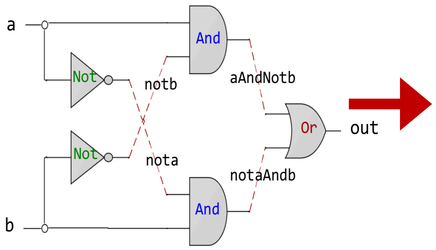
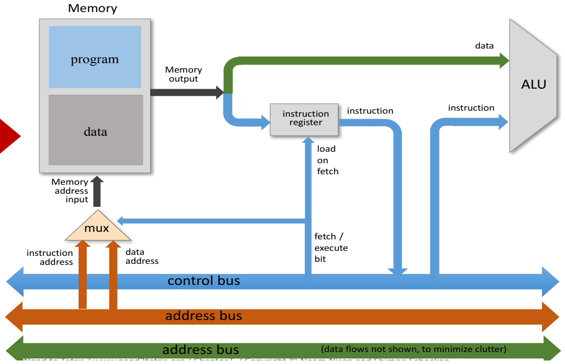
Computer Organization & Assembly Language Programming



```
CHIP Xor {
   IN a, b;
   OUT out;
   PARTS:
   Not(in=a, out=nota);
   Not(in=b, out=notb);
   And(a=nota, b=b, out=w1);
   And(a=a, b=notb, out=w2);
   Or(a=w1, b=w2, out=out);
}
```



```
@R1
D=M
@temp
M=D
```

```
0000000000000001
1111110000010000
0000000000010000
1110001100001000
```

Lecture # 28

Programming Model of x86 Architecture

```
#include<stdio.h>
#include<stdlib.h>
int main(){
  printf("Learning is fun with Arif\n");
  exit(0);
}
```

```
global main
SECTION .data
   msg: db "Learning is fun with Arif", 0Ah, 0h
   len_msg: equ $ - msg
SECTION .text
   main:
      mov rax,1
      mov rdi,1
      mov rsi,msg
      mov rdx,len_msg
      syscall
      mov rax,60
      mov rdi,0
      syscall
```

```
0:  b8 01 00 00 00
5:  bf 01 00 00 00
a:  48 be 00 00 00 00 00
11: 00 00 00
14: ba 1b 00 00 00
19: 0f 05
1b: b8 3c 00 00 00
20: bf 00 00 00 00
25: 0f 05
```

For resources visit my personal website:
https://www.arifbutt.me
and course bitbucket repository:
https://bitbucket.org/arifpucit/coal-repo

Instructor: Muhammad Arif Butt, Ph.D.

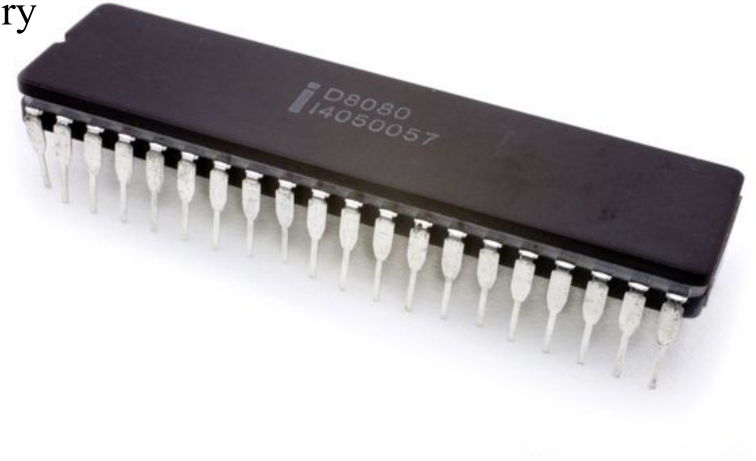
Today's Agenda

- Intel 8080
  - Memory Model
  - Register Set
- Intel 8086
  - Memory Model
  - Register Set
  - Organization
  - Limitation of Intel Segmented Memory Model
- Intel 80386
  - Memory Model
  - Register Set
- AMD and Intel x86-64
  - Memory Model
  - Register Set

Intel 8080

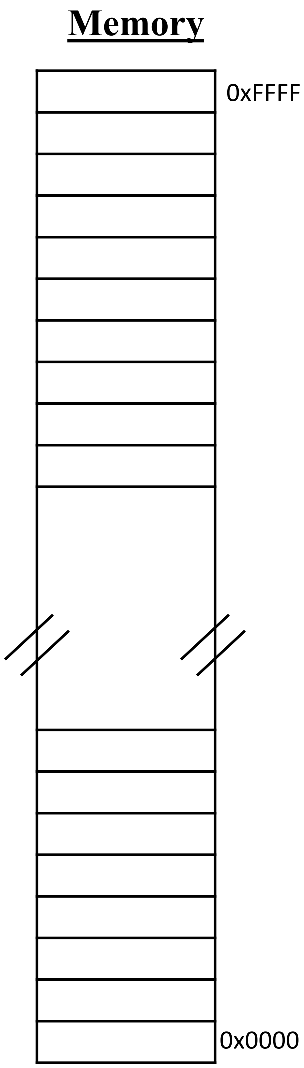
Characteristics:

- 8-bit data bus (register size)
- 16-bit address bus that could address 64 KiB of memory
- 4500 transistors
- 2 MHz
- 40-pin DIP package



