

# Αρχιτεκτονικές Σύγχρονων Επεξεργαστών

## Αρχιτεκτονικές CPU για μικροϋπολογιστές

### **-x86**

Intel's IA-32 architecture, also called x86-32  
x86-64 with AMD's AMD64 and Intel's Intel 64 version of it

### **ARM**

ARM (32-bit) (previously Advanced RISC Machines' ARM, originally Acorn's RISC Machine) and Strong ARM/XScale architectures ARM64 (64/32-bit)

## **Αρχιτεκτονικές CPU για server και workstation**

**-x86 Xeon**

**Power Architecture** (formerly POWER and PowerPC)

**ARM64** (64/32 bit)

## **Αρχιτεκτονικές CPU για embedded systems**

**Atmel AVR** architecture

**ARM**

ARM architecture (32-bit)

ARM64 (64/32-bit)

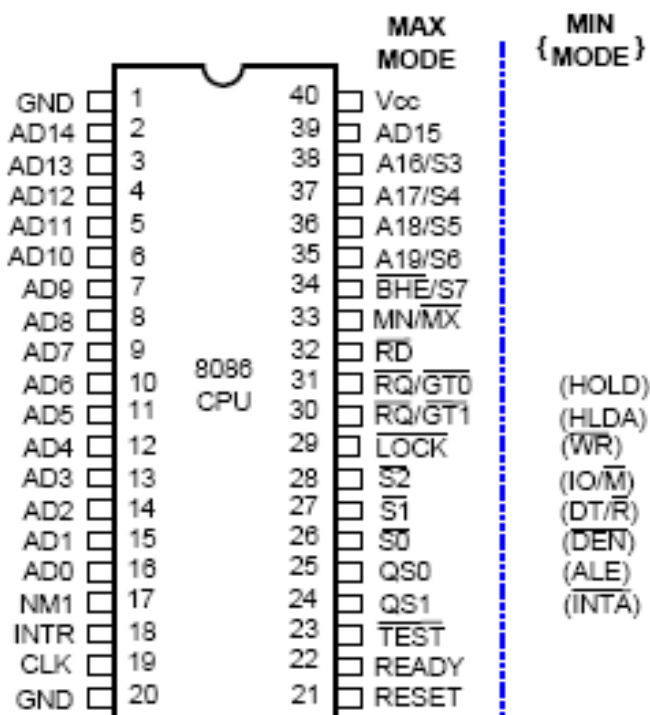
**Microchip's PIC** architecture

**Intel's 8051** architecture

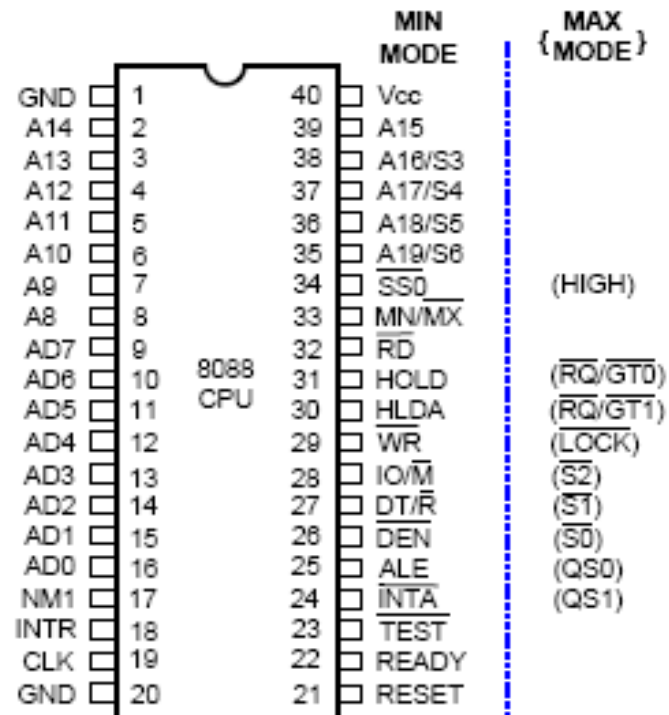
## Επεξεργαστές που χρησιμοποιήθηκαν στα PC

Στο πρώτο PC χρησιμοποιήθηκε ο επεξεργαστής 8088 που ήταν των 16 bit. Στο AT χρησιμοποιήθηκε ο επεξεργαστής 80286 που ήταν επίσης των 16 bit, αλλά είχε εξελιγμένα χαρακτηριστικά. Στην συνέχεια χρησιμοποιήθηκαν πιο ισχυροί επεξεργαστές όπως οι 80386, 80486, Pentium, Pentium II, Pentium III, Pentium 4 οι οποίοι ήταν των 32 bit. Στα σημερινά PC χρησιμοποιούνται ισχυροί **πολύ-πύρρηνοι** επεξεργαστές (Core 2, Core i3, i5, i7, i9) των 32/64 bit.

## Επεξεργαστές 8086, 8088



8086 pin diagram



8088 pin diagram

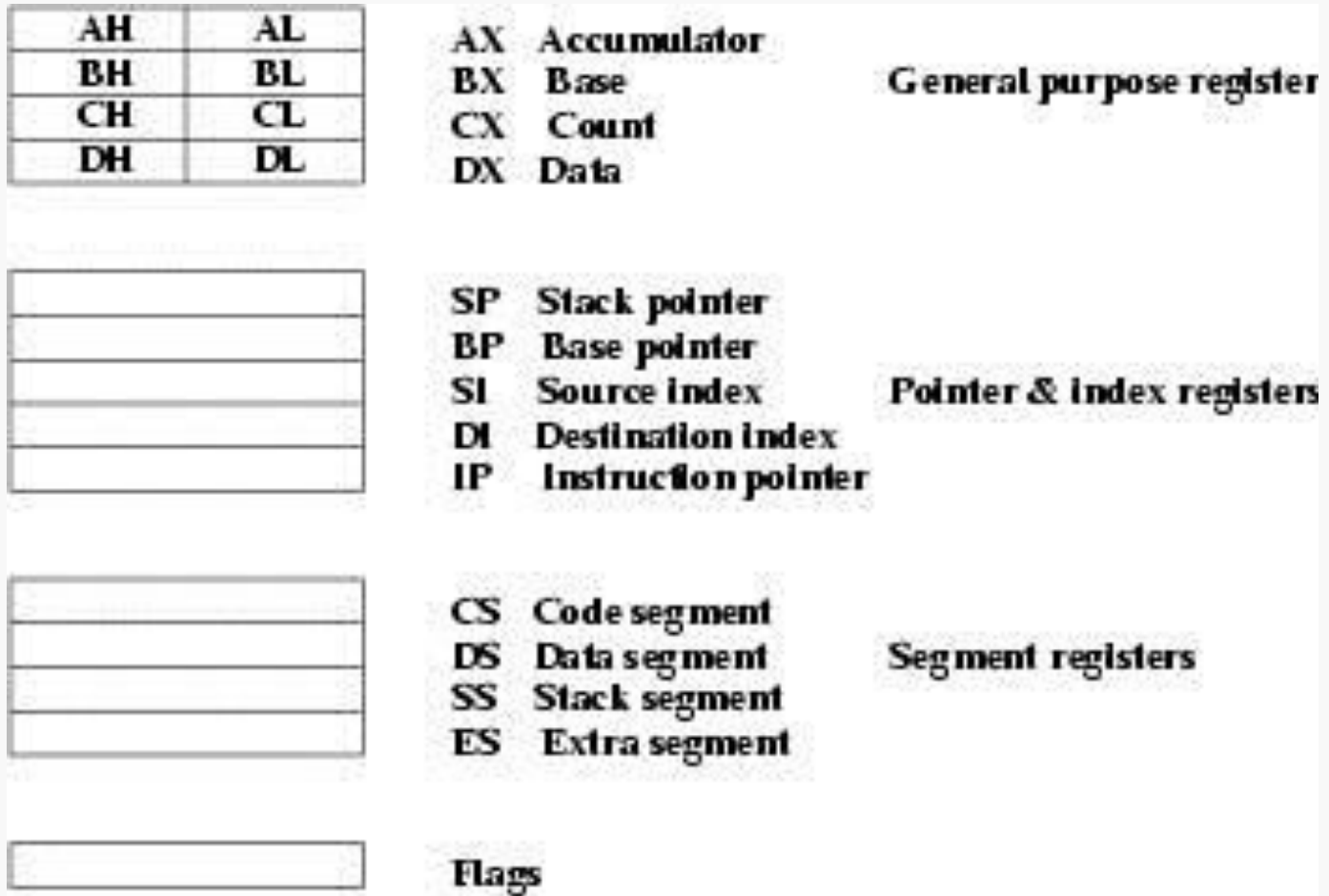
Οι επεξεργαστές 8086 και 8088 έχουν πολυπλεγμένους διαύλους διευθύνσεων και δεδομένων. Δηλαδή, σε μία προσπέλαση μνήμης η ίδιες ακίδες χρησιμοποιούνται αρχικά σαν γραμμές διεύθυνσεων και στην συνέχεια σαν γραμμές δεδομένων. Όταν στις ακίδες υπάρχει διεύθυνση ταυτόχρονα είναι ενεργοποιημένο και το σήμα ALE.

Δίαυλος διευθύνσεων 20 bit

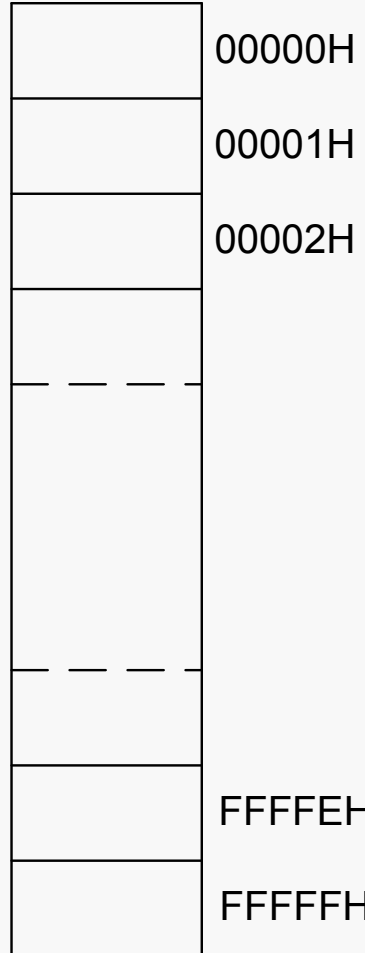
Δίαυλος δεδομένων 8086 16 bit

Δίαυλος δεδομένων 8088 8 bit

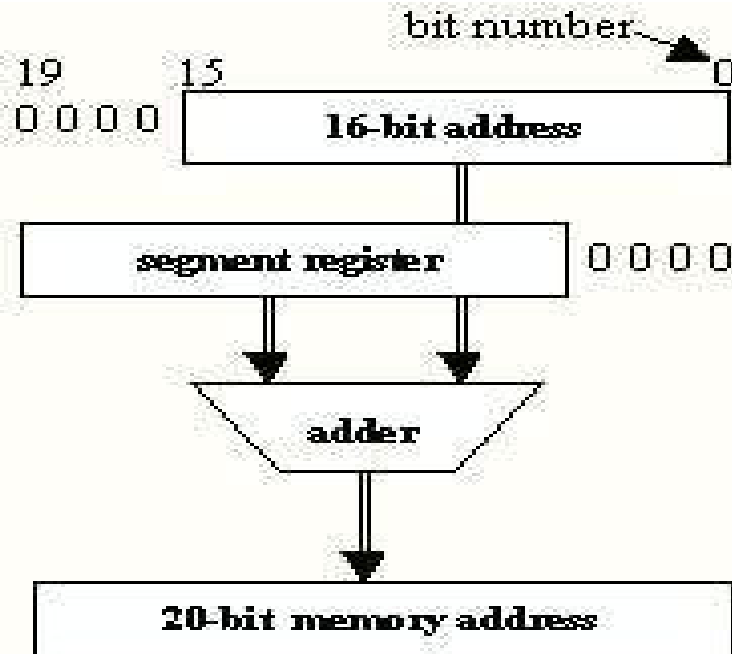
# Μοντέλο προγραμματισμού των επεξεργαστών 8086/88



Μοντέλο προγραμματισμού της μνήμης του επεξεργαστή 8088.



Για να σχηματισθούν διευθύνσεις των 20 bit γίνεται εσωτερικά στον επεξεργαστή 8086 πρόσθεση ενός segment register και ενός pointer register όπως στην συνέχεια.



