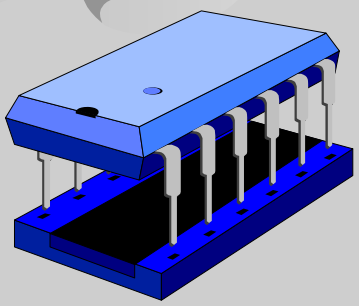
Saul Coval Computers

אסמבלי 8086

Assembler 8086

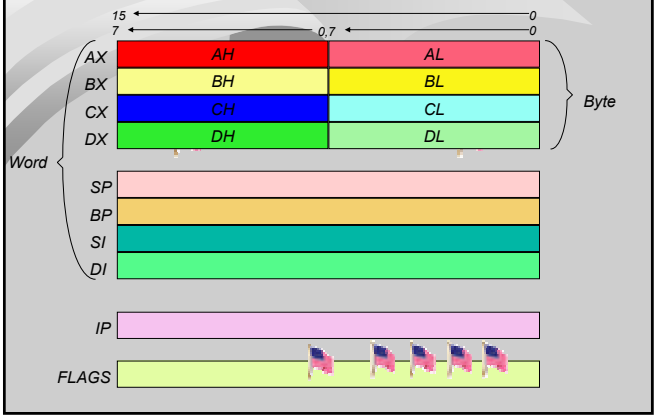
Saul Coval Computers

8086 המעבד



Saul Coval Computers

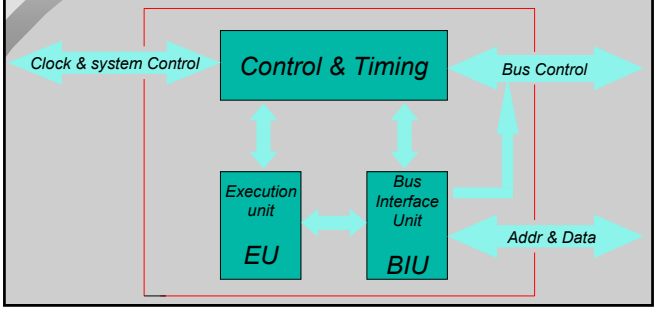
רגיסטרים פנימיים



Saul Coval Computers

8086 מבנה מעבד

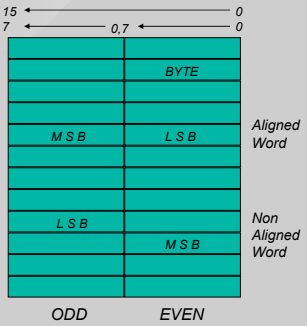
- *Two independent units - EU, BIU* ■
- *Overlapping instruction fetches with execution* ■



Saul Coval Computers

מבנה הזיכרון

- *Memory organized in two banks - ODD and EVEN* ■
- *BYTE access in either bank* ■
- *Aligned word accessed in 1 cycle* ■
- *Non Aligned word accessed in 2 cycles* ■



