

# Introduction to 8086 Assembly

## Lecture 2

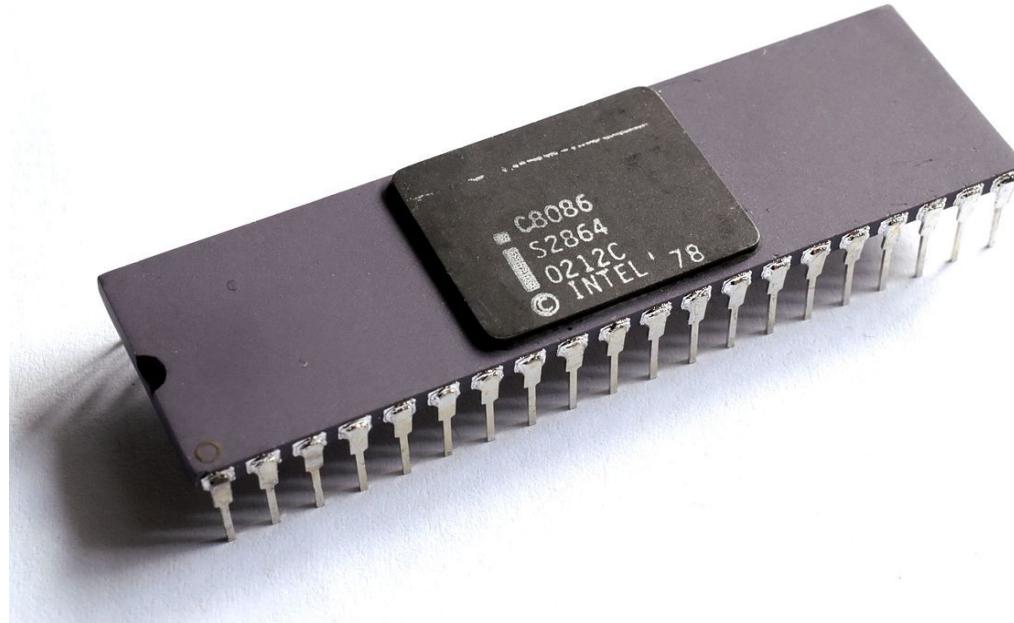
80x86 architecture, registers and basic assembly



1928  
K. N. Toosi University of Technology

# Intel 8086 microprocessor

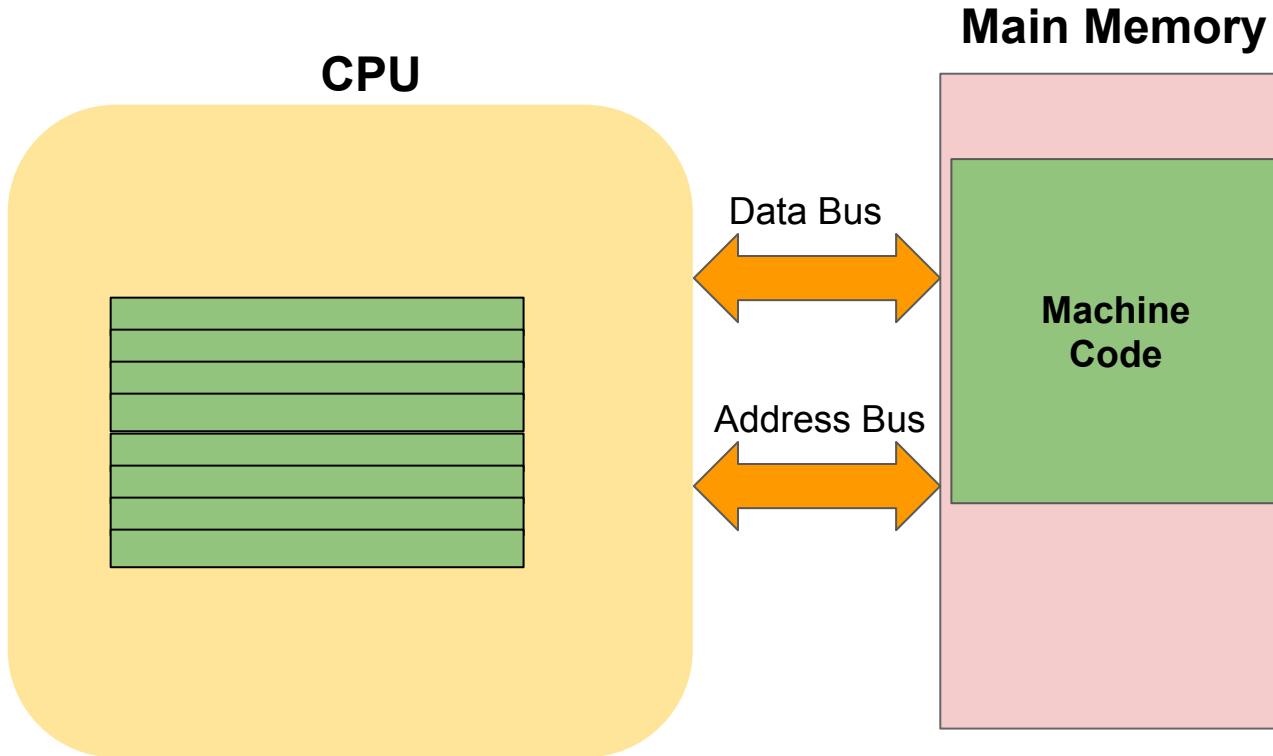
- released in 1978
- 16 bit
- The x86 Architecture