Example:

address = 1234h

seg reg = 5678h

01234h

+56780h

-----

20-bit address = 579B4h

## ENTOΛH ADD

```
REG, memory
memory, REG
ADD REG, REG
memory, immediate
REG, immediate
```

Algorithm:

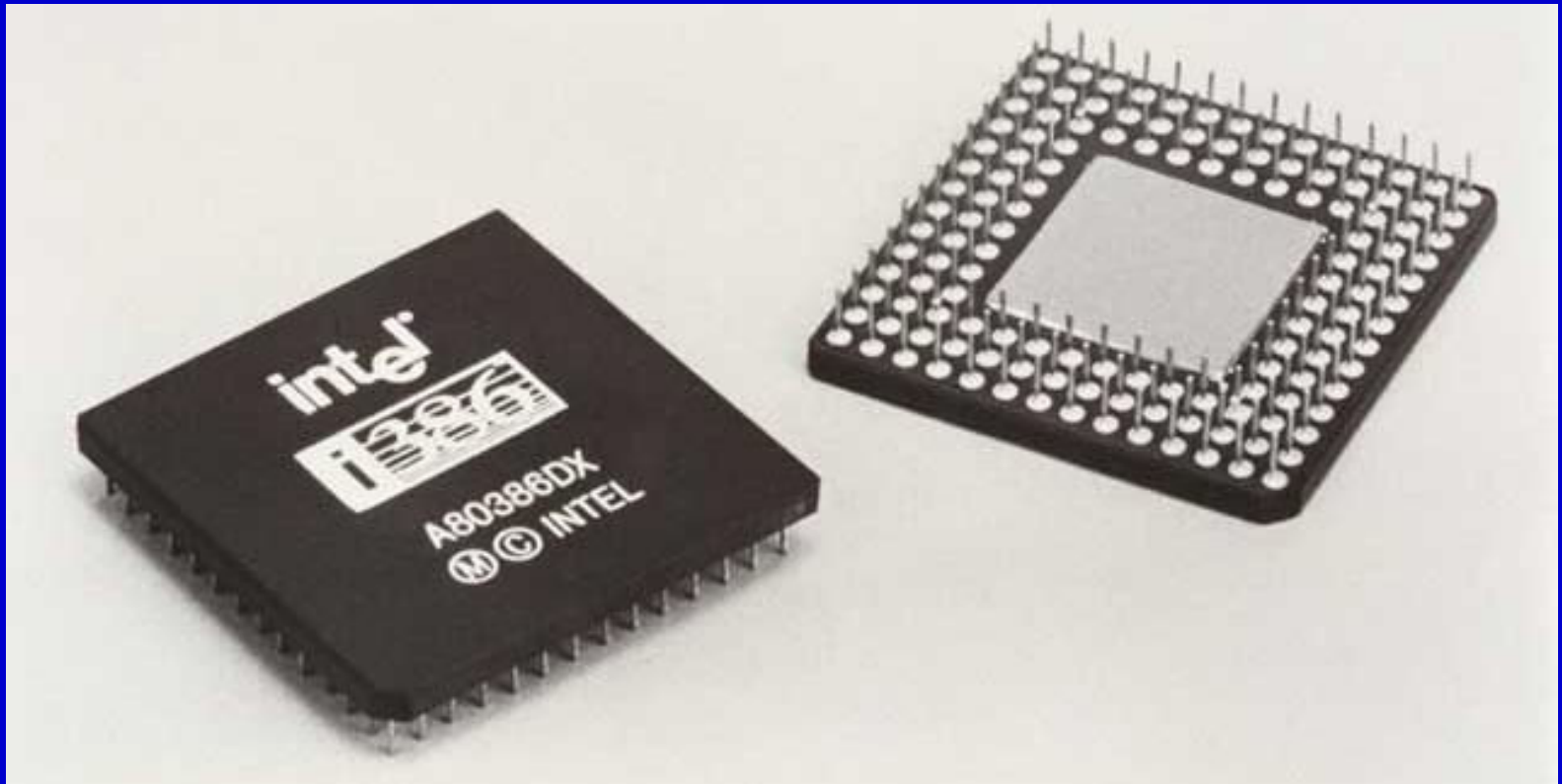
operand1 = operand1 + operand2

Example:

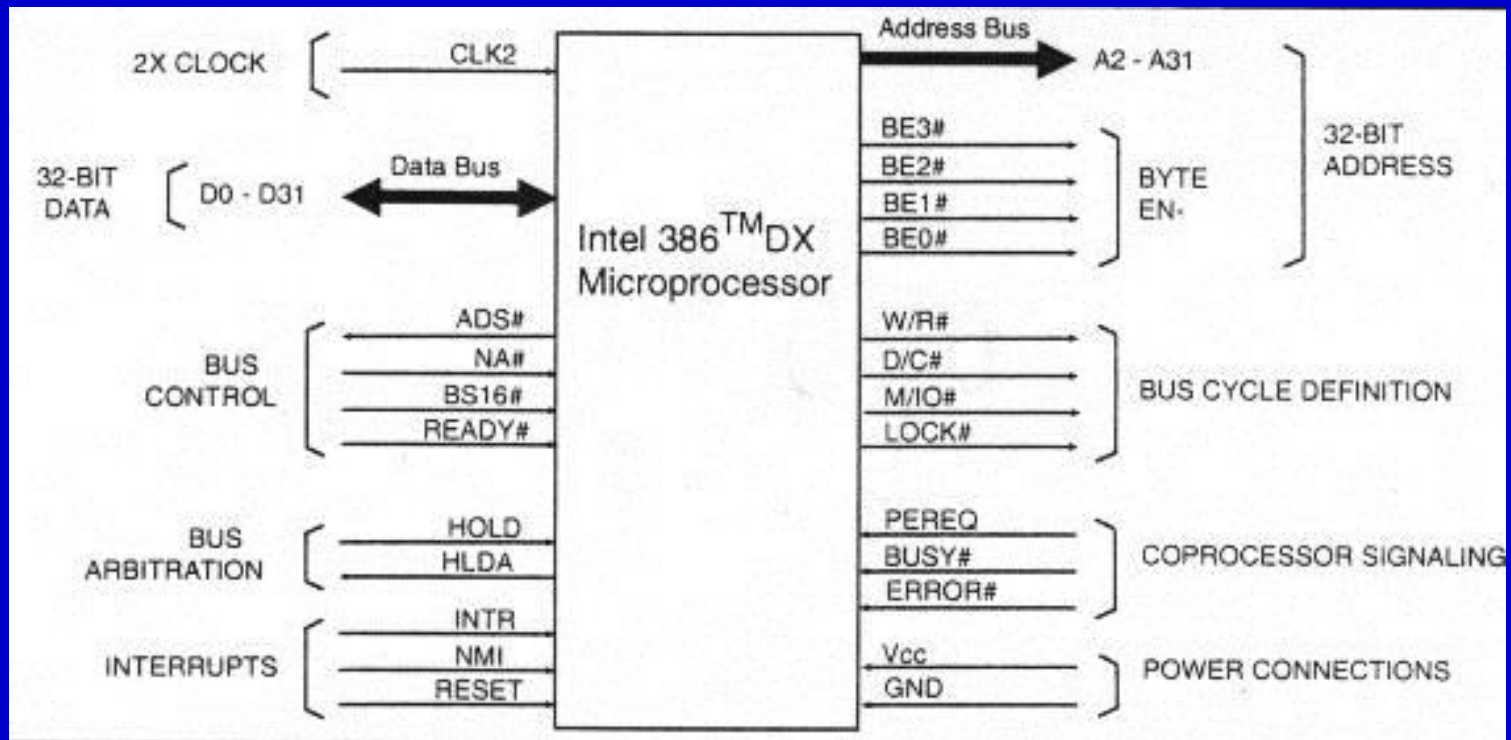
```
MOV AL, 5 ; AL = 5
ADD AL, -3 ; AL = 2
RET
```

C	Z	S	O	P	A
r	r	r	r	r	r

## Επεξεργαστής 80386



## Διάγραμμα ακίδων του $\mu\text{E}$ 80386



Οι γραμμές διευθύνσεων A0, A1 χρησιμοποιούνται για την παραγωγή των σημάτων BE0# → BE3#.

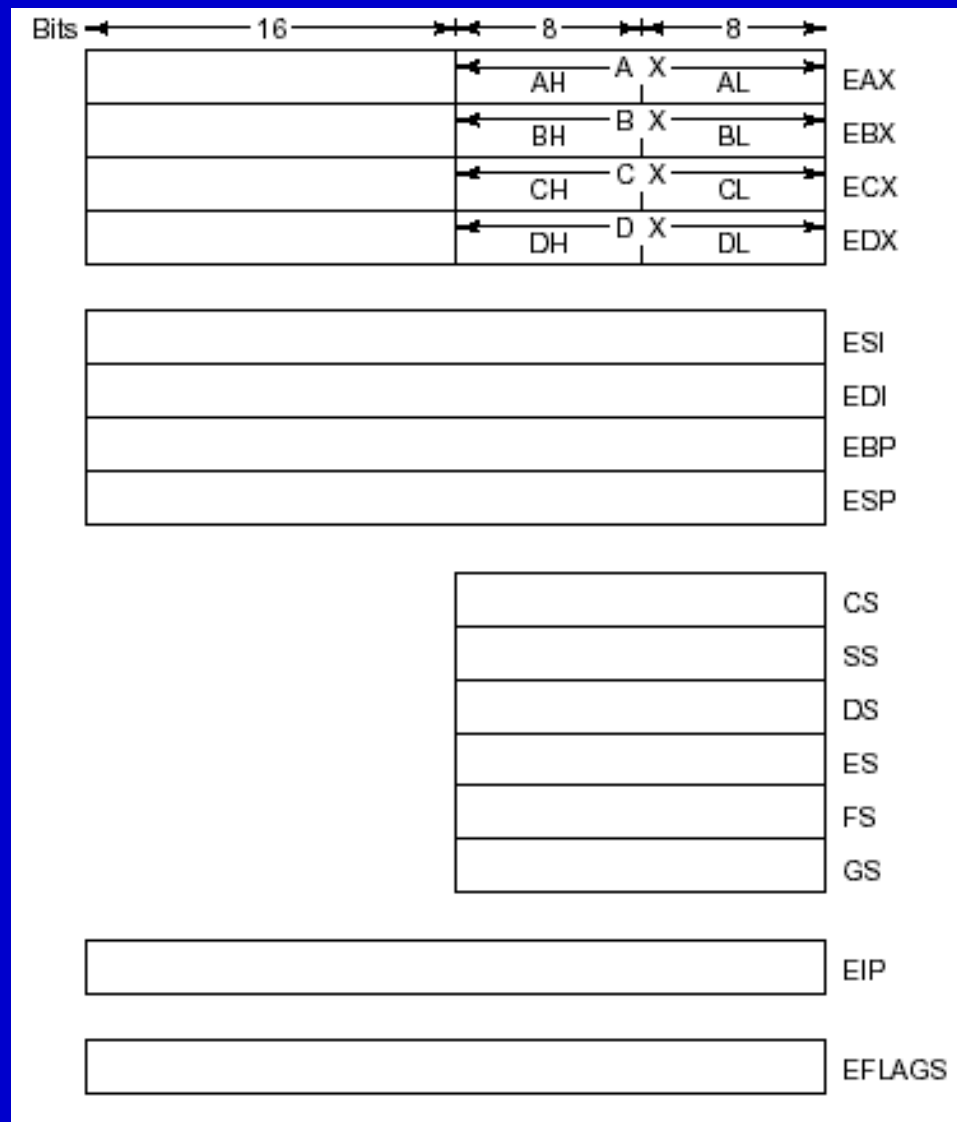
Οι γραμμές διευθύνσεων A0, A1 χρησιμοποιούνται για την παραγωγή των σημάτων BE0# → BE3#.

Δίαυλος διευθύνσεων 32 bit

Δίαυλος δεδομένων 32 bit

Μαθηματικός συνεπεξεργαστής 30387

# Βασικοί καταχωρητές της αρχιτεκτονικής x86-32 (80386→...).



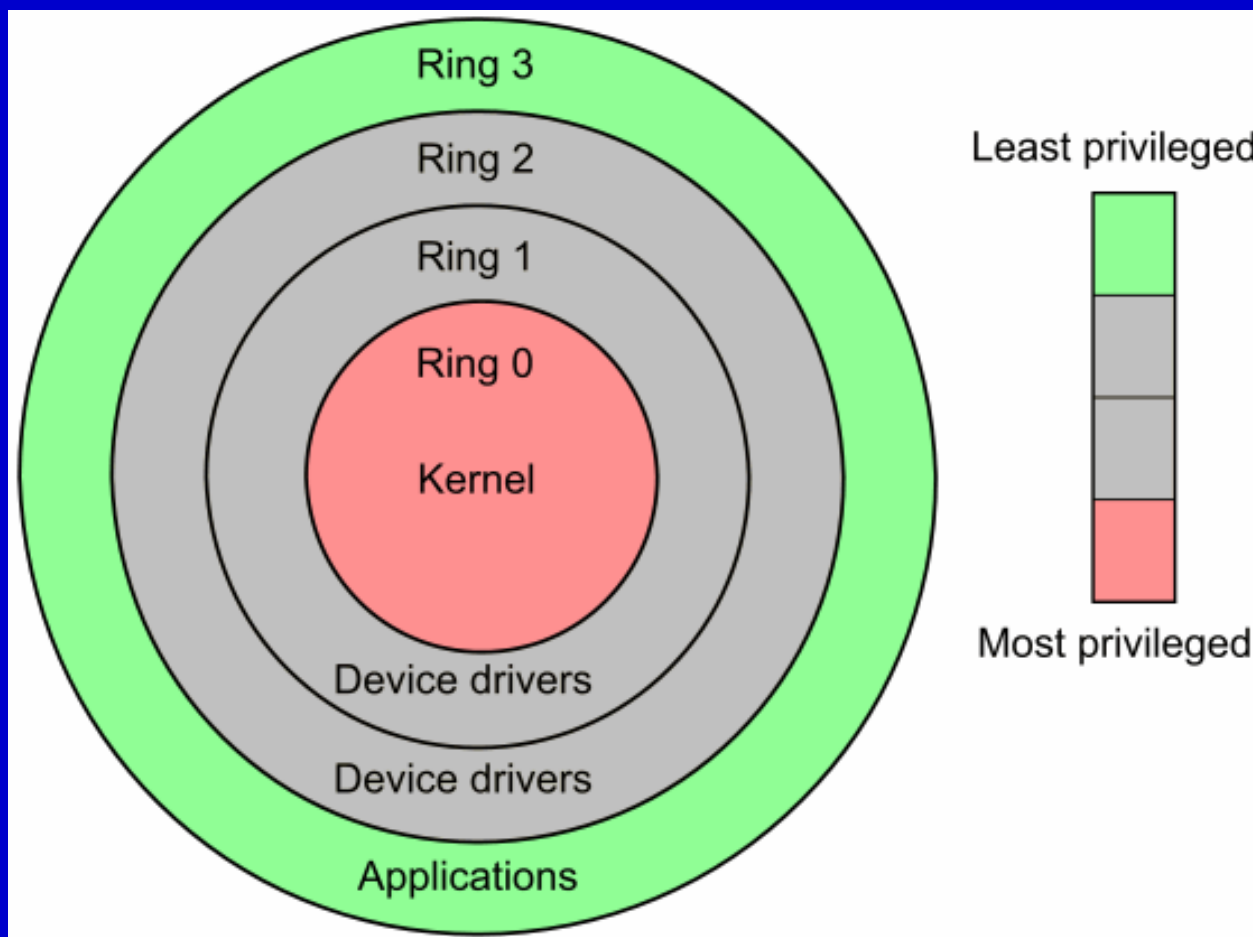
## Τρόποι λειτουργίας του επεξεργαστή 80386

**Real mode.** Λειτουργεί σαν υψηλής απόδοσης 8086.

**Protected mode.** Λειτουργεί στα 32-bit. Υποστηρίζει Memory Management, Virtual Addressing Paging, Protection, Multitasking.

