

**“Basics of microcontroller”**

*Lecture 2.*

***Microprocessor architecture***

**Lecturer: Onon Otgonbaatar, MD.**

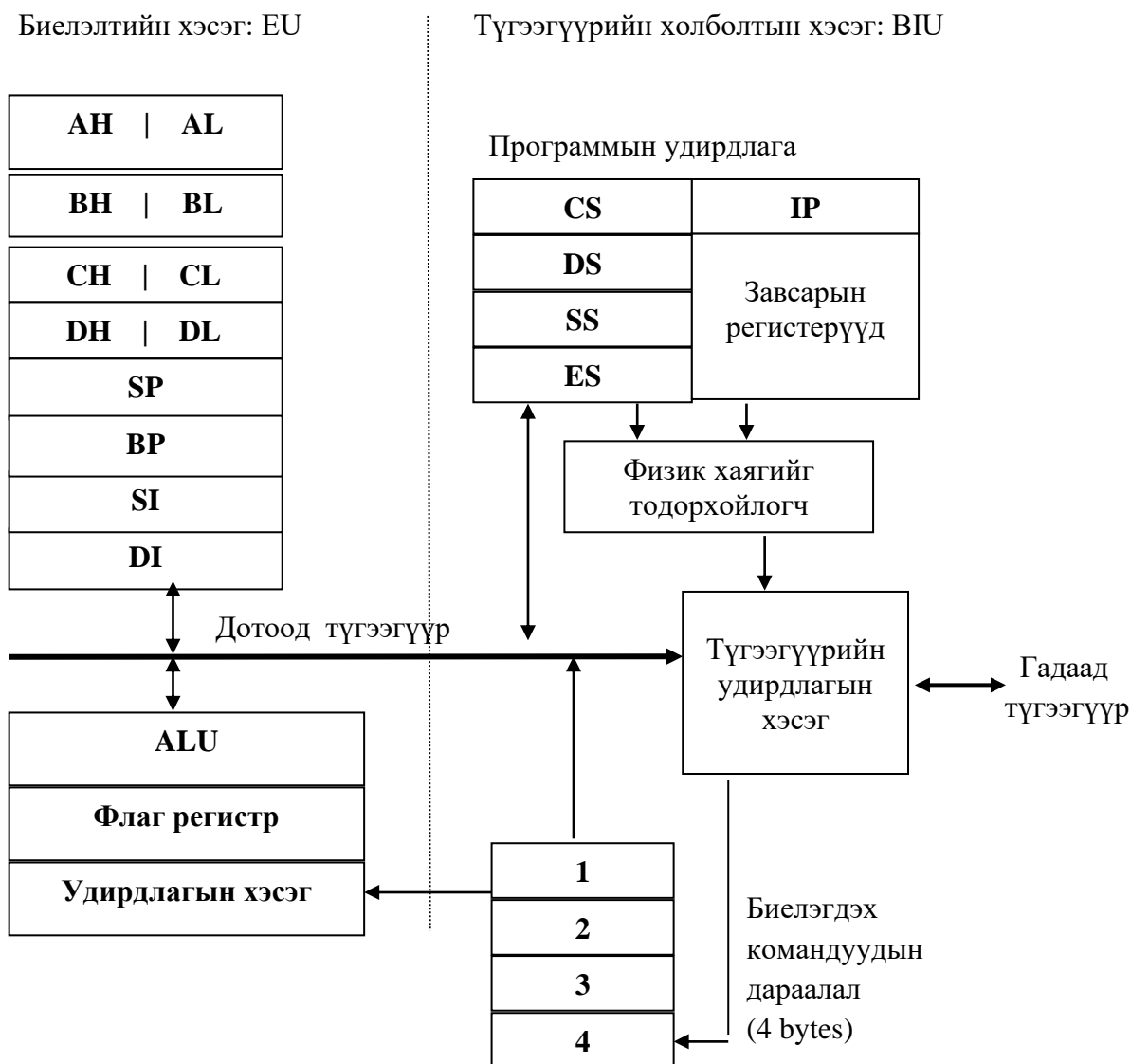
## Микропроцессорын бүтэц

Микропроцессор гэдэг нь төрөл бүрийн функцүүдийг гүйцэтгэх чадвартай программчилагддаг контроллер юм. Микропроцессорын бүтцийг ойлгохын тулд жишээ болгон 8088 микропроцессорыг авч үзье.