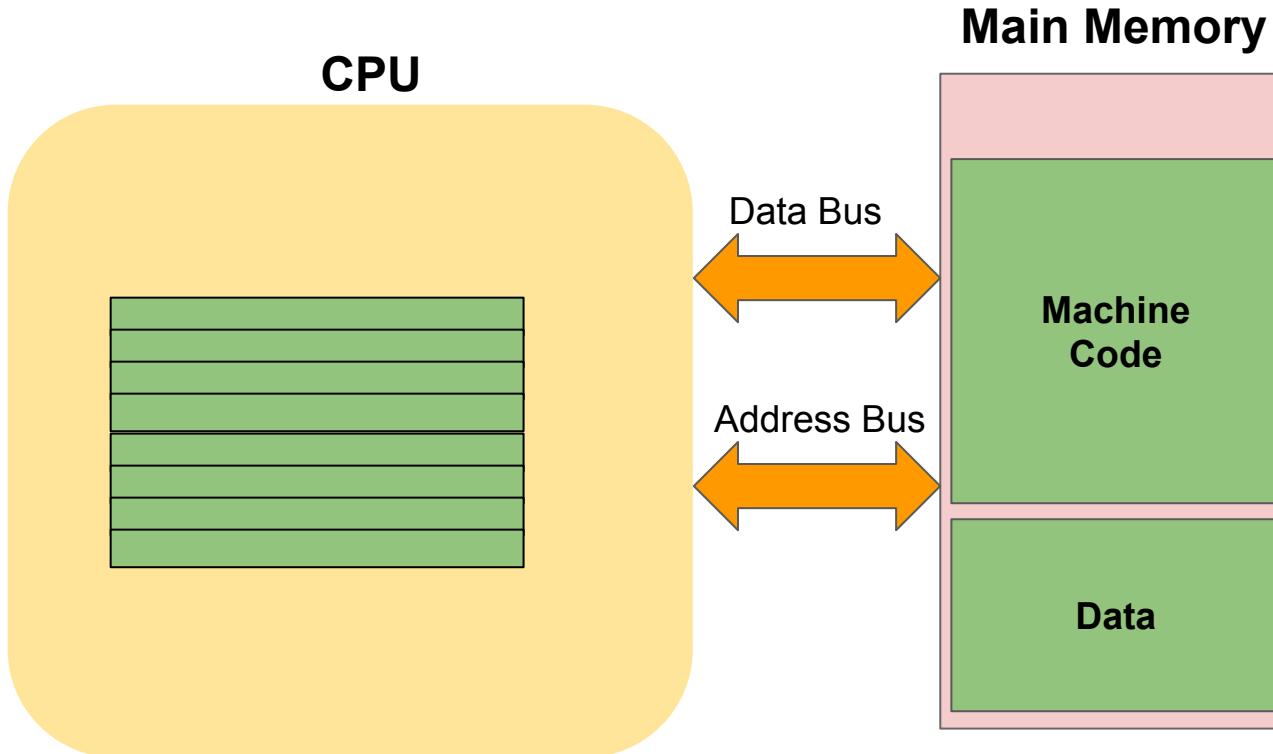
[https://en.wikipediB.org/wiki/Intel\\_8086](https://en.wikipediB.org/wiki/Intel_8086)