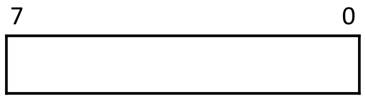
Memory Model of Intel 8080

- In 1974, Intel introduced its 8-bit **Intel-8080 CPU** with an address bus of 16 bits. The designers of Intel 8080 processor used the linear memory model to access memory and the processor could access a total memory of 64K locations using the 16 lines of the address bus
- This is called **Linear memory model**, also known as the **Flat memory model,** which refers to a memory addressing technique in which memory is organized in a single, sequential and contiguous address space
- The addressing is simple, you put a 16-bit address on the address bus and you get back the 8-bit value that is stored at that address
- It is important to note that there is no necessary relation between the number of address lines in a memory system and the size of the data stored at each location. The 8080 stored 8 bits at each location, but it could have stored 16, 32, or even 64 bits at each location, and still have 16 memory address lines



Register Set: 8080 Processor

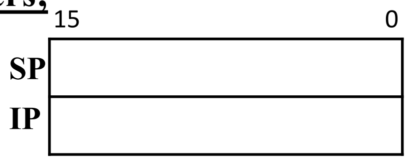
Accumulator Register:



General Registers

	15 7 0	0 7 0
BC = B+C	B	C
DE = D+E	D	E
HL = H+L	H	L

Special Registers;



Flags Register:

7	6	5	4	3	2	1	0
SF	ZF	-	AF	-	PF	-	CF

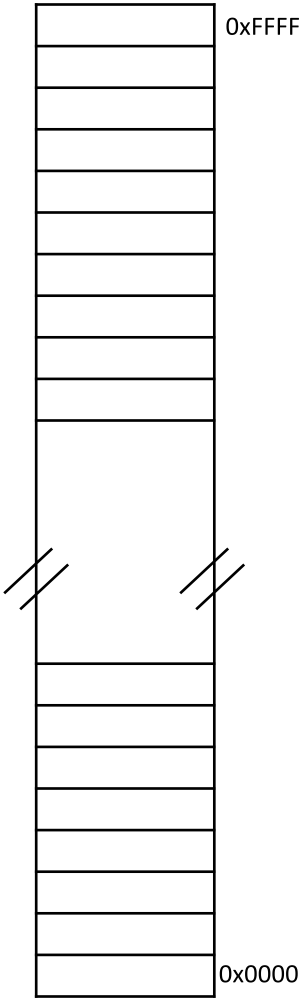
Instruction Set:

- Data moving instructions
- Arithmetic - add, subtract, increment and decrement
- Logic - AND, OR, XOR and rotate
- Control transfer - conditional, unconditional, call subroutine, return from subroutine and restarts
- Input/Output instructions

Adddressing Modes:

- Immediate
- Register
- Direct
- Register indirect

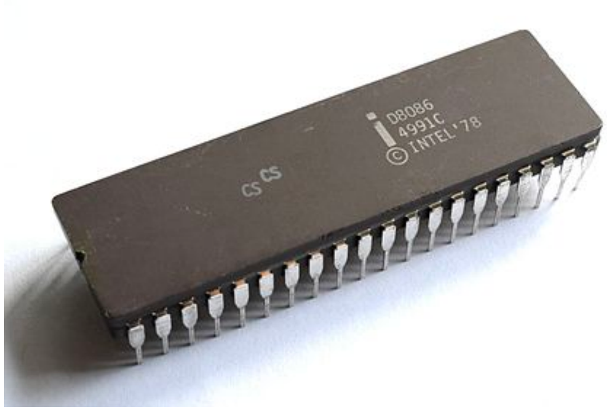
Memory

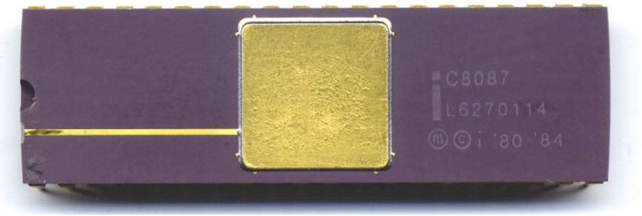


Intel 8086

Characteristics

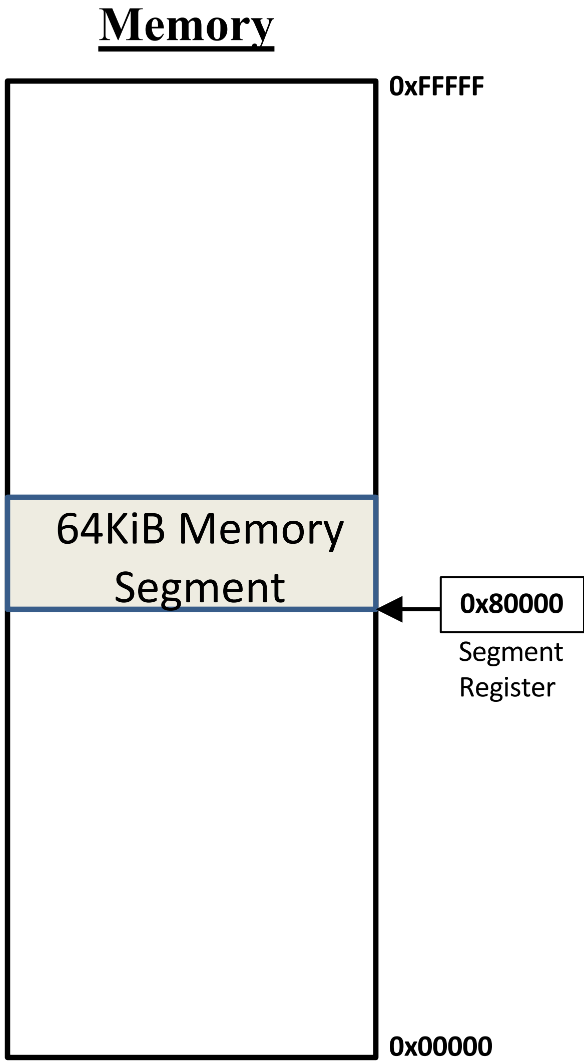
- 16-bit data bus (register size)
- 20-bit address bus that could address 1 MiB of memory
- Introduced Segmented memory model
- Separate 8087 Floating Point Unit (Math co-processor)
- Used in low cost microcontroller now