**Virtual 8086 mode.** Επιτρέπει την εκτέλεση προγραμμάτων 8086 ενώ ο 80386 τρέχει λειτουργικό σύστημα που απαιτεί protected mode. Μπορεί και προσομοιώνει την λειτουργία πολλών 8086 ταυτόχρονα.

**Protection rings των επεξεργαστών –x86 και τα προγράμματα που εκτελούνται σε κάθε ένα από αυτά.**



## Εντολή ADD της αρχιτεκτονικής x86-32

### ADD — Add

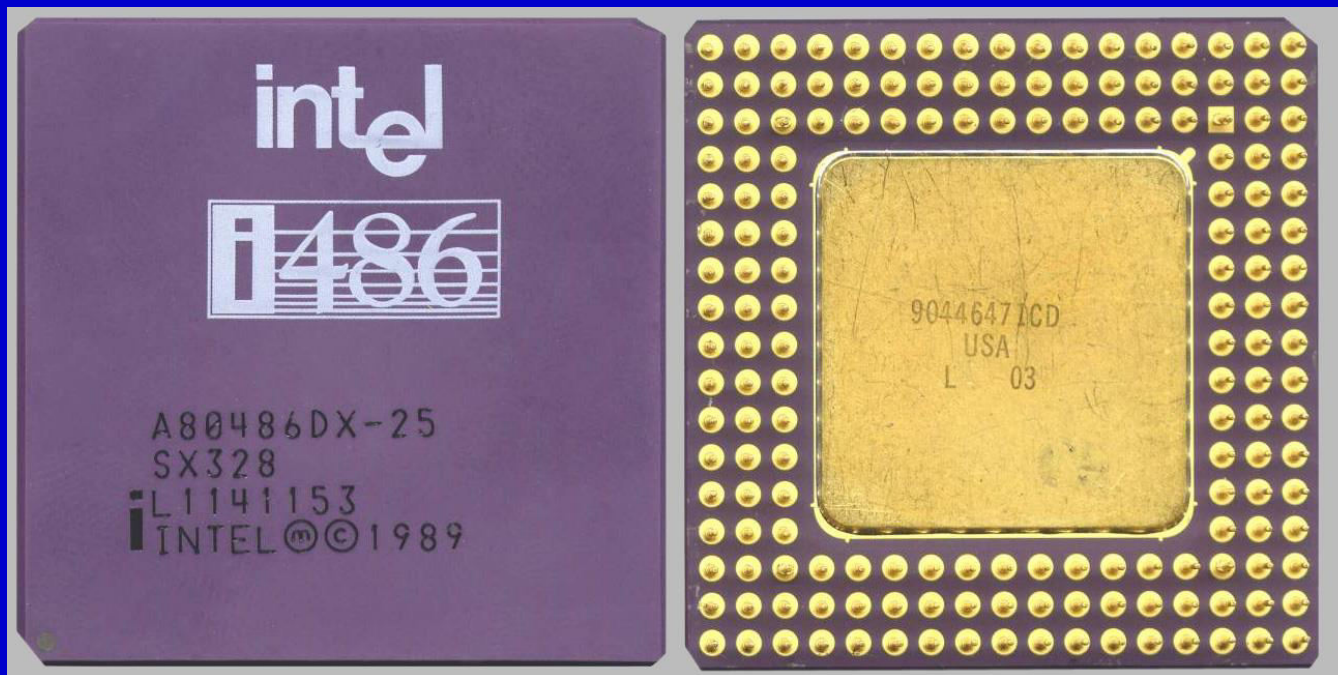
Opcode	Instruction	Clocks	Description
04 ib	ADD AL,imm8	2	Add immediate byte to AL
05 iw	ADD AX,imm16	2	Add immediate word to AX
05 id	ADD EAX,imm32	2	Add immediate dword to EAX
80 /o ib	ADD r/m8,imm8	2/7	Add immediate byte to r/m byte
81 /o iw	ADD r/m16,imm16	2/7	Add immediate word to r/m word
81 /o id	ADD r/m32,imm32	2/7	Add immediate dword to r/m dword
83 /o ib	ADD r/m16,imm8	2/7	Add sign-extended immediate byte to r/m word
83 /o id	ADD r/m32,imm8	2/7	Add sign-extended immediate byte to r/m dword
00 /r	ADD r/m8,r8	2/7	Add byte register to r/m byte
01 /r	ADD r/m16,r16	2/7	Add word register to r/m word
01 /r	ADD r/m32,r32	2/7	Add dword register to r/m dword
02 /r	ADD r8,r/m8	2/6	Add r/m byte to byte register
03 /r	ADD r16,r/m16	2/6	Add r/m word to word register
03 /r	ADD r32,r/m32	2/6	Add r/m dword to dword register

### Operation

DEST ← DEST + SRC;

Address	Machine Code	Assembly	
PUBLIC	_main		;COMDAT _main
_TEXT	SEGMENT		
_a\$ = -4			
_b\$ = -8			
_c\$ = -12			
_main	PROC NEAR; COMDAT		
; 3 :{			
00000	55	push ebp	
00001	8b ec	mov ebp, esp	
00003	83 ec 4c	sub esp, 76	; 0000004cH
00006	53	push ebx	
00007	56	push esi	
00008	57	push edi	
00009	8d 7d b4	lea edi, DWORD PTR [ebp-76]	
0000c	b9 13 00 00 00	mov ecx, 19	; 00000013H
00011	b8 cc cc cc cc	mov eax, -858993460	; ccccccccH
00016	f3 ab	rep stosd	
; 4 : int a, b, c;			
; 5 : a=b+c;			
00018	8b 45 f8	mov eax, DWORD PTR _b\$[ebp]	
0001b	03 45 f4	add eax, DWORD PTR _c\$[ebp]	
0001e	89 45 fc	mov DWORD PTR _a\$[ebp], eax	
; 6 :}			
00021	5f	pop edi	
00022	5e	pop esi	
00023	5b	pop ebx	
00024	8b e5	mov esp, ebp	
00026	5d	pop ebp	
00027	c3	ret 0	
_main	ENDP		
_TEXT	ENDS		
END			

# 80486



## 80486

The Intel 80486, also known as the i486 or 486, is a higher performance follow-up to the Intel 80386 microprocessor. The 80486 was introduced in 1989 and was the first tightly pipelined x86 design as well as the first x86 chip to use more than a million transistors, due to a large on-chip cache and an integrated floating-point unit.

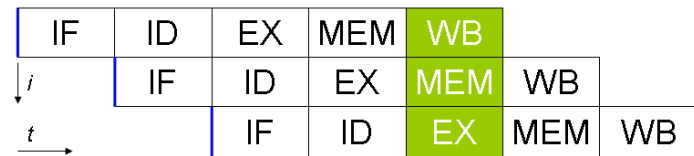
## CPU pipelining

Το *pipelining* είναι τεχνική αλληλοεπικάλυψης της εκτέλεσης των εντολών γλώσσας μηχανής σε έναν επεξεργαστή, ώστε να περιορισθεί ο χρόνος εκτέλεσης ενός συνόλου εντολών. Για την υλοποίηση του *pipelining* η μονάδα επεξεργασίας δεδομένων διαιρείται σε βαθμίδες και τοποθετούνται pipeline latches (flip-flops) μεταξύ των βαθμίδων.

Στην αρχή κάθε κύκλου ωρολογίου γίνεται εγγραφή των εξόδων των βαθμίδων στα pipeline latches των οποίων οι έξοδοί τους παραμένουν σταθερές κατά το υπόλοιπο του κύκλου για να χρησιμοποιηθούν σαν είσοδοι από την επόμενη βαθμίδα.

## Non Pipelined and Pipelined execution

Το *pipelining* είναι τεχνική αλληλοεπικάλυψης της εκτέλεσης των εντολών γλώσσας μηχανής σε έναν επεξεργαστή, ώστε να περιορισθεί ο χρόνος εκτέλεσης ενός συνόλου εντολών.



## Μνήμη cache

Η **μνήμη cache** παρεμβάλλεται μεταξύ της CPU και της κύριας μνήμης.

Είναι πολύ μικρότερου μεγέθους, αλλά μεγαλύτερης ταχύτητας συγκρινόμενη με την κύρια μνήμη.

Τα κύτταρα αποθήκευσης των 0,1 αυτής είναι, για λόγους ταχύτητας, στατικής τεχνολογίας.

Η μνήμη cache γεφυρώνει την διαφορά ταχύτητας μεταξύ της CPU και της κύριας μνήμης.

Είναι διάφανη στον προγραμματιστή, δηλαδή οι θέσεις της δεν μπορούν να προσπελασθούν άμεσα με εντολές γλώσσας μηχανής.

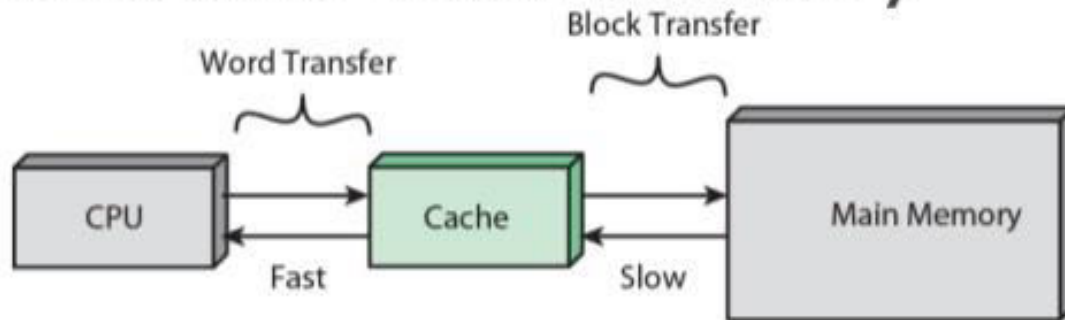
## Αρχή της τοπικότητας

Η αύξηση της απόδοσης των συστημάτων Η/Υ με χρήση μνήμης cache βασίζεται σε μία ιδιότητα των προγραμμάτων που ονομάζεται *αρχή της τοπικότητας* (*principle of locality*). Σύμφωνα με αυτή στα προγράμματα των υπολογιστών οι αναφορές στην μνήμη δεν γίνονται τυχαία, αλλά τείνουν να εντοπίζονται. Υπάρχουν δύο τύποι τοπικότητας αναφοράς στην μνήμη.

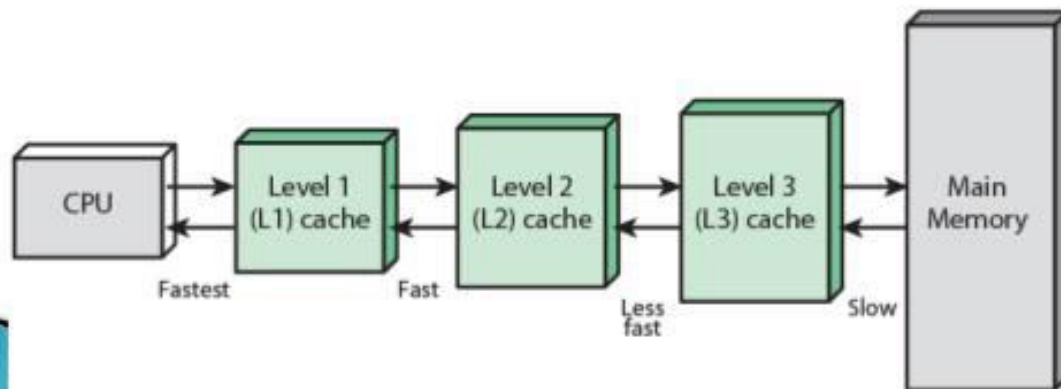
*Temporal locality.* Όταν γίνει μία προσπέλαση στην μνήμη υπάρχει πολύ μεγάλη πιθανότητα η θέση αυτή να προσπελασθεί στο άμεσο μέλλον.

*Spatial locality.* Όταν γίνει προσπέλαση σε μία θέση μνήμης είναι πολύ πιθανόν γειτονικές θέσεις μνήμης να προσπελασθούν άμεσα.