# CPU, Memory, instructions and data



# CPU, Memory, instructions and data

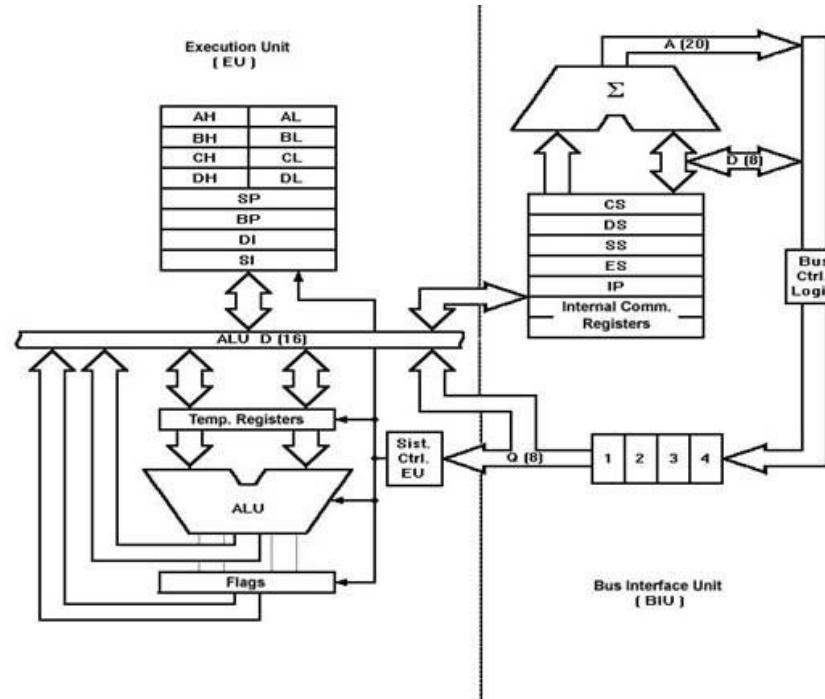




1928

K. N. Touei University of Technology

# Intel 8086 architecture



[http://www.cosc.brocku.ca/~bockusd/3p92/Local\\_Pages/8086\\_architecture.htm](http://www.cosc.brocku.ca/~bockusd/3p92/Local_Pages/8086_architecture.htm)

# Intel 8086 registers

- 16 bit registers
- 8 bit access
  - AX,BX,CX,DX
  - e.g. AX = (AH | AL)