Memory Model of Intel 8086

- **Intel-8086 CPU** has an addressable memory of 1 MiB, which is 16 times more than Intel 8080
- Intel wanted to port all assembly programs running on 8080 to run on 8086 as well
- To make this porting possible, the designers of 8086 divided its memory in 64 KiB segments, so that a 8080 program could be loaded into a 64 KiB memory segment and can execute successfully
- Intel 8086 memory model is known as **Segmented memory model**, which divides the memory into groups of independent segments referenced by pointers located in special CPU registers called segment registers
- Code, data and stack can appear as three distinct units in memory



Register Set: 8086 Processor

General Purpose Registers

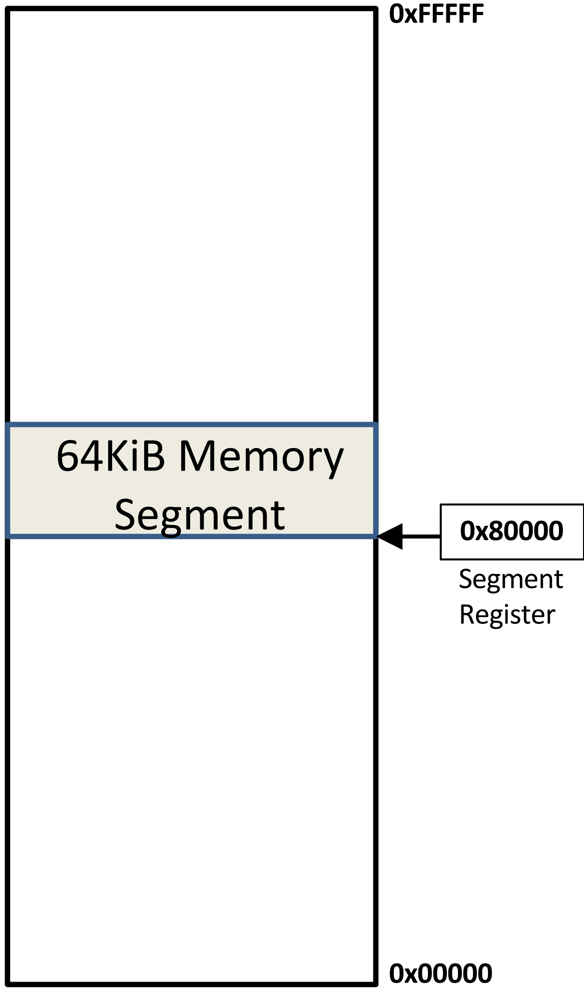
	15 7 … 0	0 7 … 0
AX = AH+AL	AH	AL
BX = BH+BL	BH	BL
CX = CH+CL	CH	CL
DX = DH+DL	DH	DL

Special Purpose Registers

	15 … 0
CS	
DS	
SS	
ES	

	15 … 0
SP	
BP	
SI	
DI	
IP	

Memory



Flags Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	OF	DF	IF	TF	SF	ZF	-	AF	-	PF	-	CF

Segmented Memory of 8086 Processor

Segment registers (DS, CS, SS, ES) hold the upper 16 bits of the starting addresses of the respective four memory segments

Example:

CS = 0x3F2A

IP = 0x1B08

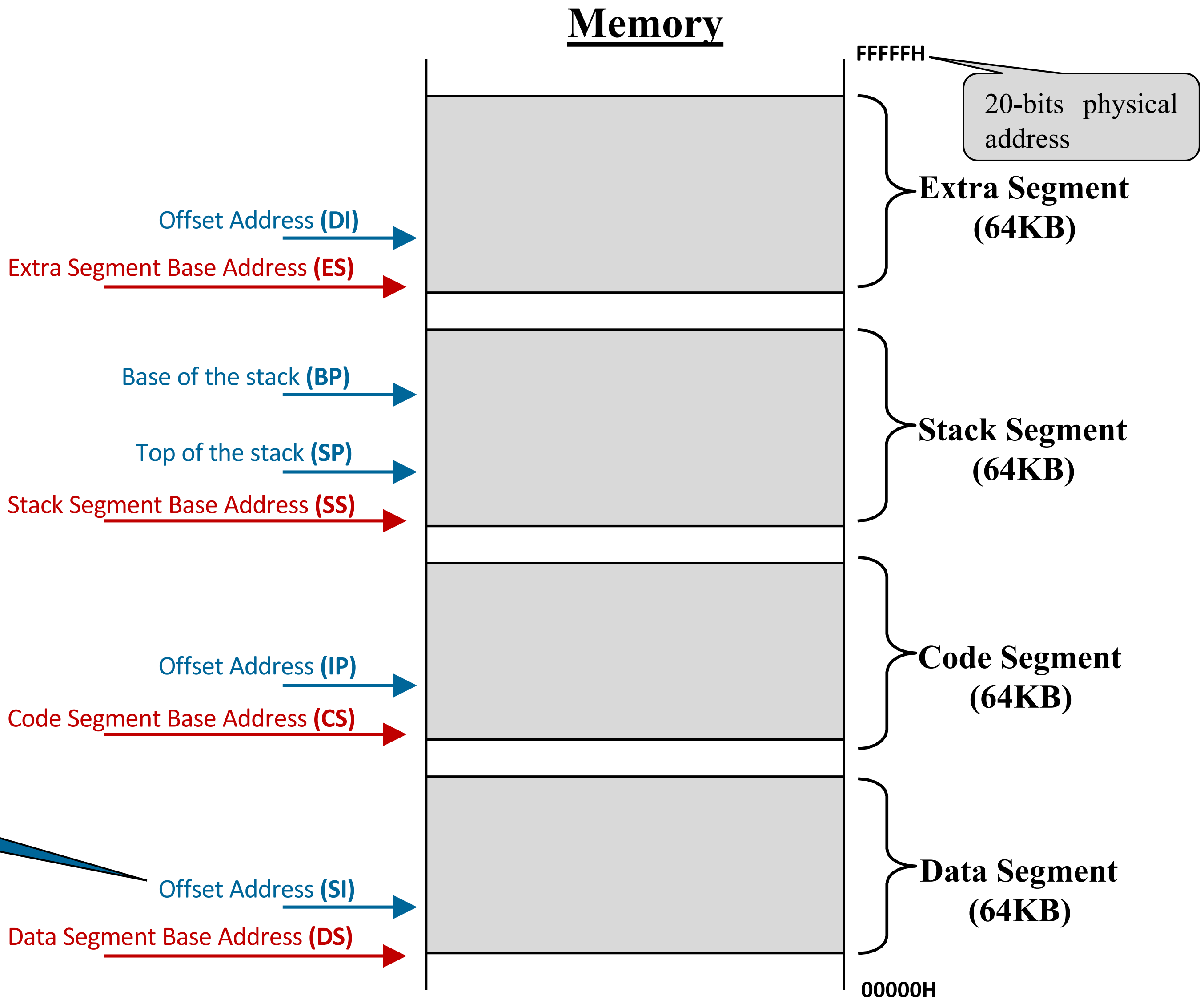
CS:IP = 3F2A:1B08

$$\mathbf{P.A = CS * 10_{16} + IP}$$

P.A = 3F2A * 10 + 1B08

P.A = 3F2A0 + 1B08 = 40DA8

Offset registers (IP, SP, BP, SI, DI) contains the 16 bits address within the respective memory segments



Memory

Segmented Memory of 8086 Processor

Example:

CS = 0x3F2A

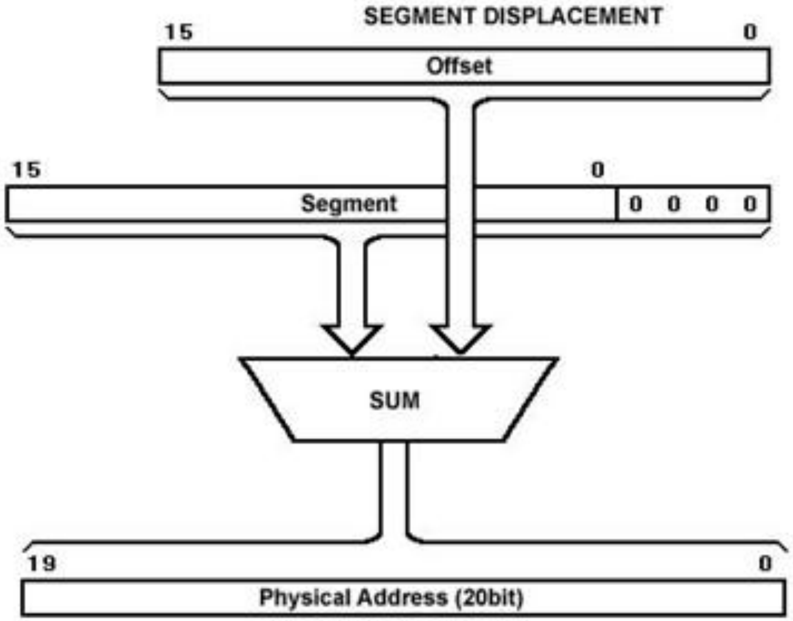
IP = 0x1B08

CS:IP = 3F2A:1B08

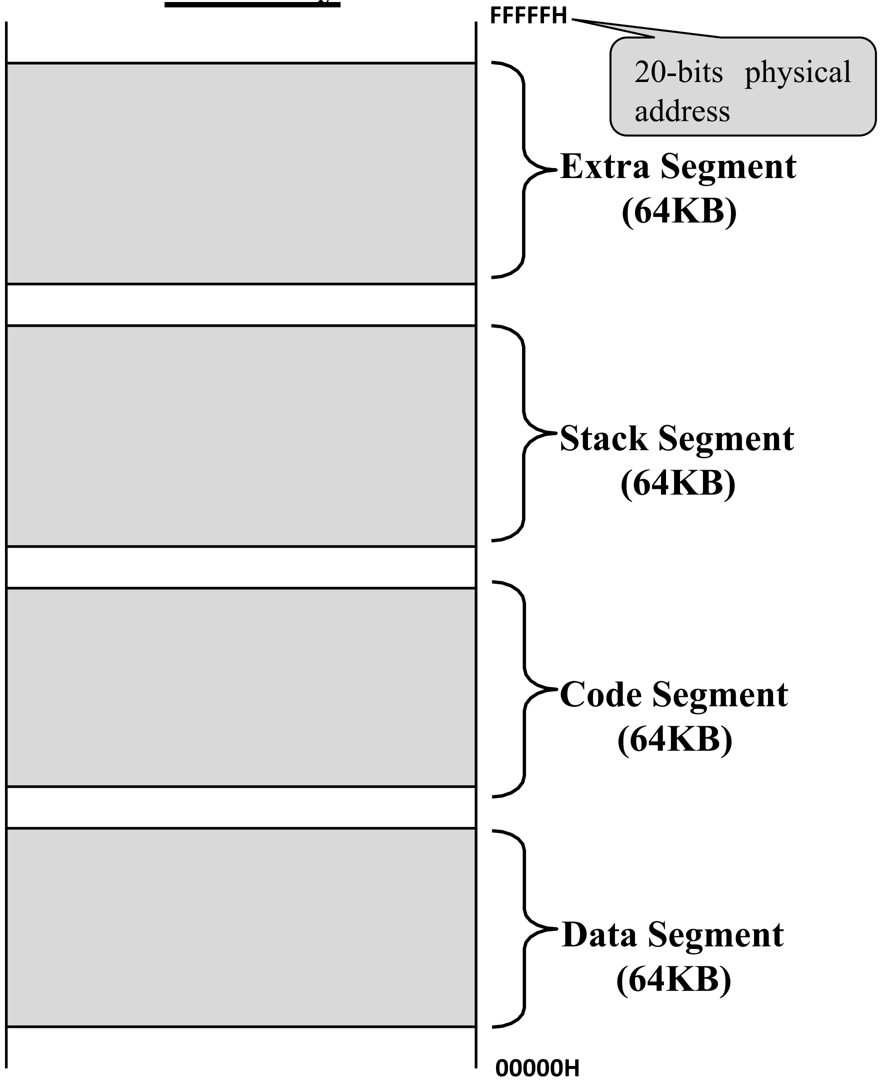
P.A = CS * 10_{16} + IP

P.A = 3F2A * 10 + 1B08

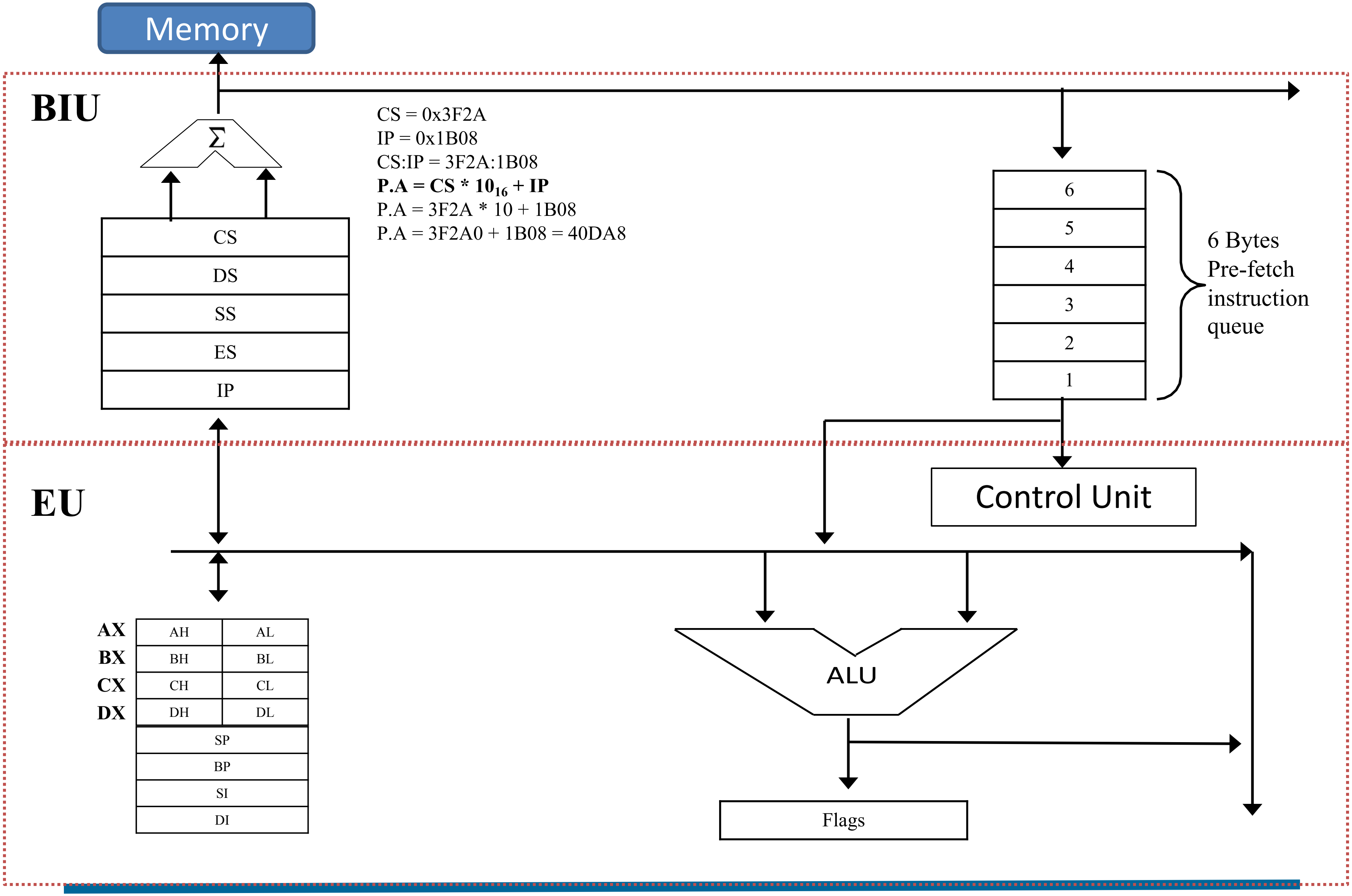
P.A = 3F2A0 + 1B08 = 40DA8



Memory



Organization of 8086 Processor



CS = 0x3F2A

IP = 0x1B08

CS:IP = 3F2A:1B08

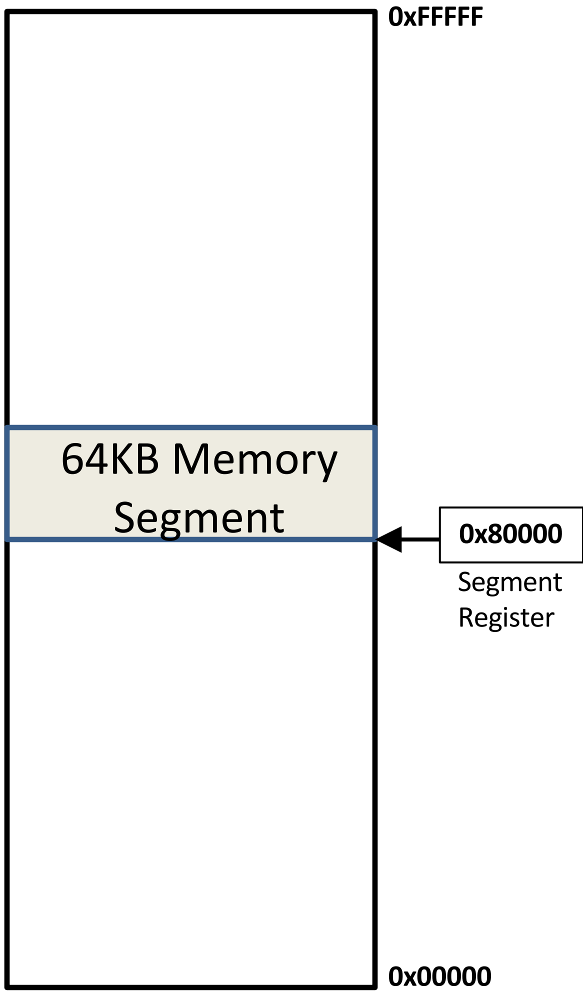
P.A = CS * 10_{16} + IP

P.A = 3F2A * 10 + 1B08

P.A = 3F2A0 + 1B08 = 40DA8

Limitation of Intel Segmented Memory Model

- The Intel 8080 assembly programmers were happy as all their programs were successfully ported to Intel 8086 machines