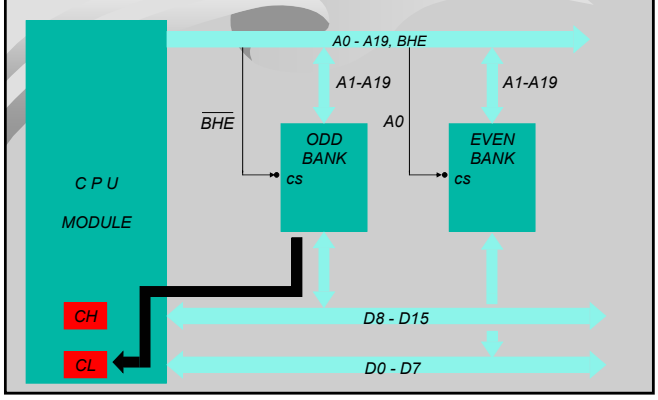
Saul Coval Computers

AX (16 bit)	AH (8 bit)	AL (8 bit)	Accumulator	צובר - אקומולטור
BX (16 bit)	BH (8 bit)	BL (8 bit)	Base	מצביע בסיס לאזור הנתונים וגם אוגר כללי
CX (16 bit)	CH (8 bit)	CL (8 bit)	Count	אוגר מונה וגם אוגר כללי
DX (16 bit)	DH (8 bit)	DL (8 bit)	Data	מצביע I/O גם ערך לכפל / חילוק , וכללי
	SP (16 bit)		Stack Pointer	מצביע מחסנית
	BP (16 bit)		Base Pointer	מצביע בסיס לאזור המחסנית
	Si (16 bit)		Source Index	מצביע לתוכן בסיס
	Di (16 bit)		Destination Index	מצביע לתוכן יעד
	IP (16 bit)		Instruction Pointer	מצביע פקודות
	FR (16 bit)		Flag Register status	אוגר דגלים מצב
	CS (16 bit)		Code Segment	מצביע למקטע התוכנית
	DS (16 bit)		Data Segment	מצביע למקטע הנתונים
	SS (16 bit)		Stack Segment	מצביע למקטע המחסנית
	ES (16 bit)		Extra Segment	מצביע למקטע הנוסף
	IR (16 bit)		Instruction Register	שומר בצופן של הפקודה לפענוח
	CR (16 bit)		Control Register	מפקח לבקרה ותנועת הנתונים ב CPU

Saul Coval Computers

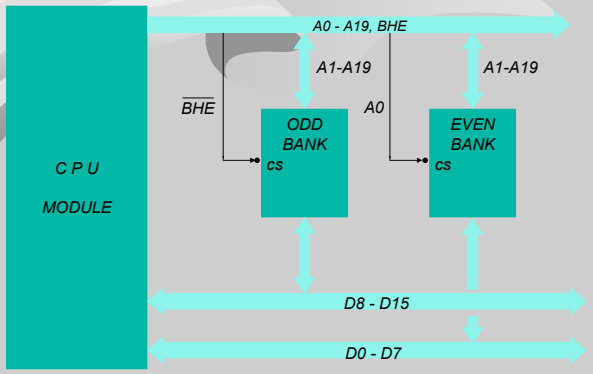
עבודה עם בתים בודדים

The CPU will swap bytes between the odd and even banks if necessary



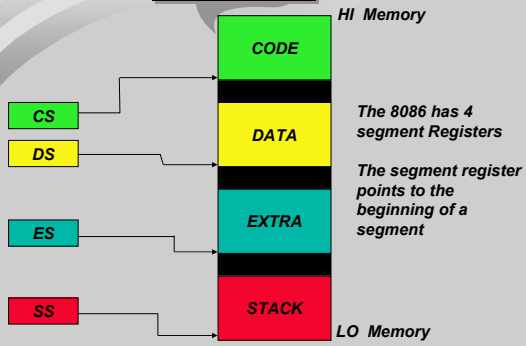
Saul Coval Computers

מבנה מערכת זיכרון טיפוסית



Saul Coval Computers

סגמנטציה



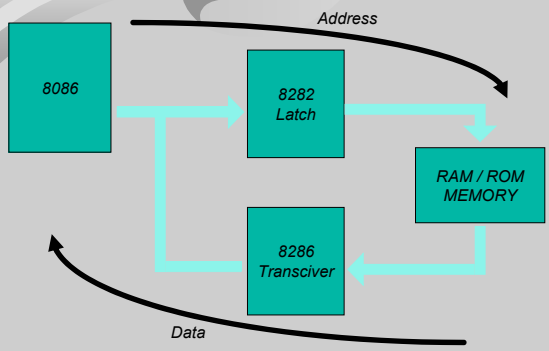
The 8086 has 4 segment Registers

The segment register points to the beginning of a segment

Saul Coval Computers

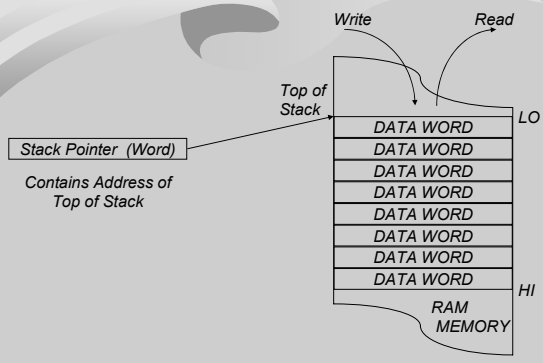
Wait States

You must consider the delays encountered by Address and Data on a memory cycle. The memory system Adds wait-states (bus cycles beyond the min. for memory cycle)