General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

Pointer and Index Registers

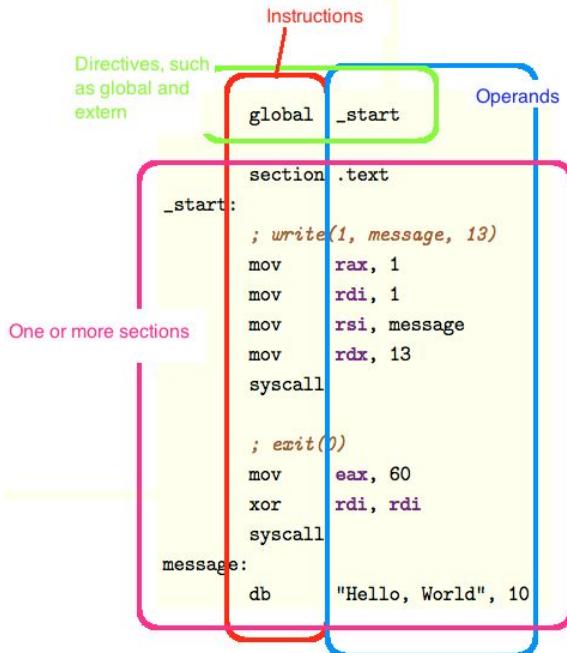
SP	Stack Pointer
BP	Base Pointer
SI	Source Index
DI	Destination Index
IP	Instruction Pointer

Segment Registers

CS	Code Segment
DS	Data Segment
SS	Stack Segment
ES	Extra Segment

Flags

# Intel 8086 assembly syntax



## General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

## Pointer and Index Registers

SP	Stack Pointer
BP	Base Pointer
SI	Source Index
DI	Destination Index
IP	Instruction Pointer

## Segment Registers

CS	Code Segment
DS	Data Segment
SS	Stack Segment
ES	Extra Segment

Flags

<http://cs.lmu.edu/~ray/notes/nasmtutorial/>

<http://bucarotechelp.com/computers/architecture/86011201.asp>

# Intel 8086 assembly commands

```
mov x,y
xchg x,y
```

```
inc x
dec x
```

```
add x,y
sub x,y
neg x
```

**; bitwise operators**

```
and x,y
or x,y
xor x,y
not x
```

**x,y: register, constant, memory (address)**

## General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

## Pointer and Index Registers

SP		Stack Pointer
BP		Base Pointer
SI		Source Index
DI		Destination Index
IP		Instruction Pointer

## Segment Registers

CS		Code Segment
DS		Data Segment
SS		Stack Segment
ES		Extra Segment

## Flags

<http://bucarotechelp.com/computers/architecture/86011201.asp>

# Intel 8086 assembly commands

```

mov x,y      ; x = y
xchg x,y    ; x <-> y

inc x        ; x++
dec x        ; x--

add x,y      ; x += y
sub x,y      ; x -= y
neg x        ; x = -x

; bitwise operators
and x,y      ; x = x & y
or x,y       ; x = x | y
xor x,y      ; x = x ^ y
not x        ; x = ~x

```

**x,y:** register, constant, memory (address)

## General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

## Pointer and Index Registers

SP		Stack Pointer
BP		Base Pointer
SI		Source Index
DI		Destination Index
IP		Instruction Pointer

## Segment Registers

CS		Code Segment
DS		Data Segment
SS		Stack Segment
ES		Extra Segment

## Flags

<http://bucarotechelp.com/computers/architecture/86011201.asp>

# Intel 8086 assembly commands

**assume u, v are memory locations (variables)**

```

mov ax, bx
mov ax, 1
mov al, ah
mov cl, 123
mov ax, [u]
mov [v], bx
mov [v], 12

mov [v], [u]

```

General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

Pointer and Index Registers

SP		Stack Pointer
BP		Base Pointer
SI		Source Index
DI		Destination Index
IP		Instruction Pointer

Segment Registers

CS		Code Segment
DS		Data Segment
SS		Stack Segment
ES		Extra Segment

Flags

# Intel 8086 assembly commands

**assume u, v are memory locations (variables)**

```

mov ax, bx
mov ax, 1
mov al, ah
mov cl, 123
mov ax, [u]
mov [v], bx
mov [v], 12
mov [v], [u]

```

## General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

## Pointer and Index Registers

SP		Stack Pointer
BP		Base Pointer
SI		Source Index
DI		Destination Index
IP		Instruction Pointer

## Segment Registers

CS		Code Segment
DS		Data Segment
SS		Stack Segment
ES		Extra Segment

## Flags

# Intel 8086 assembly commands

assume u, v are memory locations (addresses)

```
mov ax, bx
mov ax, 1
mov al, ah
mov cl, 123
mov ax, [u]
mov [v], bx
mov [v], 12
```

