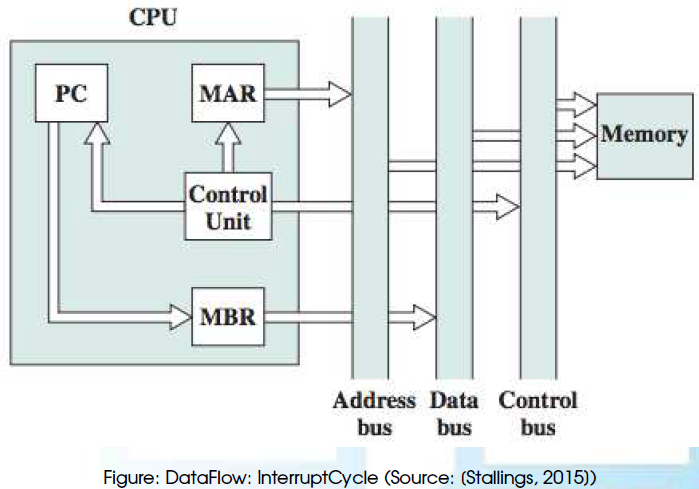
• May involve:

• transferring data among registers

• read or write from memory or I/O

• and/or the invocation of the ALU.

Then comes the interrupt cycle:



The interruption cycle:

• Contents of the PC must be saved;

• Thus the contents of the PC are

• Transferred to the MBR to be written into memory.

• Special memory location is loaded into the MAR:

• E.g.: stack pointer (SP)

• PC is loaded with the address of the interrupt routine.