# Cache and Main Memory

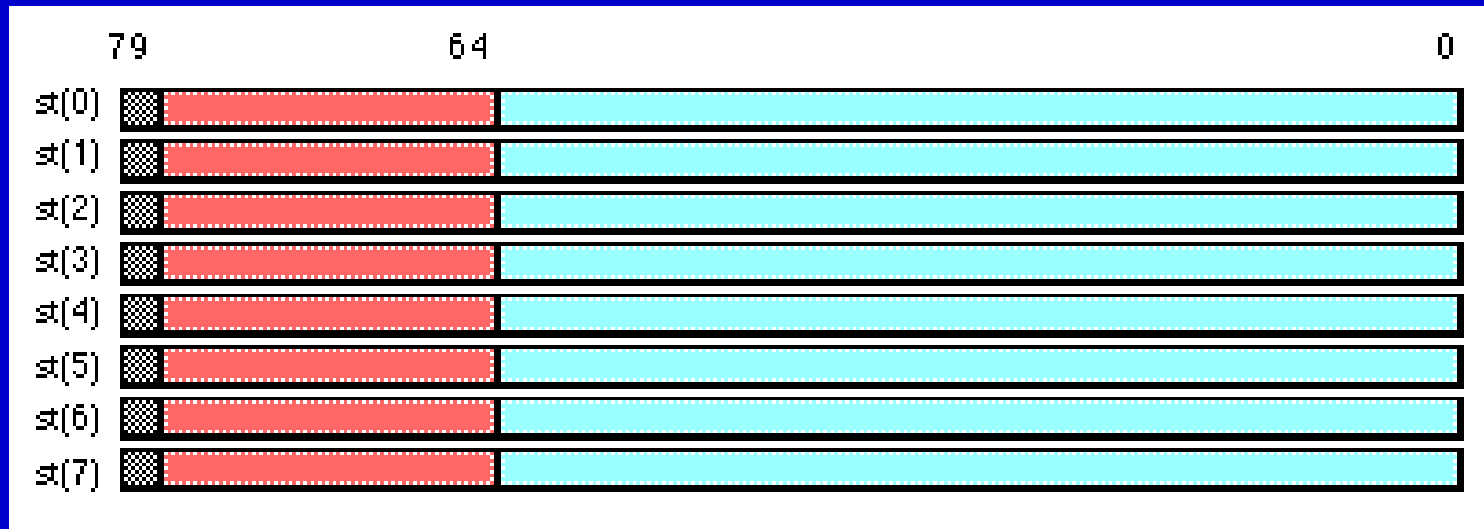


(a) Single cache



## Floating point registers

Σύνολο καταχωρητών των επεξεργαστών με αρχιτεκτονική -x86 που χρησιμοποιούνται για τις πράξεις κινητής υποδιαστολής.



## **Επεξεργαστές X86-32**

### **Pentium (1993)**

**2-way superscalar architecture  
instruction and data cache**

### **Pentium MMX**

**+MultiMedia eXtensions**

### **Pentium Pro**

**Out-Of-Order execution of instructions**

## **Pentium II**

Pentium Pro+MMX

## **Pentium III**

SSE

## **Pentium 4**

SSE2, SSE3

NetBurst

Hyperthreading

Multicore

## Superscalar execution

Στους επεξεργαστές με *υπερβαθμωτή (superscalar)* αρχιτεκτονική υπάρχουν δύο ή περισσότερες λειτουργικές μονάδες όπως Integer Execution Units, Floating Point Units, Branch Units, ενώ μία ειδική μονάδα (Instruction Issue Unit) ανακαλεί εντολές από την μνήμη cache, ανιχνεύει εάν υπάρχει δυνατότητα παραλληλισμού μεταξύ των εντολών (ύπαρξη ανεξάρτητων εντολών) και αναθέτει κατάλληλα την εκτέλεση των εντολών στις λειτουργικές μονάδες ώστε αυτές να εκτελούνται ταυτόχρονα.

## Εκτέλεση εντολών από επεξεργαστή με superscalar αρχιτεκτονική και δύο μονάδες εκτέλεσης εντολών

	IF	ID	EX	MEM	WB					
	IF	ID	EX	MEM	WB					
↓		IF	ID	EX	MEM	WB				
↓		IF	ID	EX	MEM	WB				
→			IF	ID	EX	MEM	WB			
			IF	ID	EX	MEM	WB			
				IF	ID	EX	MEM	WB		
				IF	ID	EX	MEM	WB		
					IF	ID	EX	MEM	WB	
					IF	ID	EX	MEM	WB	

## Out-of-Order Execution (OoOE)

Η τεχνική *Out-of-Order Execution* (OoOE) ενσωματώνεται στους περισσότερους σύγχρονους επεξεργαστές για να εκτελεστούν εντολές σε κύκλους ωρολογίου των οποίων η μη χρήση θα είχε σαν αποτέλεσμα την καθυστέρηση στην εκτέλεση του προγράμματος.

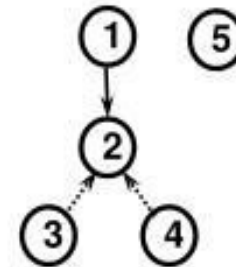
Με την τεχνική OoOE ο επεξεργαστής ανακαλεί ομάδες και τις εκτελεί με σειρά η οποία προσδιορίζεται από την διαθεσιμότητα δεδομένων εισόδου σε αυτές, παρά με την σειρά τους στο πρόγραμμα.

Με αυτό τον τρόπο αποφεύγεται ο επεξεργαστής να παραμένει ανενεργός, ενώ ανακτώνται τα δεδομένα για την επόμενη εντολή στο πρόγραμμα, και εκτελεί αντί της αμέσως επόμενης εντολής κάποια άλλη που μπορεί να εκτελεστεί άμεσα.

# Out Of Order Execution – Example

- ◆ Assume that executing a divide operation takes 20 cycles

(1)  $r1 \leftarrow r5 / r4$   
(2)  $r3 \leftarrow r1 + r8$   
(3)  $r8 \leftarrow r5 + 1$   
(4)  $r3 \leftarrow r7 - 2$   
(5)  $r6 \leftarrow r6 + r7$



- ◆ **Inst2 has a RAW dependency on r1 with Inst1**
  - ❖ It cannot be executed in parallel with Inst1
- ◆ **Can successive instructions pass Inst2 ?**
  - ❖ Inst3 cannot since Inst2 must read r8 before Inst3 writes to it
  - ❖ Inst4 cannot since it must write to r3 after Inst2
  - ❖ Inst5 can

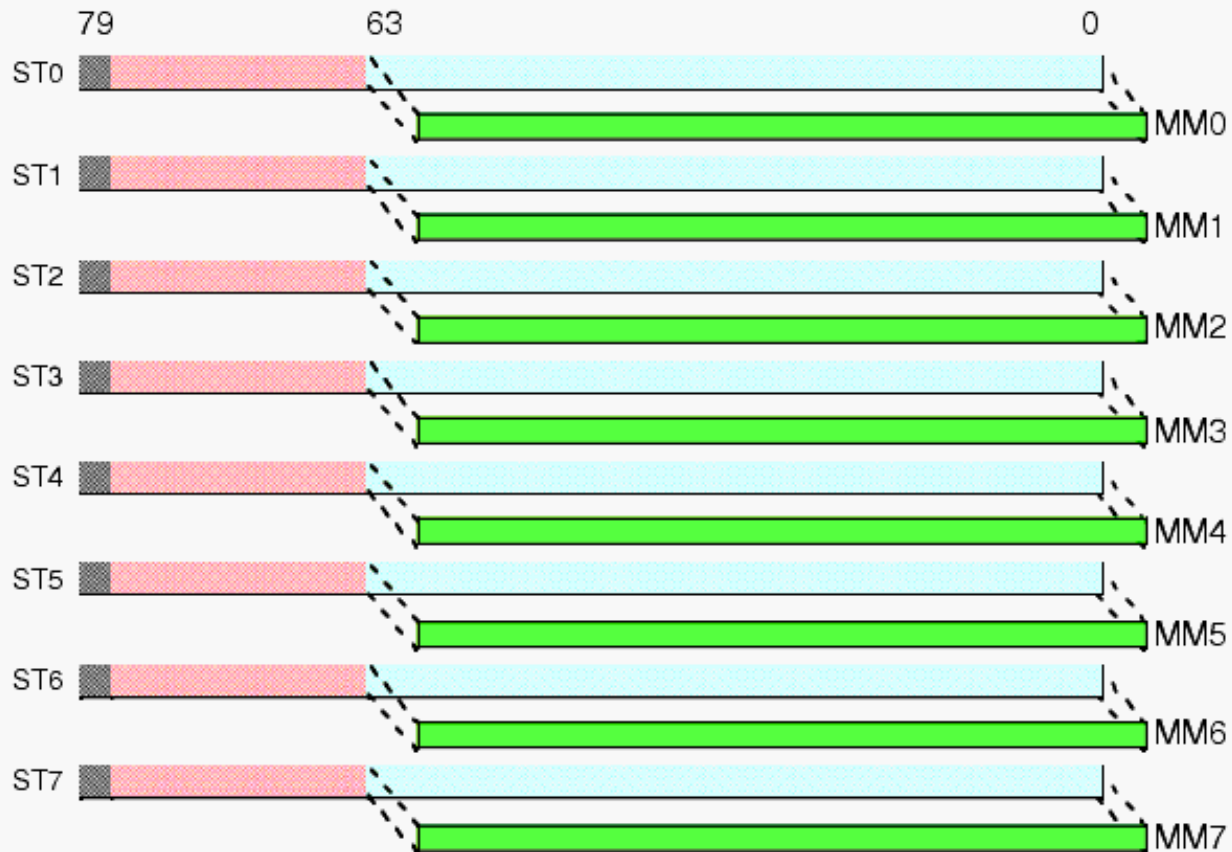
## Τεχνολογίες MMX και SSE

Η τεχνολογία **MMX**(**M**ulti-**M**edia **eX**tensions) περιλαμβάνει 57 πρόσθετες εντολές που ενσωματώθηκαν αρχικά στον επεξεργαστή Pentium MMX για αυξημένη απόδοση σε εφαρμογές πολυμέσων. Οι εντολές αυτές εκτελούν μαθηματικές πράξεις σε πολλαπλά σύνολα δεδομένων την ίδια χρονική στιγμή (SIMD). Η CPU σε MMX mode, χρησιμοποιεί τους floating-point registers για MMX registers.

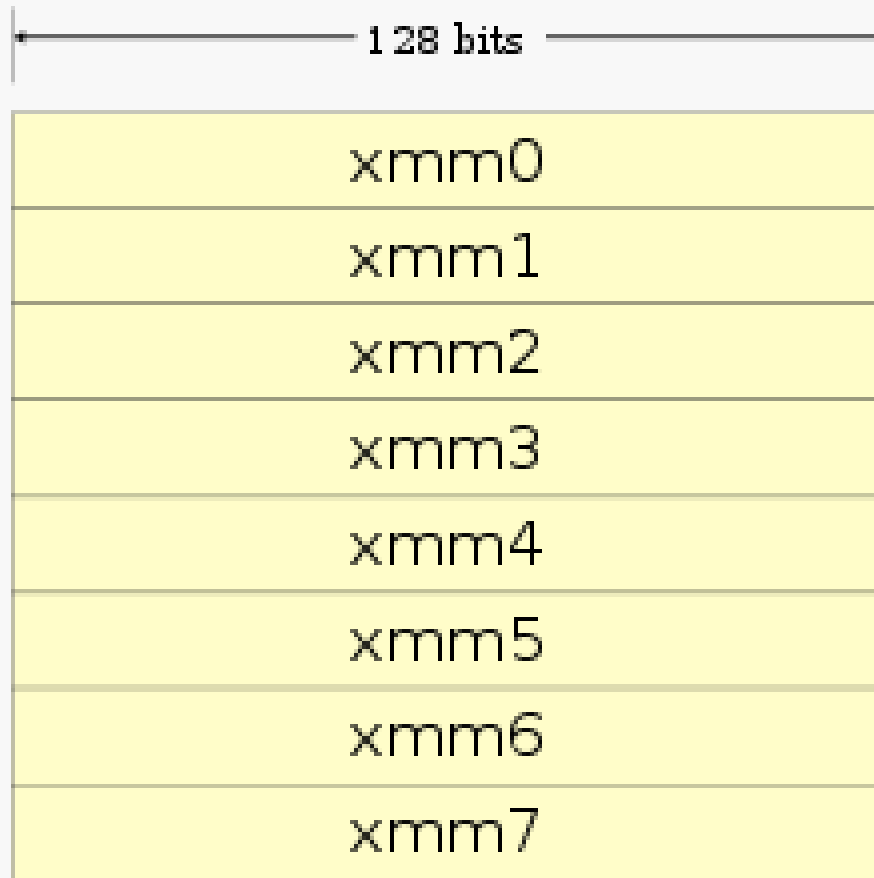
Όλες οι CPU από τον Pentium MMX και εξής, εκτός από τον Pentium Pro, περιλάμβαναν την τεχνολογία MMX.

Στην συνέχεια προστέθηκαν στο σύνολο εντολών οι επεκτάσεις SSE (SIMD Streaming Extensions). Με την τεχνολογία SSE προσετέθησαν επί πλέον καταχωρητές, έτσι ώστε floating-point και multimedia operations να μπορούν να εκτελούνται ταυτόχρονα.