~~mov [v], [u]~~

```
mov ax, [u]
mov [v], ax
```

## General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

## Pointer and Index Registers

SP	Stack Pointer
BP	Base Pointer
SI	Source Index
DI	Destination Index
IP	Instruction Pointer

## Segment Registers

CS	Code Segment
DS	Data Segment
SS	Stack Segment
ES	Extra Segment

## Flags

# Intel 8086 assembly commands

**assume u, v are memory locations (variables)**

```

add ax, bx
add ax, 1
add al, ah
add cl, 123
add ax, [u]
add [v], bx
add [v], 12
add [v], [u]

```

General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

Pointer and Index Registers

SP		Stack Pointer
BP		Base Pointer
SI		Source Index
DI		Destination Index
IP		Instruction Pointer

Segment Registers

CS		Code Segment
DS		Data Segment
SS		Stack Segment
ES		Extra Segment

Flags

# Intel 8086 assembly commands

**assume u, v are memory locations (variables)**

```

add ax, bx
add ax, 1
add al, ah
add cl, 123
add ax, [u]
add [v], bx
add [v], 12
add [v], [u]

```

## General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

## Pointer and Index Registers

SP		Stack Pointer
BP		Base Pointer
SI		Source Index
DI		Destination Index
IP		Instruction Pointer

## Segment Registers

CS		Code Segment
DS		Data Segment
SS		Stack Segment
ES		Extra Segment

## Flags

# Intel 8086 assembly commands

**assume u, v are memory locations (variables)**

```
add ax, bx
add ax, 1
add al, ah
add cl, 123
add ax, [u]
add [v], bx
add [v], 12
```

~~add [v], [u]~~

```
mov ax, [u]
add [v], ax
```

## General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

## Pointer and Index Registers

SP	Stack Pointer
BP	Base Pointer
SI	Source Index
DI	Destination Index
IP	Instruction Pointer

## Segment Registers

CS	Code Segment
DS	Data Segment
SS	Stack Segment
ES	Extra Segment

## Flags

# Intel 8086 assembly commands

; bx = (ax-10)\*2

; ax = (ax+1)\*8

; ax = (ax-1)\*9

General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

Pointer and Index Registers

SP		Stack Pointer
BP		Base Pointer
SI		Source Index
DI		Destination Index
IP		Instruction Pointer

Segment Registers

CS		Code Segment
DS		Data Segment
SS		Stack Segment
ES		Extra Segment

Flags

# Intel 8086 assembly commands

```
; bx = (ax-10)*2
```

```
mov bx, ax
sub bx, 10
add bx, bx
```

```
; ax = (ax+1)*8
```

```
inc ax
add ax,ax
add ax,ax
add ax,ax
```

```
; ax = (ax-1)*9
```

```
dec ax
mov si,ax
add ax,ax
add ax,ax
add ax,ax
add ax,si
```

## General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

## Pointer and Index Registers

SP	Stack Pointer
BP	Base Pointer
SI	Source Index
DI	Destination Index
IP	Instruction Pointer