Энэ нь Intel компанийн гаргасан дотроо 16 битээр, гаднаа 8 битээр өгөгдөл солилцдог, IBM фирмийн анхны компьютерийн хэрэглэсэн микропроцессор юм. Энэ процессор нь 16 битээр арифметик үйлдлийг гүйцэтгэж чадах ба регистерүүд нь 16 битийн, дотоод түгээгүүрүүд нь мөн 16 битийнх бөгөөд харин гадаад өгөгдлийн түгээгүүр нь 8 битийнх юм. Иймд микропроцессор ба санах ой, микропроцессор ба оролт/гаралтын портуудын хооронд нэгэн зэрэг 8 бит өгөгдлийг солилцоно.

8088 микропроцессорыг бүтцийн хувьд 2 хэсэгт хувааж болно.

- EU (execution unit) буюу Биелэлтийн хэсэг
- BIU (bus interface unit) буюу Түгээгүүрийн холболтын хэсэг



ALU – арифметик логик байгууламж нь микропроцессор, санах ой, регистрүүд хооронд мэдээлэл солилцох, арифметик ба логик үйлдлүүд, шилжүүлэх, эргүүлэх зэрэг үйлдлүүдийг гүйцэтгэдэг микропроцессорын өгөгдөл боловсруулалтад гол үүрэгтэй хэсэг юм. Энэ нь 2 оролттой (оролтууд нь регистр ба санах ой байна.) бөгөөд, үйлдлийн үр дүнгээ мөн нэг регистрт (зарим процессорууд аккумуляторт) хадгалдаг. Удирдлагын хэсэг ирсэн командыг тайлж, АЛУ-д тохирох удирдлагын сигналыг явуулан, АЛУ-гийн ямар үйлдэл хийхийг тодорхойлно. Үр дүнгийн өөрчлөлтөөр Flag регистр мөн зохих утгуудаа авдаг.

Түгээгүүрийн холболтын хэсгийн хангаж өгсөн команд ба өгөгдлөөр Биелэлтийн хэсэг нь командуудыг биелүүлнэ. Түгээгүүрийн холболтын хэсэг нь түгээгүүрүүдийг удирдан, өгөгдлийг биелэлтийн хэсэг, санах ой, гадаад оролт гаралтын төхөөрөмжүүд үрүү дамжуулна. Сегмент регистрүүд нь хаяглалтыг удирдах ба санах ой дахь командуудын дарааллуудыг хангана.

Физик хаягийг тодорхойлогч нь 16 бит сегмент хаяг ба offset хаяг 2-оор тухайн мэдээллийн физик хаягийг нэмэх зарчимаар гаргана. Энэ тухай дараагийн бүлэгт тодорхой ярина.

## ХАЯГЛАЛТ

Санах ойг хаяглах хэд хэдэн төрлийн арга байдаг. Сегмент хаяг нь CS, DS, SS, ES гэсэн 4 сегмент регистрийн аль нэгийг нь зааж байгаа утга байдаг ба сегмент доторх offset хаяг нь заагч (SP, BP) регистрүүдийн утга, индекс (DI, SI) регистрүүдийн утга зэргээр, эсвэл дээрх 3 төрлийн регистрүүдийн комбинацуудаар тодорхойлогддог.