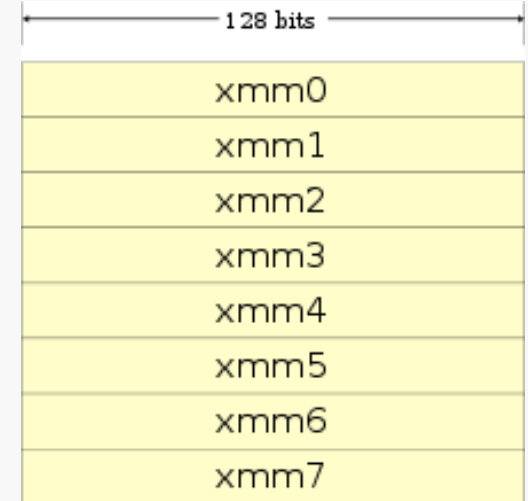
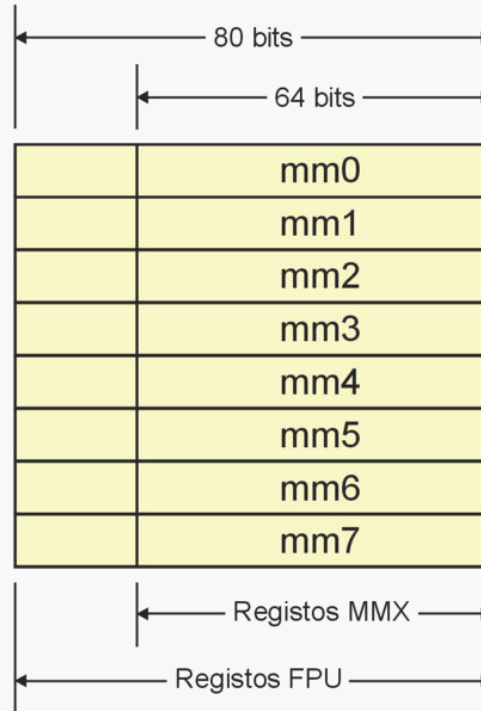
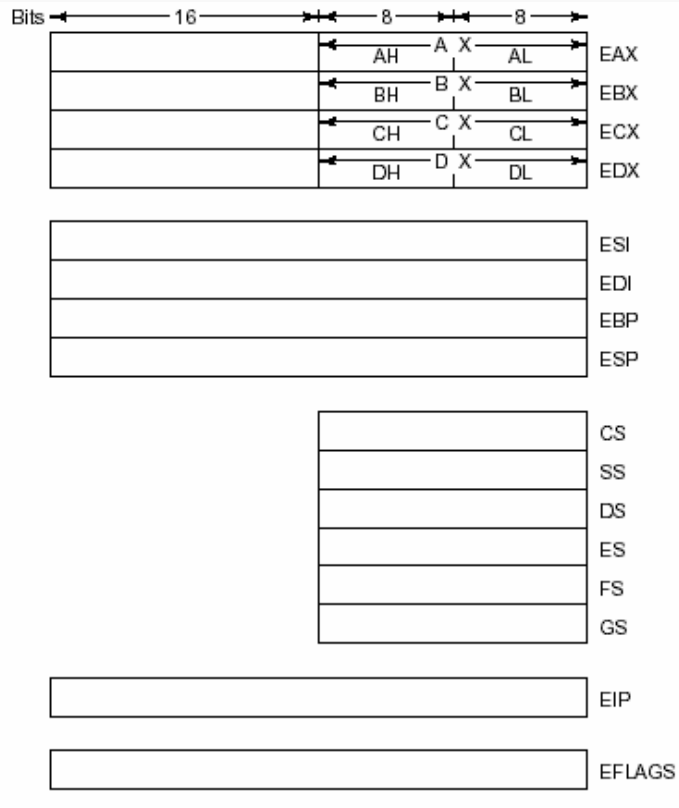
## Καταχωρητές MMX (τμήμα των floating-point registers)



## Σύνολο καταχωρητών SSE (Streaming SIMD Extensions)



# Μοντέλο προγραμματισμού x86-32



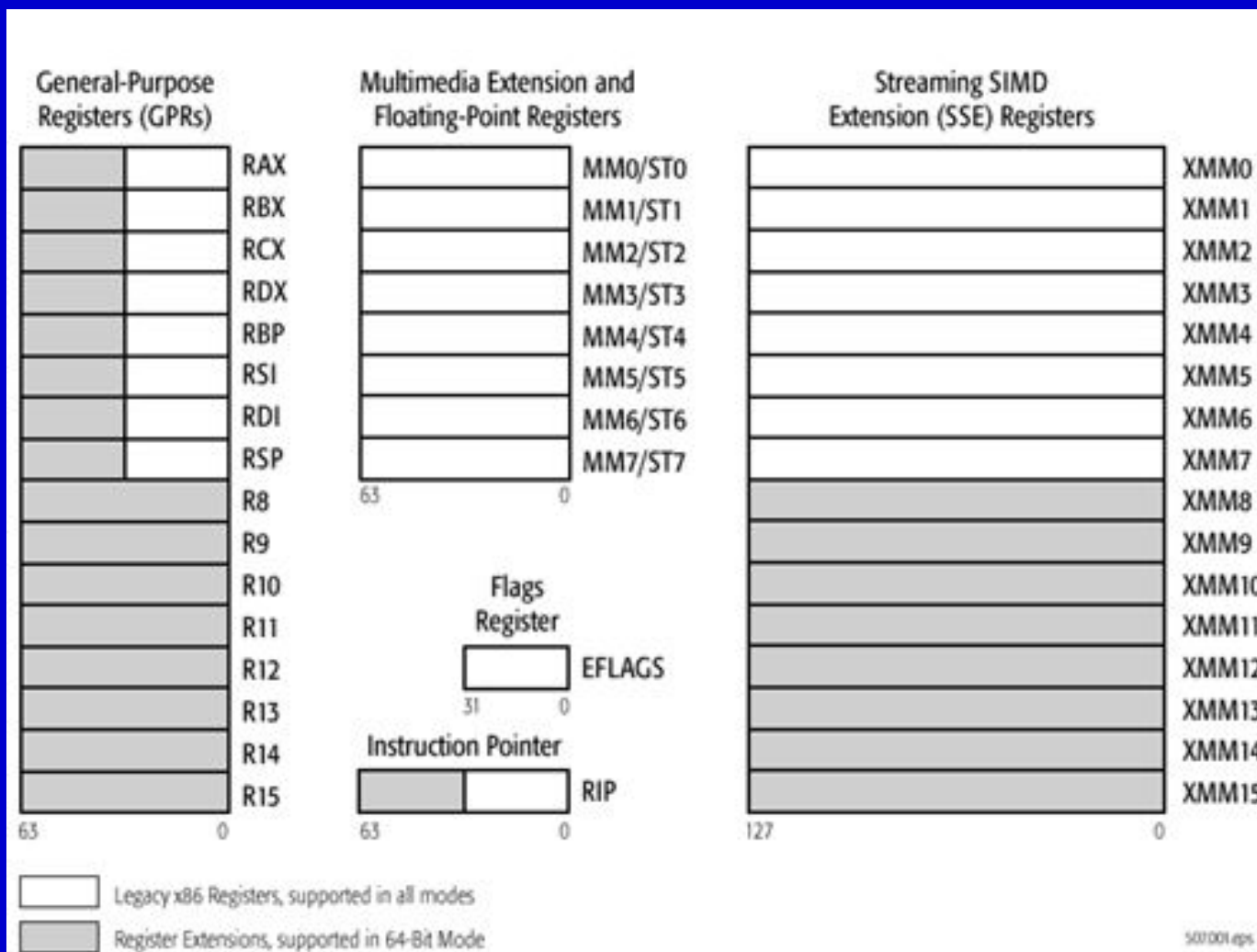
## Αρχιτεκτονικές, υλοποιήσεις και sockets επεξεργαστών IA-64

Core 2 Duo, Core 2 Quad

**Core i3, i5, i7, Celeron, Pentium**

**Core i9**

# Μοντέλο προγραμματισμού x86-64 (Core 2, i3/i5/i7/i9)



## Τρόποι λειτουργίας των επεξεργαστών x86-64.

Operating Mode		Operating System Required	Application Recompile Required	Defaults		Register Extensions	Typical
				Address Size (bits)	Operand Size (bits)		GPR Width (bits)
Long Mode	64-Bit Mode	New 64-bit OS	yes	64	32	yes	64
	Compatibility Mode		no	32		no	32
				16	16		16
Legacy Mode	Protected Mode	Legacy 32-bit OS	no	32	32	no	32
				16	16		
	Virtual-8086 Mode			16	16	no	16
	Real Mode			Legacy 16-bit OS			

## Τρόποι λειτουργίας των επεξεργαστών x86-64

Register or Stack	Legacy and Compatibility Modes			64-Bit Mode		
	Name	Number	Size (bits)	Name	Number	Size (bits)
General-Purpose Registers (GPRs)	EAX, EBX, ECX, EDX, EBP, ESI, EDI, ESP	8	32	RAX, RBX, RCX, RDX, RBP, RSI, RDI, RSP, R8–R15	16	64
128-Bit XMM Registers	XMM0–XMM7	8	128	XMM0–XMM15	16	128
64-Bit MMX Registers	MMX0–MMX7	8	64	MMX0–MMX7	8	64
x87 Registers	FPR0–FPR7	8	80	FPR0–FPR7	8	80
Instruction Pointer	EIP	1	32	RIP	1	64
Flags	EFLAGS	1	32	RFLAGS	1	64
Stack	–		16 or 32	–		64

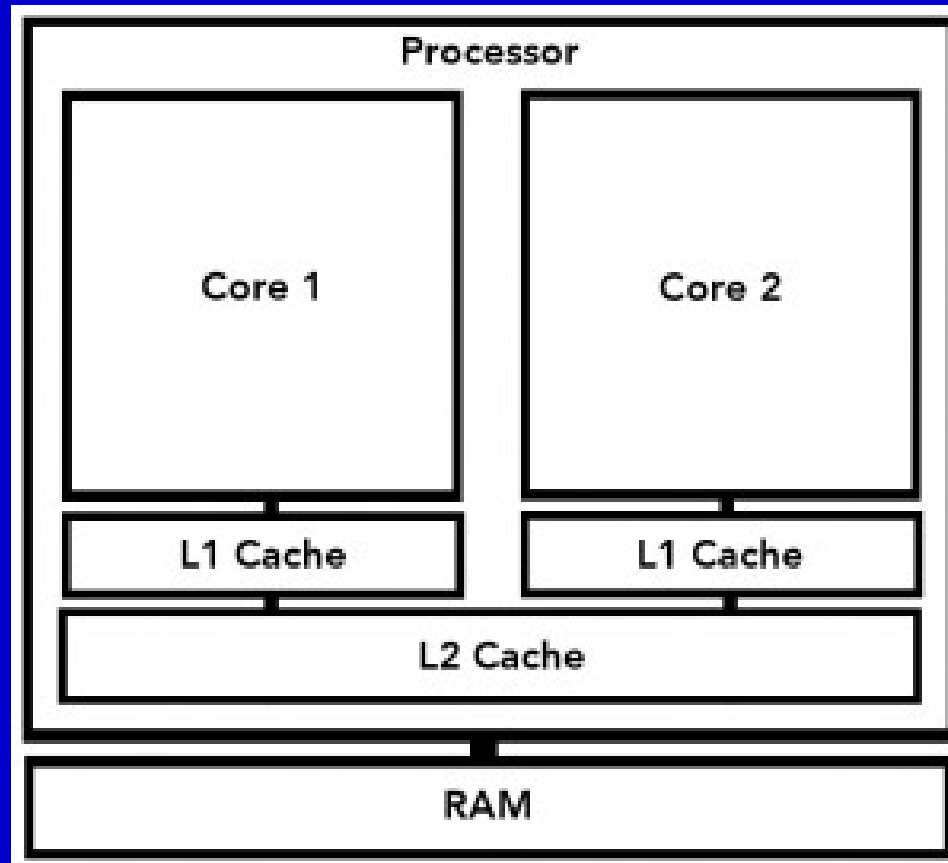
## Multi-core processor

A **multi-core processor** is a single computing component with two or more independent processing units called **cores**, which read and execute machine code instructions.

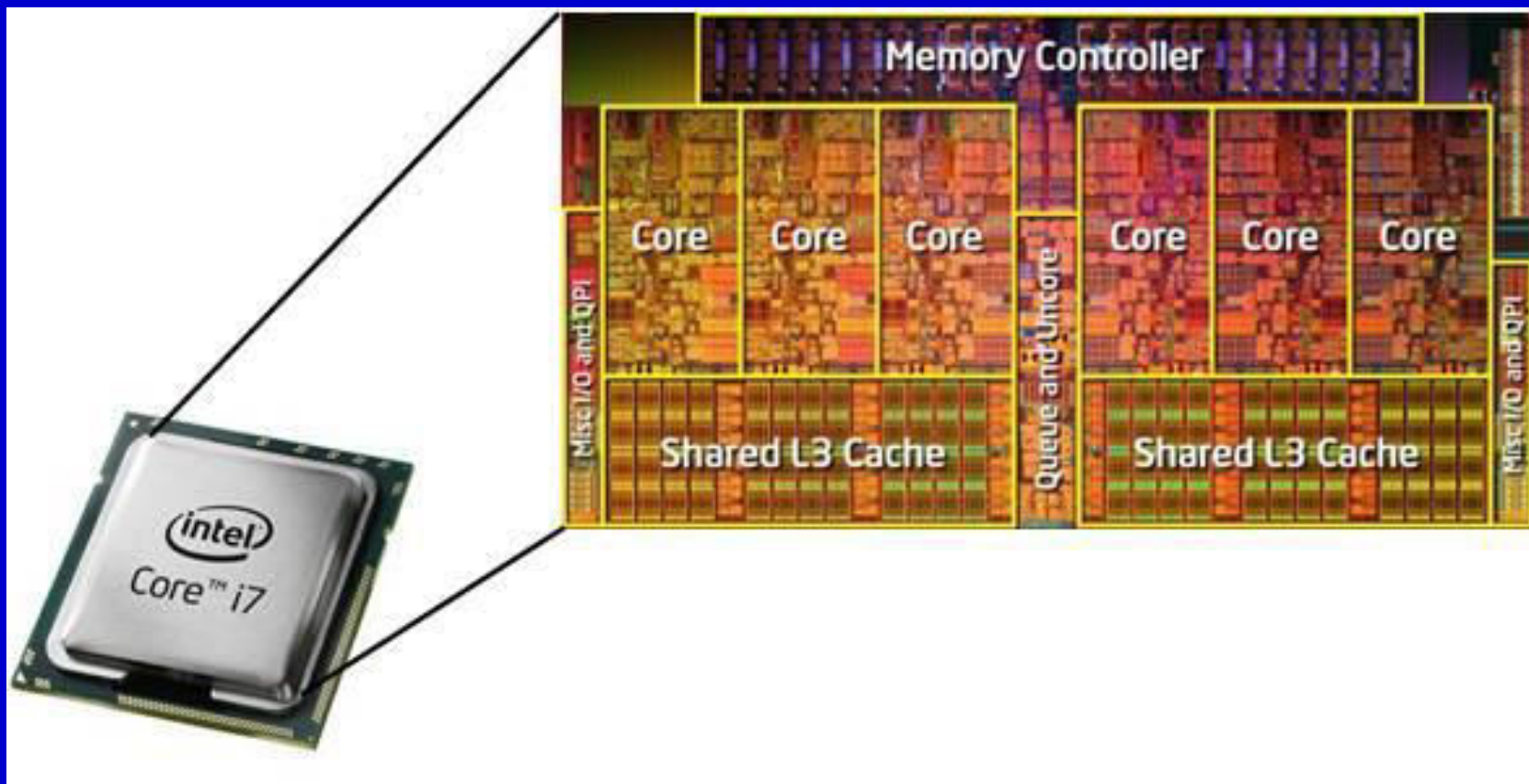
The single processor can run multiple instructions on separate cores at the same time, increasing the overall speed. Manufacturers typically integrate the cores onto a single integrated circuit die or onto multiple dies in a single chip package. The processors currently used in almost all personal computers are multi-core. A multi-core processor implements multiprocessing in a single physical package.