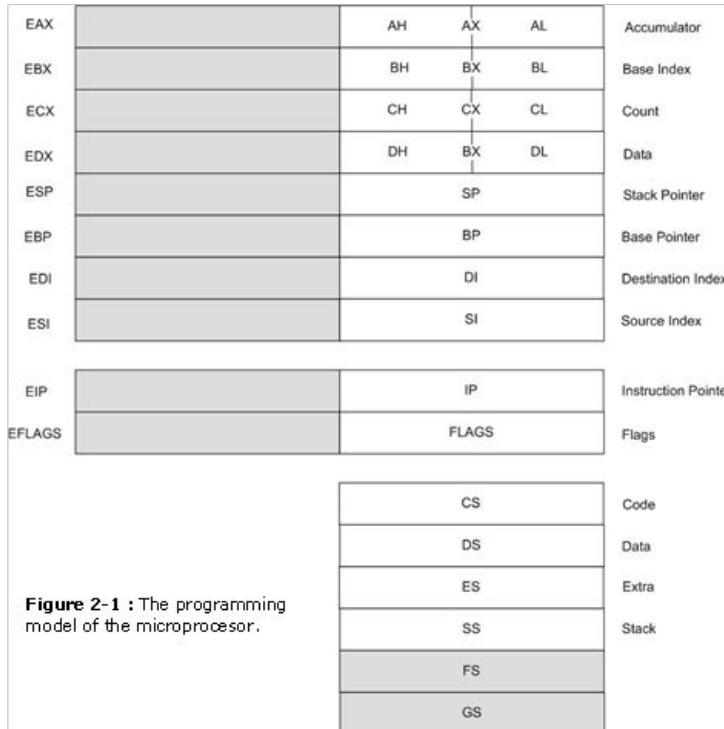
## Segment Registers

CS	Code Segment
DS	Data Segment
SS	Stack Segment
ES	Extra Segment

## Flags



# 80386 (IA-32, i386)



[http://www.byclb.com/TR/Tutorials/microprocessors/ch2\\_1.htm](http://www.byclb.com/TR/Tutorials/microprocessors/ch2_1.htm)

# 8086, 80286



1928  
Toosi University of Technology

## General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

## Pointer and Index Registers

SP		Stack Pointer
BP		Base Pointer
SI		Source Index
DI		Destination Index
IP		Instruction Pointer