Санах ойн хаяглалтын горимуудыг дурдвал:

### 1. Регистрийн хаяглалт

Регистрийн хаяглалтын операнд нь 8 буюу 16 битийн регистрүүдийн агуулж буй утга юм. Өөрөөр хэлбэл, операнд нь AX, BX, CX, DX зэрэг ерөнхий регистрүүд буюу DS, SS, CS, ES зэрэг сегментний регистрүүдийн аль нэгд нь байрлана.

```
MOV    AX, SI
ADD    DI, BX
AND    CL, AX
XOR    AL, AH
INC    AL
```

### 2. Утгаар хаяглалт

Энэ арга нь 8 ба 16 битийн тогтмол утгуудыг командад агуулна.

```
SUB    AL, 30H
MOV    COUNT, 10
AND    AX, 0F00H
XOR    DH, 1
CMP    CHAR, '*'
ADD    AX, -4
```

### 3. Шууд хаяглалт

*Offset address = 16 bit address*

Энэ хаяглалтын үед командын машины кодын 3 ба 4 дэх байтууд нь операндын offset хаягийг агуулдаг. Ассемблер хэлэнд тэмдэгт нэрний санах ой дахь хаягийг тайлж offset хаягийг нь олдог. Энэ хаяглалт аль нэг гишүүнд нь л хэрэглэгдэнэ, нэг зэрэг хоёуланд нь хэрэглэгдэж болохгүй. Операндын хаяг нь командын талбарт шууд агуулагдана. Жишээ нь:

```
MOV    COUNT, 4
INC    MM1
MOV    AX, GAMMA
ADD    TEMP, BL
```

### 4. Регистрээр дамжуулан шууд бусаар хаяглалт

*Offset address = [SI / DI / BX]*

Энэ аргад санах ой дахь гишүүний offset хаяг нь BX, DI, SI регистрүүдийн 16 бит утга бөгөөд дөрвөлжин хаалтанд бичигддэг. Сегмент хаягаараа өгөгдлийн сегмент регистрийн утгыг авна. Жишээ нь:

MOV BX, [SI] гэсэн үйлдэл нь өгөгдлийн сегментийн SI гэсэн offset хаяган доторх утгыг авч BX регистрт хийнэ. DS регистрээс сегмент хаягаа авна. DS:SI регистрийн хос санах ойг хаяглаж байна гэсэн үг.

Харин ES:DI регистрийн хос санах ойг хаяглаж байвал командаа MOV BX, ES:[DI] гэж бичнэ, учир нь DS регистрээр сегментний хаягаа авдаг болохоор extra (нэмэлт) сегмент ашиглах үед ES регистрийг зааж өгөх шаардлагатай.

```
ADD    AX, [DI]
MOV    [SI], CL
```

### 5. Индексний хаяглалт

*Offset address = [SI / DI] + 8/16 bit address*

Санах ой дахь гишүүний offset хаяг нь командад заасан гишүүний хаяг ба DI, SI индекс регистрүүдийн агуулж буй (дөрвөлжин хаалтанд бичигддэг) 16 бит утгын нийлбэр байна. Жишээ нь:

Өгөгдлийн сегментэд

```
TBL    DW    40    DUP(?)    гэж зарласан бол
ADD    TBL[DI], 3    гэсэн команд нь хэрэв DI=2(n-1) бол 3 гэсэн
утгыг TBL-ийн n-р элемент дээр нэмнэ. Өөрөөр хэлбэл, word урттай утгууд учир
DI=6=2(4-1) учир TBL-ийн 4-р элемент дээр 3-г нэмнэ. (4-р элемент нь 6 ба 7
дугаар байтууд байна.)
```

```
MOV    ARRAY[SI], AX
ADD    CX, ROW[DI]
XOR    DL, MATRIX[SI] гэх мэтээр бичигдэнэ.
```

## 6. Бааз хаяглалт (base addressing)

$$\text{Offset address} = [BX / BP] + 8/16 \text{ bit address}$$

Санах ой дахь гишүүний offset хаяг нь команд дахь хаяг ба BP, BX региструудийн нийлбэр юм. Жишээ нь:

MOV DX, 6[BP] нь MOV DX, [BP]+6 эсвэл MOV DX, [BP+6] гэсэнтэй ижил юм.