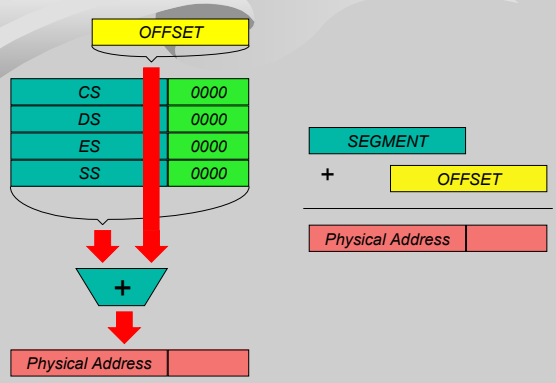
Saul Coval Computers

מחסנית



Saul Coval Computers

גישה לזיכרון בסגמנט



Saul Coval Computers

Addressing Modes

MOV AX,MVAR	:Direct	:Offset=Variable Name ■
MOV AX,[BX]	:InDirect	:Offset=[BX] ■
MOV AX,MVAR[SI]	:Indexed	:Offset=Var. name+[SI] ■
MOV AX,[BX]+5	:Based	:Offset=[BX]+[Displ.] ■
MOV AX,[BX][DI]	:Based Indexed	:Offset=[BX]+[DI] ■
MOV AX,[BP][SI]+15	:Based Indexed and Displacement	:Offset=[BP][SI]+ Dsipl. ■

[BX] [SI]
Offset=[Var. name]+ [BP] + [DI] + [Displacement]

Saul Coval Computers

פקודות העברת מידע:

MOV AX,BX ■
XCHG BL,BH ■
IN AL,Port# ■
OUT Port#,AX ■
PUSH ES ■
POP DX ■

Saul Coval Computers

פקודות אריתמטיות - לוגיות (המשך)

MUL BL / MUL DX ■
DIV Source ■
AND / OR ■
NOT ■
XOR ■
TEST ■

Saul Coval Computers

פקודות אריתמטיות - לוגיות

ADD Destination,Source ■
SUB Destination,Source ■
INC Destination ■
DEC Destination ■
NEG Destination ■
CMP Destination,Source ■

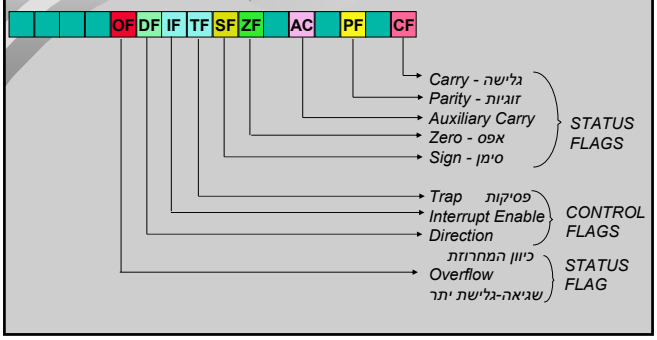
Saul Coval Computers

פקודות קפיצה

JMP Short ■
JMP Near ■
JMP Far ■
JMP ■

Saul Coval Computers

Flag Register – רגיסטר דגלים



פקודות קפיצה מותנות (המשך)

SIGNED:

- JL OR JNGE ■
- JLE OR JNG ■
- JNL OR JGE ■
- JNLE OR JG ■
- JO ■
- JS ■
- JNO ■
- JNS ■

UNSIGNED:

- JB OR JNAE OR JC ■
- JBE OR JNA ■
- JNB OR JAE OR JNC ■
- JNBE OR JA ■

פקודות קפיצה מותנות

Signed & Unsigned :

- JE OR JZ ■
- JP OR JPE ■
- JNE OR JNZ ■
- JNP OR JPO ■
- JCXZ ■

פסיקות

Automatic upon detecting Interrupt:
- Current instruction finishes execution ■
- Vector nubmer fetched from 8259A ■
- Push Flags ■
- Clear IF and TF ■
- Transfer control to service routine ■

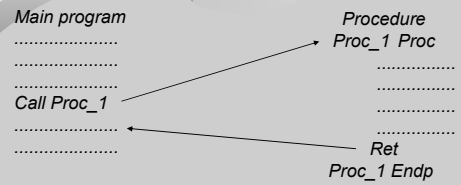
Service routine

Iret:
- Return ■
- Pop Flags ■

Interrupt processing time - 61 clocks (not including ■ current instruction, saving register data and wait-states)