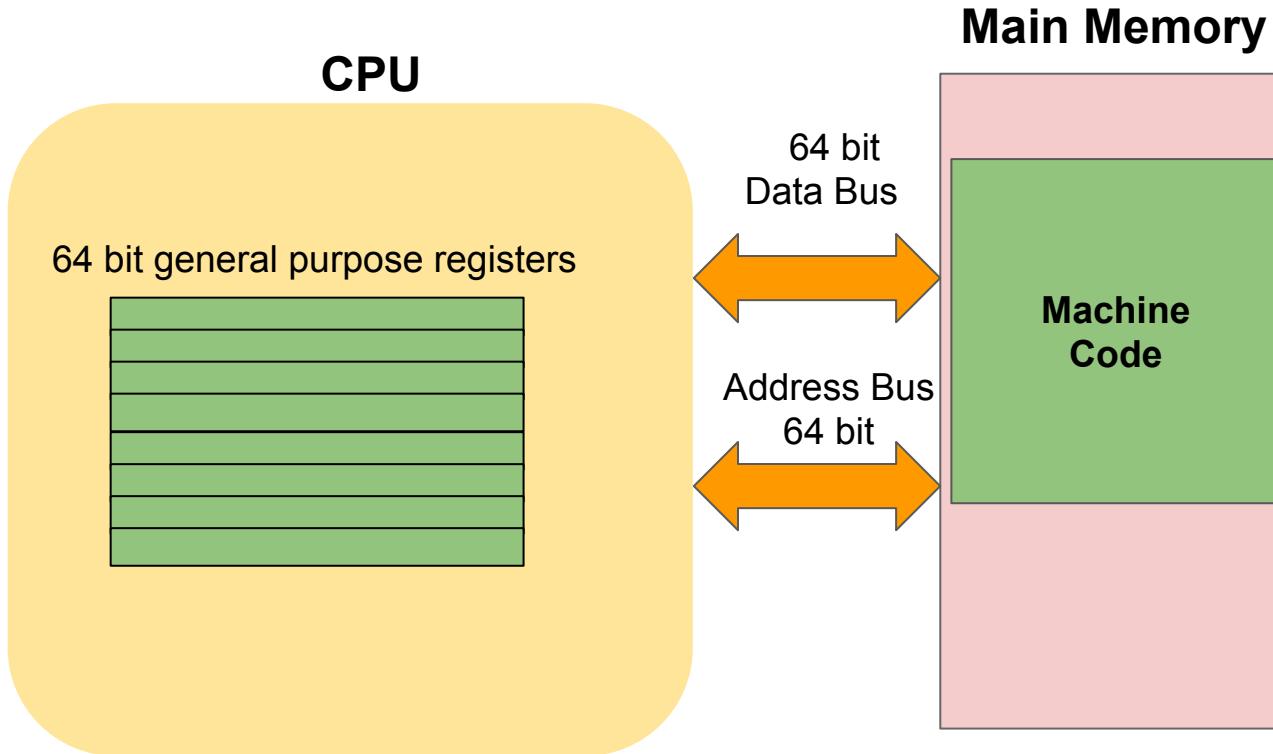
## Segment Registers

CS		Code Segment
DS		Data Segment
SS		Stack Segment
ES		Extra Segment

## Flags

<http://bucarotechelp.com/computers/architecture/86011201.asp>

# 64 bit x86 systems (x86-64)

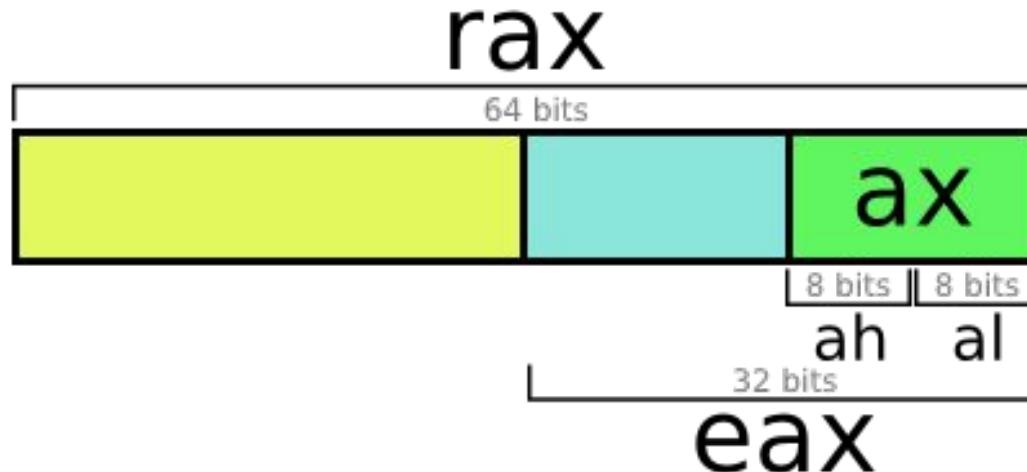




1928

K. N. Toosi University of Technology

# x86-64 bit registers



<http://nullprogram.com/blog/2015/05/15/>



1928

K. N. Toosi University of Technology

# x86-64 bit registers (x86-64, AMD64, Intel64, x64)

General-Purpose  
Registers (GPRs)

RAX	63	0
RBX		
RCX		
RDX		
RBP		
RSI		
RDI		
RSP		
R8	63	0
R9		
R10		
R11		
R12		
R13		
R14		
R15		

64-Bit Media and  
Floating-Point Registers

MMX0/FPR0	63	0
MMX1/FPR1		
MMX2/FPR2		
MMX3/FPR3		
MMX4/FPR4		
MMX5/FPR5		
MMX6/FPR6		
MMX7/FPR7		

Flags Register

0	EFLAGS	RFLAGS
63		0

Instruction Pointer

	EIP	RIP
63		0

128-Bit Media  
Registers

XMM0	127	0
XMM1		
XMM2		
XMM3		
XMM4		
XMM5		
XMM6		
XMM7		
XMM8		
XMM9		
XMM10		
XMM11		
XMM12		
XMM13		
XMM14		
XMM15		



Legacy x86 registers, supported in all modes



Register extensions, supported in 64-bit mode

Application-programming registers also include the 128-bit media control-and-status register and the x87 tag-word, control-word, and status-word registers

<https://www.viva64.com/en/a/0029/>