Common topologies to interconnect cores are **bus, crossbar, ring, and two-dimensional mesh** networks.

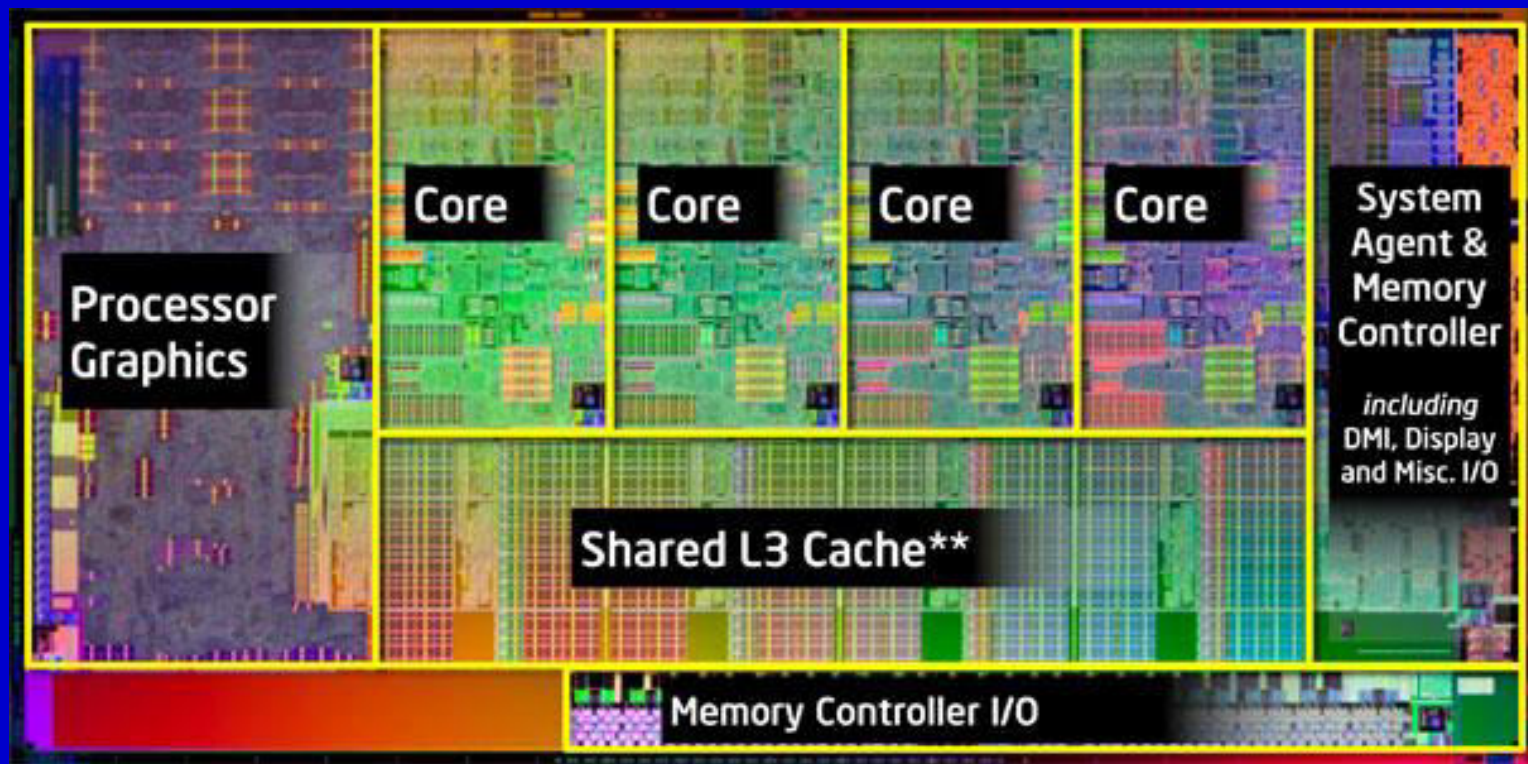
## Dual-core επεξεργαστής με κοινή L2 cache



# Επεξεργαστής i7



# Multi-core επεξεργαστής με ενσωματωμένη GPU



## IEEE floating-point standard 754

Έως το 1980 κάθε εταιρεία είχε το δικό της floating-point format. Σήμερα οι περισσότεροι επεξεργαστές διαθέτουν μονάδες κινητής υποδιαστολής που ακολουθούν το πρότυπο 754 της IEEE. Στο πρότυπο αυτό καθορίζονται τρία formats. Single precision (32-bit), Double precision (64-bit) και Extended precision (80-bit) formats.

# Single and double precision floating-point IEEE 754 format

## Single Precision



## Double Precision



S: Sign

E: Exponent

F: Fraction

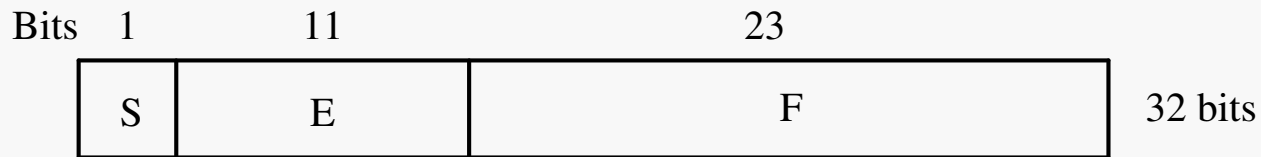
Exponent: Excess 127 (Single Precision)

Exponent: Excess 1023 (Double Precision)

Fraction: Normalized

# Υπολογισμός της τιμής αριθμού σε double precision IEEE 754 format

## Single Precision



Εάν  $0 < E < 255$ , τότε  $N = (-1)^S \cdot 2^{E-127} \cdot (1.F)$

Εάν  $E = 0$  και  $F \neq 0$ ,  $N = (-1)^S \cdot 2^{126} \cdot (0.F)$

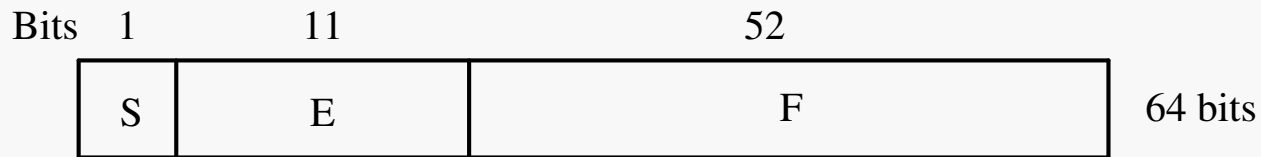
Εάν  $E = 0$  και  $F = 0$ ,  $N = (-1)^S \cdot 0$

Εάν  $E = 255$  και  $F \neq 0$ ,  $N = \text{NaN (Not a Number)}$

Εάν  $E = 255$  και  $F = 0$ ,  $N = (-1)^S \cdot \infty$

# Υπολογισμός της τιμής αριθμού σε double precision IEEE 754 format

Single Precision



Εάν  $0 < E < 2047$ , τότε  $N = (-1)^S \cdot 2^{E-1023} (1.F)$

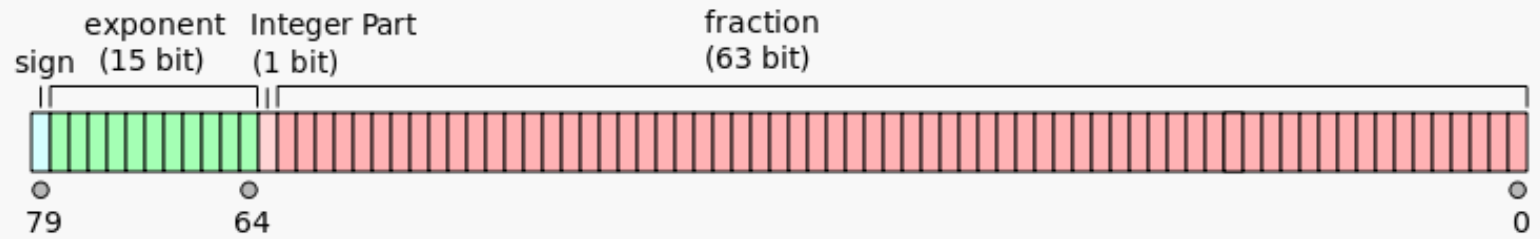
Εάν  $E = 0$  και  $F \neq 0$ ,  $N = (-1)^S \cdot 2^{1022} \cdot (0.F)$

Εάν  $E = 0$  και  $F = 0$ ,  $N = (-1)^S \cdot 0$

Εάν  $E = 2047$  και  $F \neq 0$ ,  $N = \text{NaN (Not a Number)}$

Εάν  $E = 2047$  και  $F = 0$ ,  $N = (-1)^S \cdot \infty$

# -x86 extended precision floating-point format



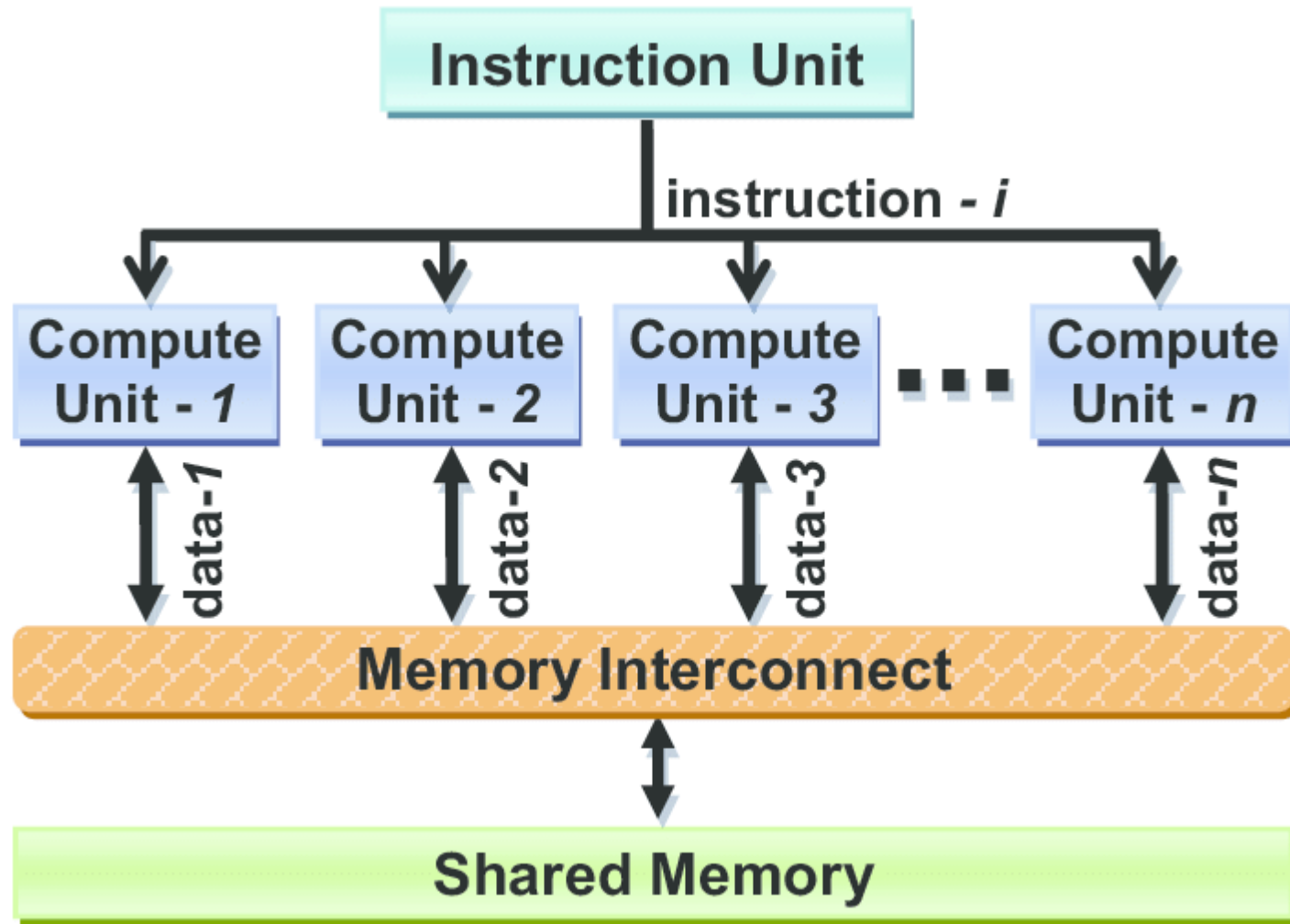
## FLOPS

**FLOPS** (**F**loating-point **O**perations **P**er **S**econd) είναι ένα μέτρο της απόδοσης των υπολογιστών, χρήσιμο σε πεδία όπως επιστημονικοί υπολογισμοί που κάνουν εκτεταμένη χρήση floating-point υπολογισμών. Για τέτοιες περιπτώσεις είναι πιο ακριβές μέτρο από το γενικό instructions per second. Γενικά τα FLOPS δεν προσδιορίζονται από θεωρητικούς υπολογισμούς, αλλά μετρούνται στην πράξη με benchmarks.