MOV AX, [BP]10

MOV AX, [BP+10]

ADD [BX]TEMP, CX ; temp нь тогтмол утга гэсэн үг.

Стек ойтой ажиллах үед 2-г нэмэх нь стекийн оройн утгыг стек ойн 1 нүдээр хорогдуулна, 4-г хасвал стекийн оройн утгыг стек ойн 2 нүдээр нэмэгдүүлнэ гэсэн үг.

## 7. Бааз-индексний хаяглалт (base indexed addressing)

Бааз-индекс хаяглалтын үед санах ой дахь гишүүний offset хаяг нь доорхуудын аль нэг нь байна.

$$\text{Offset address} = [BX / BP] + [SI / DI]$$

$$\text{Offset address} = [BX / BP] + [SI / DI] + 8/16 \text{ bit address}$$

Бааз-индексний хаяглалт нь доорх хэлбэрүүдээр бичигдэнэ.

<name> [base] [<index>]

[<base> + <constant>] [<index>]

[<base>] [<index> + <constant>]

[<base> + <index> + <constant>]

[<base> + <constant> + <index>]

name – санах ойн үүрэн дэх гишүүний нэр

base - дөрвөлжин хаалтанд BX, BP региструудийн нэг байна.

index – дөрвөлжин хаалтанд DI, SI региструудийн нэг байна.

constant – 8 буюу 16 битийн тогтмол утга байна. 2 хэмжээст матрицын элементэд хандахад хэрэглэгдэнэ. Жишээ нь:

ADD [BX]ALPHA[SI], DI

CMR [BP]BETA[DI], 4000H

SUB [BX]GAMMA[DI], 1000

## 8. Оролт/гаралтын портын хаяглалт

Порт гэдэг нь процессорыг гадаад орчинтой холбодог төхөөрөмж юм. Портоор процессор нь оролтын төхөөрөмжөөс сигналыг хүлээж авах ба гаралтын төхөөрөмж үрүү сигналыг илгээнэ. Процессор нь 65536 хүртэл портуудтай холбогдох боломжтой. Оролтийн горимд заасан хаягтай портоос байт өгөгдлийг авч AL регистрт, эсвэл word хэмжээтэй өгөгдлийг авч AX регистрт хийнэ.

Гаралтийн горимд DS:SI регистрийн хосоор хаяглагдсан санах ойн үүрнээс өгөгдлийг авч DX регистрээр хаяглагдсан порт уруу гаргана. Хэрэв чиглэлийн флаг бит 0 бол SI регистр нэмэгдэх замаар, 1 бол SI регистр хорогдох замаар дараагийн санах ойн үүрийг заана. Байтыг гаргах бол 1-ээр, word-ийг гаргах бол 2-оор өөрчлөгдөнө.