פרוצדורות

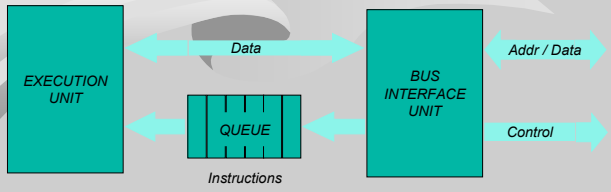
מודולי תוכנה עצמאיים שניתן לקרוא להם והם חוזרים למקום הקריאה



The Call instruction writes the return address (Next addr) into the stack

The Ret instruction reads the return address from the stack and jumps to it

Instruction Prefetch Queue



- Invisible to users ■
- Instruction Queue is 6 bytes in 8086, 4 bytes in 8088 ■
- Data access has priority over Instruction fetches ■
- Queue "flushes" automatically on JMP ■

וקטור הפסיקות

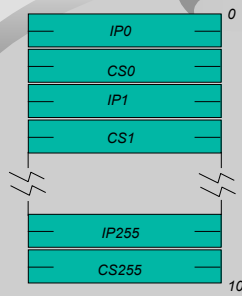


Table starts at absolute address 0

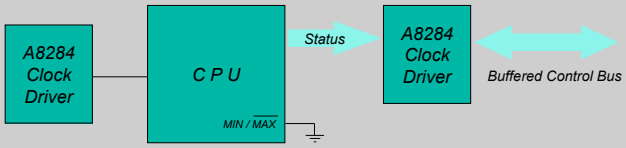
Each interrup has two words -
Instruction pointer
Code segment

Int 0: Divide error
Int 1: Single step
Int 2: NMI
Int 3: Breakpoint trap
Int 4: Overflow trap
Int5-31: Reserved by Intel

Saul Coval Computers

מערכת מקסימלית(MAX)

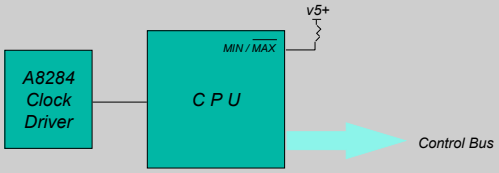
- Maximum mode designed for large systems
- Bus controller decodes status signals to 8288 generate control signals
- CPU Uses control pins freed by 8288 to coordinate other processors



Saul Coval Computers

מערכת מינימלית(MIN)

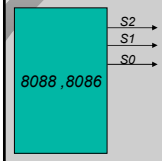
- Minimum mode designed for small systems
- Control signals to Memory and I/O supplied directly by the CPU
- Used in single processor systems only



Saul Coval Computers

תאור קווים S0,S1,S2

Status lines that inform the 8288 (Bus controller) of the type of Bus cycle that the CPU is running



S2	S1	S0	SIGNAL
0	0	0	INTA
0	0	1	I/O READ
0	1	0	I/O WRITE
0	1	1	HALT
1	0	0	CODE ACCESS
1	0	1	READ MEMORY
1	1	0	WRITE MEMORY
1	1	1	PASSIVE

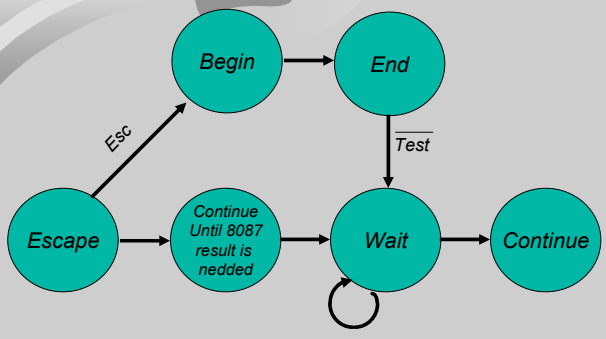
Saul Coval Computers

MIN / MAX

- $\overline{WR}$
- $\overline{INTA}$
- ALE
- M / $\overline{IO}$
- DT / $\overline{R}$
- $\overline{DEN}$
- HLDA
- HOLD
- LOCK
- QS1
- QS0
- S0
- S1
- S2
- RQ/GT0
- RQ/GT1

Saul Coval Computers

עבודה עם מעבד מתמטי 8087



Saul Coval Computers

Test Pin

- $\overline{TEST}$ is used by WAIT instruction
  - if $\overline{TEST}$ pin is low, execute continues –
  - if $\overline{TEST}$ pin is high, CPU enters an idle state –
- Used to coordinate work with 8087 coprocessor