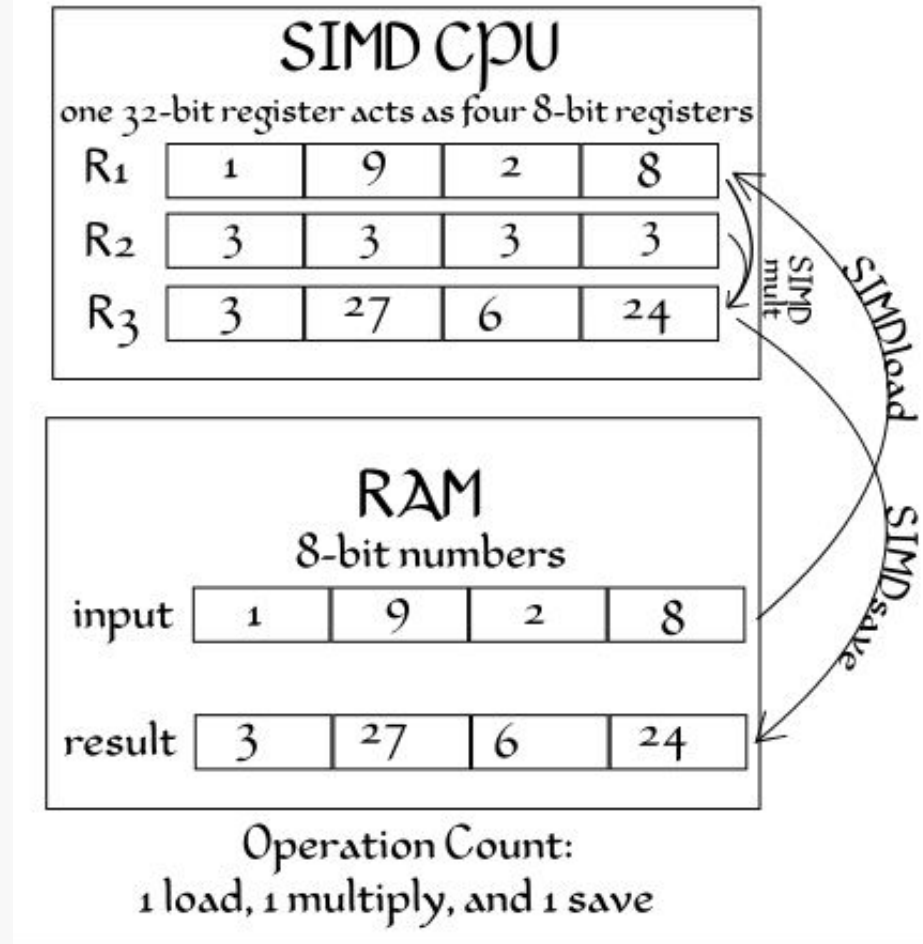
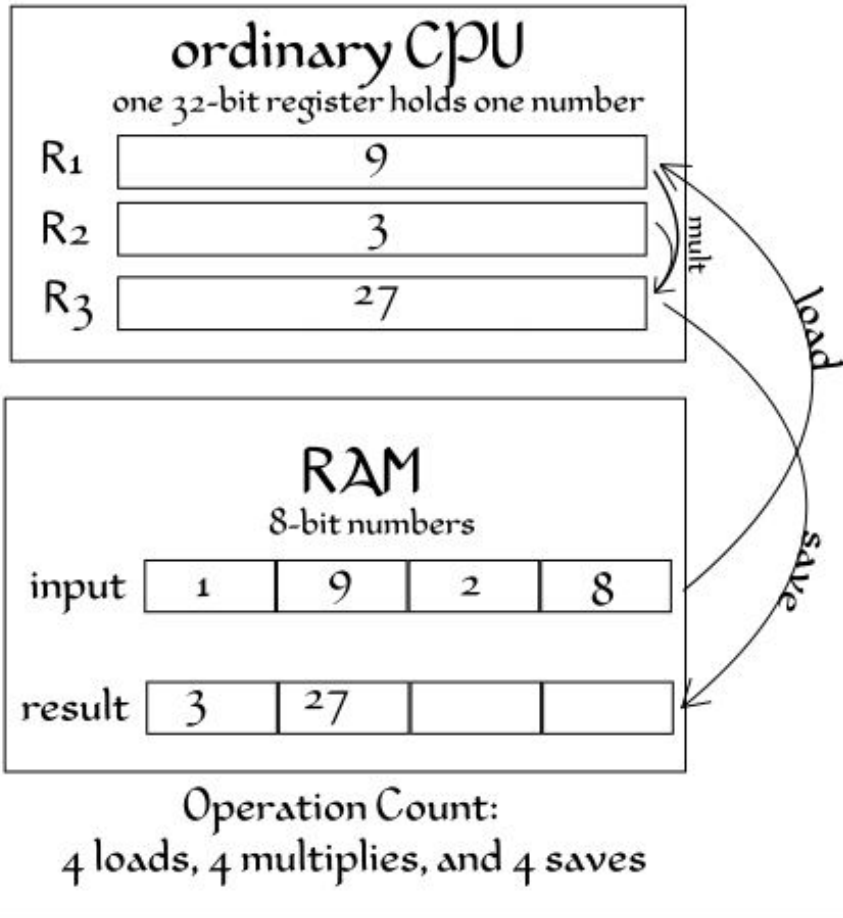
# SIMD

Με τον όρο *Single Instruction, Multiple Data (SIMD)*, εννοούμε μία κλάση παράλληλων υπολογιστών στην ταξινόμηση κατά Flynn. Ο όρος SIMD αναφέρεται σε υπολογιστές με πολλές μονάδες επεξεργασίας που εκτελούν ταυτόχρονα την ίδια λειτουργία σε διαφορετικά δεδομένα. Με υπολογιστές αυτού του τύπου επιτυγχάνεται παραλληλισμός σε επίπεδο επεξεργασίας δεδομένων. Η τεχνολογία SIMD είναι εφαρμόσιμη σε λειτουργίες των πολυμέσων, όπως ρύθμιση contrast σε ψηφιακές εικόνες ή ρύθμιση της έντασης σε ψηφιακό ήχο. Οι περισσότερες μοντέρνες CPU περιλαμβάνουν SIMD instructions με στόχο να βελτιωθεί η απόδοσή τους σε εφαρμογές πολυμέσων.

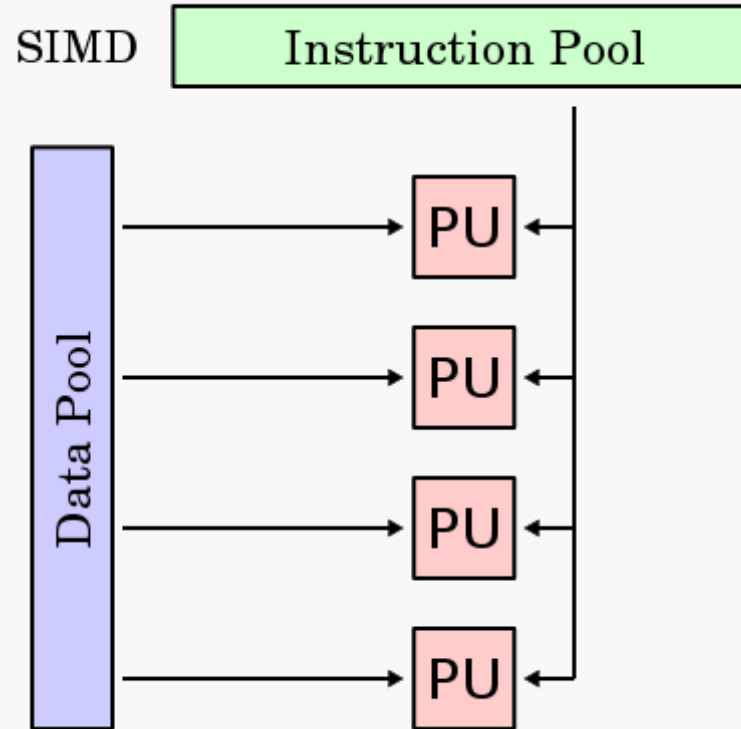
# SIMD



# SISD vs SIMD

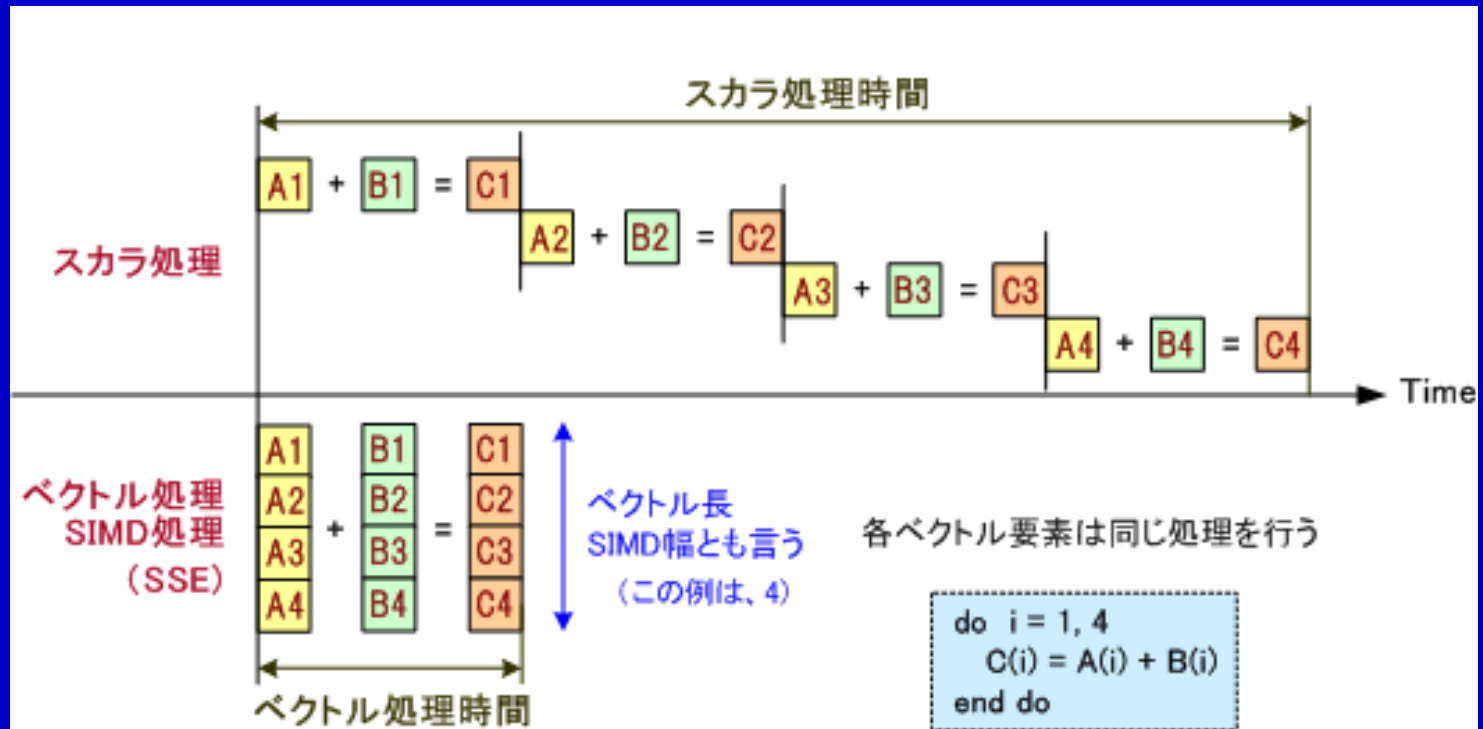


# Αρχιτεκτονική SIMD



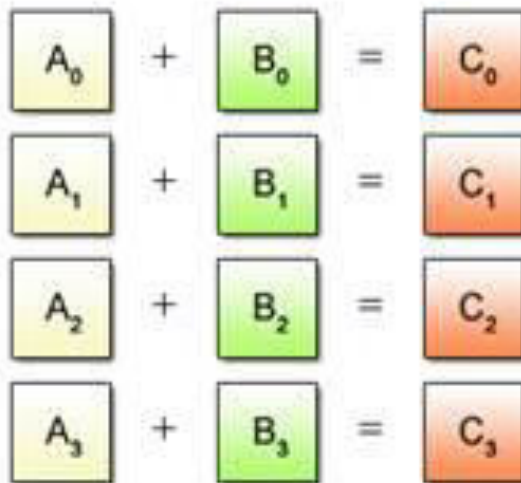
PU: Processing Unit

Εκτέλεση πράξεων σε ανύσματα δεδομένων χωρίς την χρήση και με χρήση εντολών SIMD.