- The problems began after few years, when programmers who have never seen the 8080 processor started to write new programs from scratch for the 8086 processor
- Those programmers actually do not need the segmented memory model, but have to forcefully use it
- Programs that need to access large data structures, larger than than 64K of memory at a time had to use memory in 64K chunks, switching between chunks by switching values into and out of segment registers
- Hence people say that segmented memory model was a wise short-term thinking but catastrophically bad long-term thinking



Instructor: Muhammad Arif Butt, Ph.D.

Intel 80386

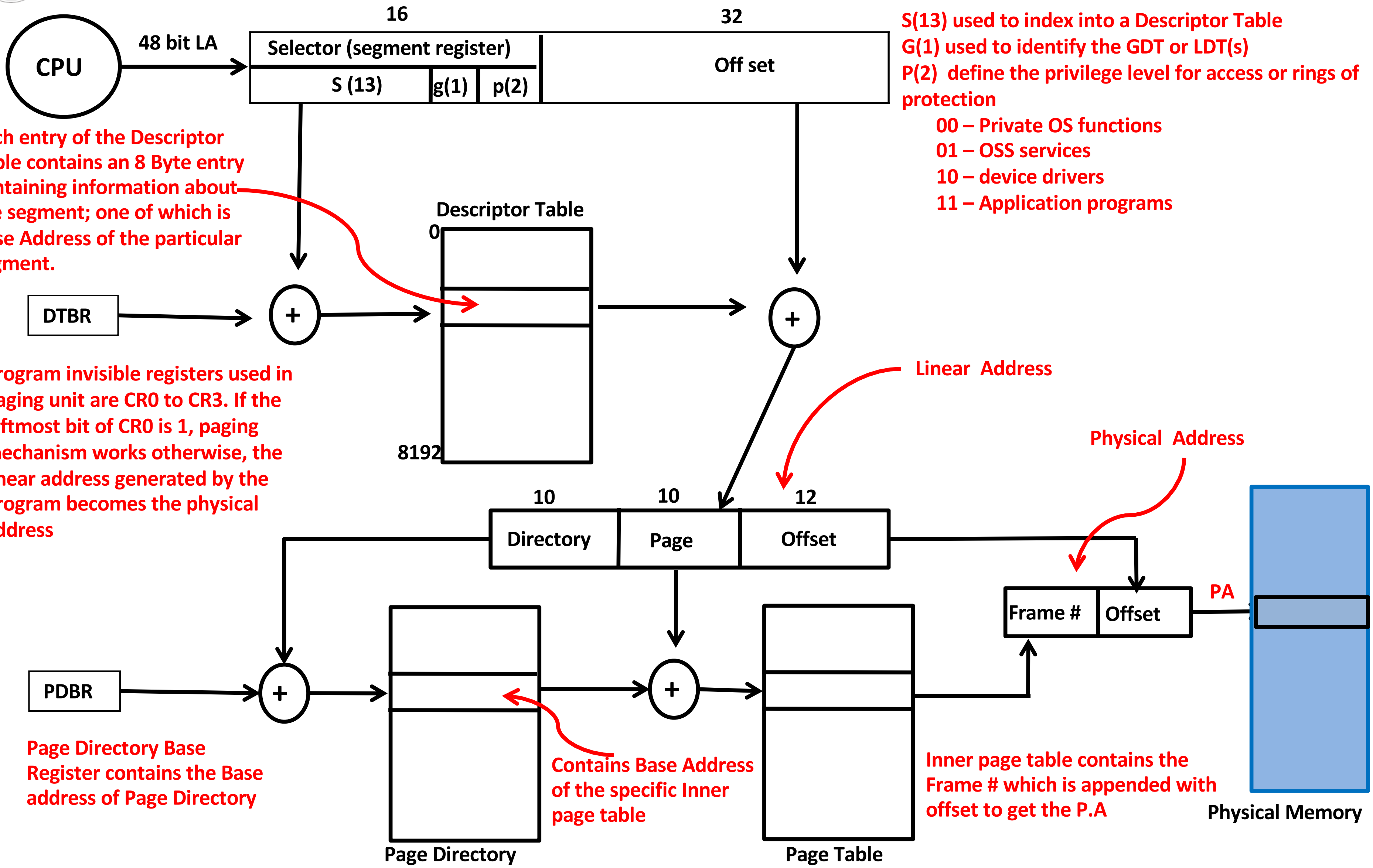
Characteristics:

- 32-bit data bus
- 32-bit address bus that could address 4 GiB of memory
- Introduced Protected memory
- Introduced the concept of paging and virtual memory

Memory Model of Intel 80386

- Intel **8086** has an address bus of 20 bits and therefore can address a physical memory of up to 1 MiB. On the contrary Intel **80386** has an address bus of 32 bits and therefore can address a physical memory of up to 4 GiB. To ensure portability once again, Intel introduced Protected mode of memory addressing in 80386 and named the older memory addressing scheme of 8086 as Real mode
- **Real Mode:**
  - In real mode, irrespective of the total available memory, only first 1 MiB of memory can be accessed
  - To translate a logical address to a physical address segment:offset (CS:IP) addressing is used (discussed on previous slides)
- **Protected Mode:**
  - This new mode of 80386 allows access to data and programs located above the first 1 MiB of memory (extended memory), as well as within the first 1 MiB of memory
  - The segment registers are now considered part of the operating system, you can neither read nor change them directly. They point to OS data structures that contain information to access a location
  - 32-bit Protected mode supports much larger data structures than Real mode

Intel 80386: L.A to P.A Translation



S(13) used to index into a Descriptor Table
G(1) used to identify the GDT or LDT(s)
P(2) define the privilege level for access or rings of protection

- 00 – Private OS functions
- 01 – OSS services
- 10 – device drivers
- 11 – Application programs

Each entry of the Descriptor Table contains an 8 Byte entry containing information about the segment; one of which is Base Address of the particular segment.

Linear Address

Physical Address

Program invisible registers used in paging unit are CR0 to CR3. If the leftmost bit of CR0 is 1, paging mechanism works otherwise, the linear address generated by the program becomes the physical address

Page Directory Base Register contains the Base address of Page Directory

Contains Base Address of the specific Inner page table

Inner page table contains the Frame # which is appended with offset to get the P.A

Register Set: 80386 Processor

General Purpose Registers

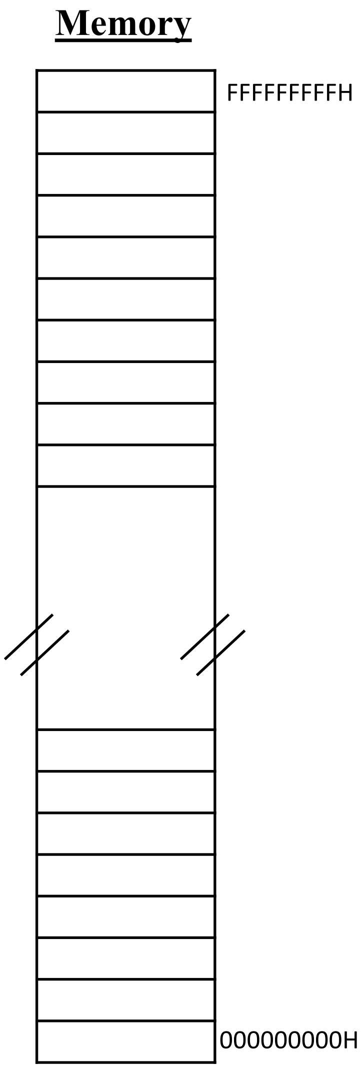
	31	15	0	
		7 0	7 0	
EAX		AH	AL	**AX** = AH+AL
EBX		BH	BL	**BX** = BH+BL
ECX		CH	CL	**CX** = CH+CL
EDX		DH	DL	**DX** = DH+DL
ESP				
EBP				
ESI				
EDI				

Instruction Pointer

EIP	

Segment Registers

	15 0
CS	
DS	
SS	
ES	
FS	
GS	



Flags Register

	31		21	20	19	18	17	16		14	13	12	11	10	9	8	7	6		4		2		0
EFLAGS	-						VM	RF	-	NT	IOP1	IOP0	OF	DF	IF	TF	SF	ZF	-	AF	-	PF	-	CF

Intel and AMD x86-64

Characteristics:

- 64 bit data/address bus and 3.8+ GHz
- 16-32 KiB on-chip data and instruction caches
- Super-scalar design with three parallel 12-stages pipeline, so can execute 3 instructions in each clock cycle
- Supports out of order execution, register renaming, improved branch prediction, and speculative instruction execution
- Pentium-III was an Intel brand/model based on 80686/P6 architecture, which introduced a new SIMD technology called Streaming SIMD Extension (**SSE**)
- Pentium-IV was an Intel brand/model based on 80686/P6 architecture, having a clock support of up to 3.8+ GHZ and support of hyper-threading technology

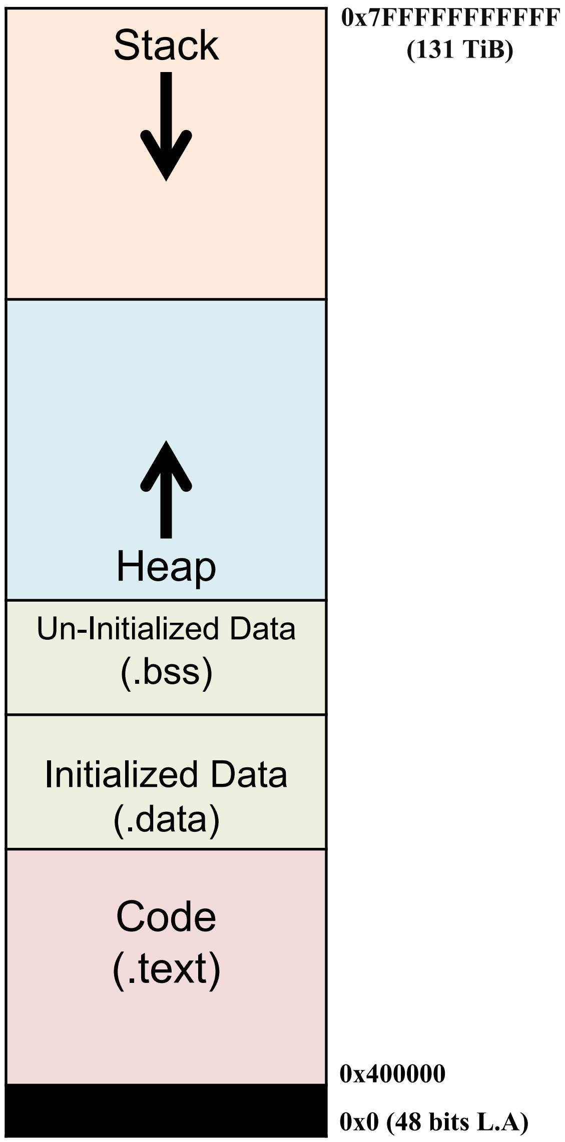
Memory Model of x86-64

- The x86-64 architecture defines three general modes: *real mode*, *protected mode*, and *long mode*
  - **Real Mode** is a compatibility mode that enables the CPU to run older real-mode operating systems and software like DOS and Windows 3.1. In real mode the x86-64 CPU works just like an 8086, and supports real mode flat model and real mode segmented model
  - **Protected Mode** is also a compatibility mode that enables the CPU to run older operating systems like Windows 2000/XP/Vista/7 and their applications. In protected mode the x86-64 CPU works just like an 80386
  - **Long Mode** is a true 64-bit mode; and when the x86-64 CPU is in long mode, all registers are 64 bits wide, and all machine instructions that act on 64-bit operands are available. All 80386 registers are available, rather extended to 64 bits in width

Memory Model of x86-64

- The layout of various segments of a process running on a Linux system is shown
- The x86-64 CPU chips that you can buy today implement 48 bit logical address for virtual memory (as shown), and 40 bits for physical memory
- The 64 bit Logical address can be broken down as:

63 -48	47 39	38 -30	29 - 21	20 - 12	11 - 0
Unused	PML4 index	Page directory pointer index	Page directory index	Page table index	Page offset



Register Set: x86-64 Processor

General Purpose Registers

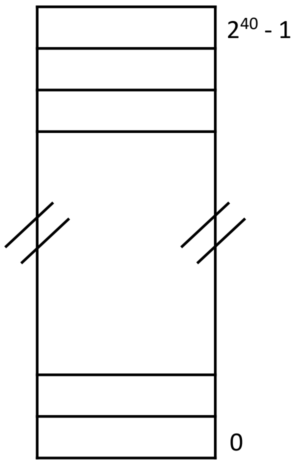
64-bit register	Lowest 32-bits	Lowest 16-bits	Lowest 8-bits
r0/rax	eax	ax	al
r1/rbx	ebx	bx	bl
r2/rcx	ecx	cx	cl
r3/rdx	edx	dx	dl
r4/rsi	esi	si	sil
r5/rdi	edi	di	dil
r6/rbp	ebp	bp	bpl
r7/rsp	esp	sp	spl
r8	r8d	r8w	r8b
r9	r8d	r9w	r9b
r10	r10d	r10w	r10b
r11	r11d	r11w	r11b
r12	r12d	r12w	r12b
r13	r13d	r13w	r13b
r14	r14d	r14w	r14b
r15	r15d	r15w	r15b

SSE Media Registers

255 127 0

ymm0	xmm0
ymm1	xmm1
ymm2	xmm2
ymm3	xmm3
ymm14	xmm14
ymm15	xmm15

Memory



63 0

RIP	EIP

Segment Registers

15 0

CS
DS
SS
ES
FS
GS

RFLAGS

63		21	20	19	18	17	16		14	13	12	11	10	9	8	7	6		4		2		0
-		ID	VIP	VIF	AC	VM	RF	-	NT	IOP1	IOP0	OF	DF	IF	TF	SF	ZF	-	AF	-	PF	-	CF

Things To Do

Coming to office hours does NOT mean you are academically week!