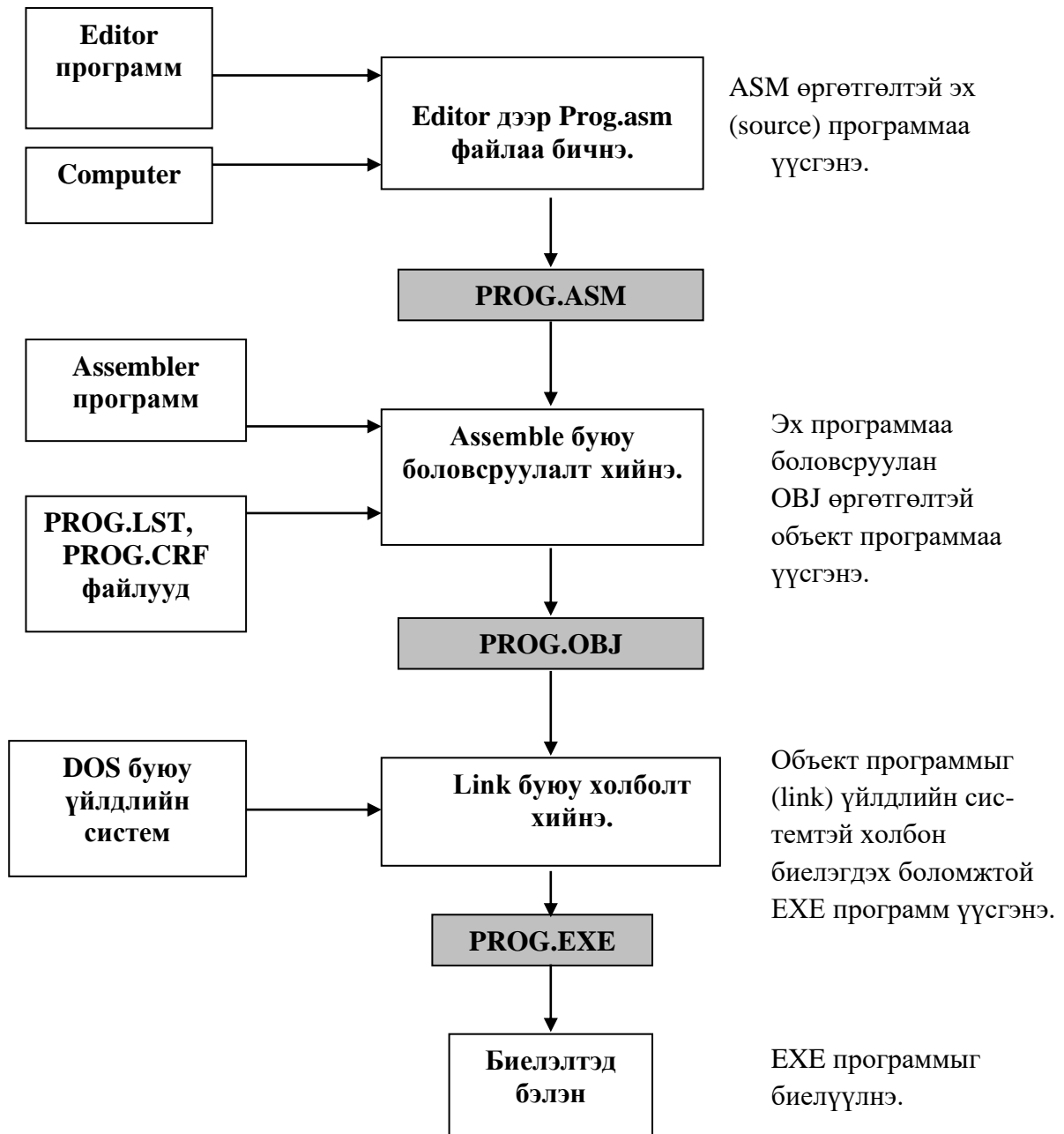
IN AL, 40H ; 40h хаягтай портноос al-д байт утга авна.

MOV DX, 50H ; 50H портноос ах-д word утга авна.

IN AX, DX

OUT DX, AX ; dx-ийн заах хаягтай порт уруу ax-ийн утгыг гаргана.

## Ассемблер программын боловсруулалтын үе шатууд



Ассемблер нь эх программаа объект файл уруу хөрвүүлэх ба link нь объект файлыг биелэгдэх боломжтой EXE файл уруу хөрвүүлнэ.

### **Textbook**

1. Peter Abel, "IBM PC Assembler language and programming", USA, 1987
2. Jim Mischel, "Macro magic with Turbo Assembler ", USA, 1993
3. William C.Runnion, "Structured programming in Assembly Language for the IBM PC", Boston, 1988
4. Thomas A.Wadlow, "Memory resident programming on the IBM PC", USA, 1987
5. Robert S.Lai, "Writing MS-DOS device drivers", USA, 1987
6. E. Majigsuren, "IBM assembly language", MGL, 2003
7. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay, "The 8051 microcontroller and Embedded Systems Using Assembly and C",USA, 2007