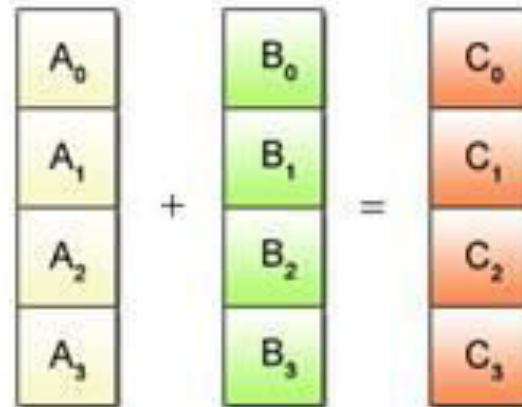


# Εκτέλεση πράξεων σε ανύσματα με τον κλασσικό τρόπο και με SIMD.

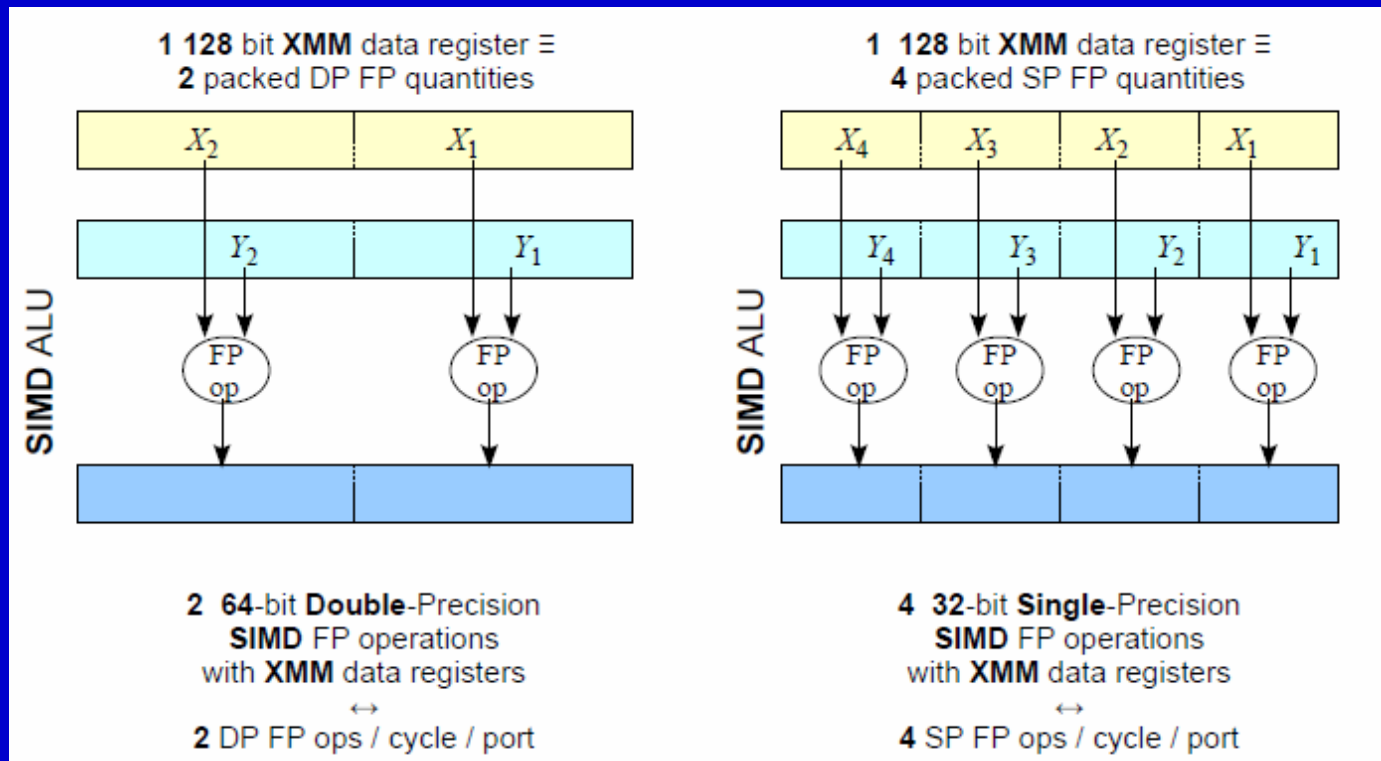
(a) Scalar Operation



(b) SIMD Operation



## Εκτέλεση των εντολών SSE



## MIPS

Η **MIPS** (Microprocessor without Interlocked Pipeline Stages) είναι αρχιτεκτονική συνόλου εντολών (ISA) γλώσσας μηχανής που αναπτύχθηκε από την εταιρεία MIPS Technologies. Οι πρώτες αρχιτεκτονικές MIPS ήταν των 32 bit, ενώ οι επόμενες ήταν των 64 bit. Υπάρχουν πολλές εκδόσεις του συνόλου εντολών MIPS. Οι τελευταίες εκδόσεις της αρχιτεκτονικής MIPS είναι η MIPS32 (για υλοποιήσεις των 32 bit) και η MIPS64 (για υλοποιήσεις των 64 bit).

Οι υλοποιήσεις της αρχιτεκτονικής MIPS προορίζονται σήμερα κύρια για χρήση σε ενσωματωμένα συστήματα (embedded systems), όπως σε συστήματα με Windows CE, routers και video game consoles.

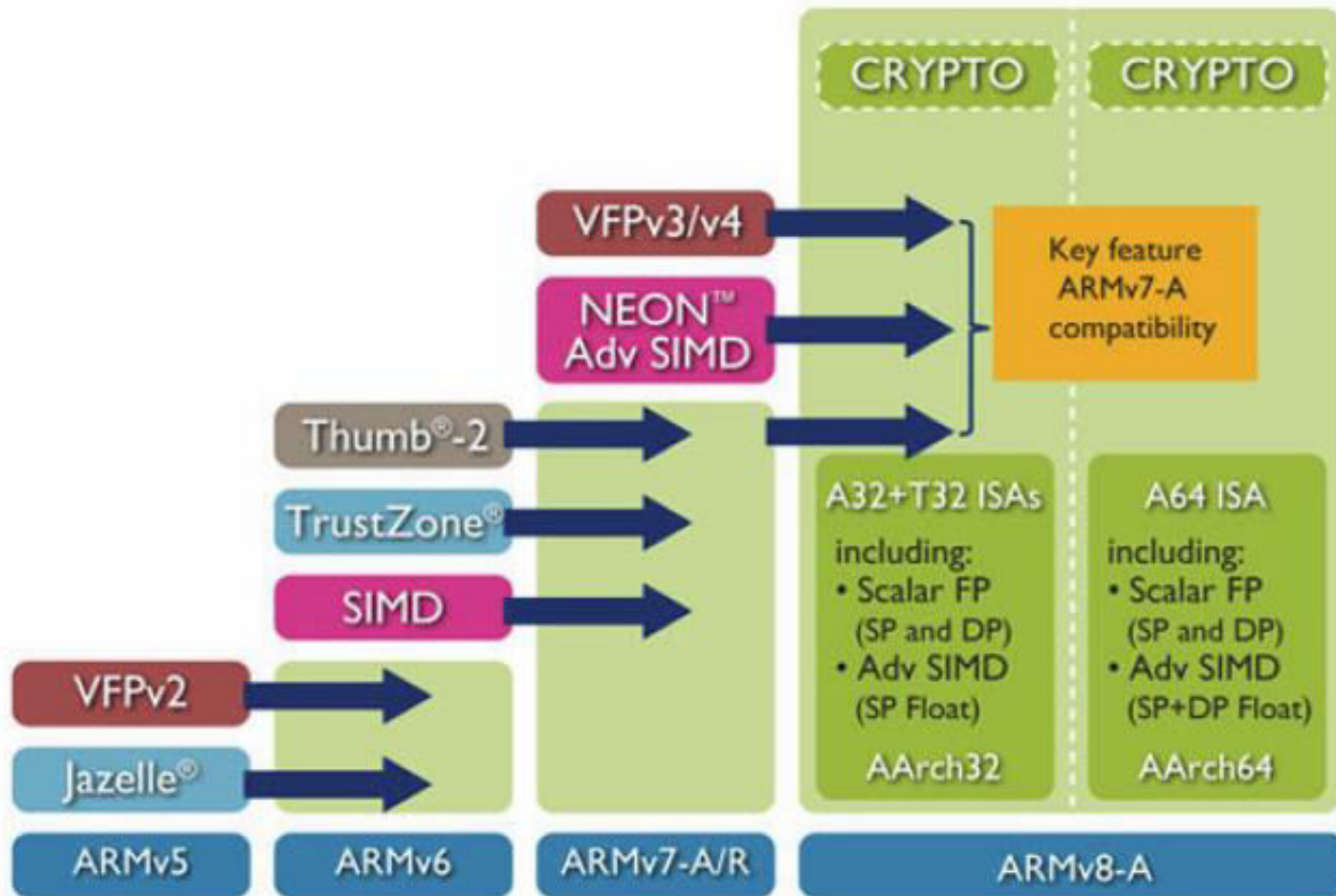
## ARM

**ARM** is a family of instruction set architectures for computer processors based on a reduced instruction set computing (RISC) architecture developed by British company ARM Holdings.

A RISC-based computer design approach means ARM processors require significantly fewer transistors than typical CISC x86 processors in most personal computers. This approach reduces costs, heat and power use. These are desirable traits for light, portable, battery-powered devices, including smartphones, laptops, tablet and notepad computers, and other embedded systems. A simpler design facilitates more efficient multi-core CPUs and higher core counts at lower cost, providing improved energy efficiency for servers.

# ARM Architecture versions

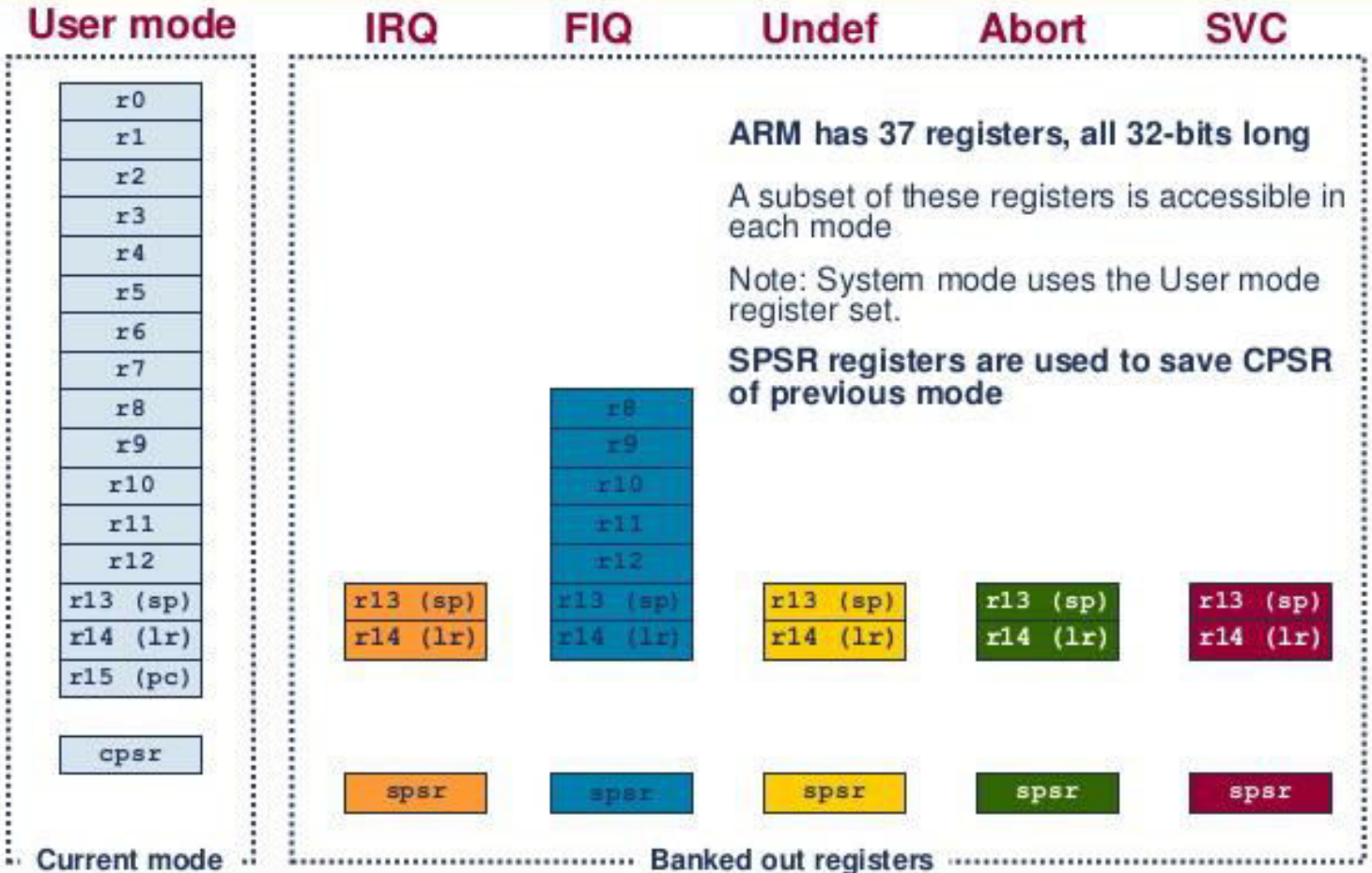
(From [arm.com](http://arm.com))



- ARM architecture has been extended over several versions.
  - ARM7TDMI.
  - ARM9 – includes “Thumb” instruction set
  - ARM10 – for multimedia (graphics, video, etc.)
  - ARM11 – high performance + Jazelle (Java)
  - SecurCore – for security app’s (smart cards)
  - Cortex-M – Optimized for microcontrollers
  - Cortex-A - High performance (multimedia systems)
  - Cortex-R – Optimized for real-time app’s
  - StrongARM – portable communication devices

Architecture	Bit width	Cores designed by ARM Holdings	Cores designed by 3rd parties	Cortex profile
ARMv1	32/26	ARM1		
ARMv2	32/26	ARM2, ARM3	Amber	
ARMv3	32	ARM6, ARM7		
ARMv4	32	ARM8	StrongARM, FA526	
ARMv4T	32	ARM7TDMI, ARM9TDMI		
ARMv5	32	ARM7EJ, ARM9E, ARM10E	XScale, FA626TE, Feroceon, PJ1/Mohawk	
ARMv6	32	ARM11		
ARMv6-M	32	ARM Cortex-M0, ARM Cortex-M0+, ARM Cortex-M1		Microcontroller
ARMv7-M	32	ARM Cortex-M3		Microcontroller
ARMv7E-M	32	ARM Cortex-M4		Microcontroller
ARMv7-R	32	ARM Cortex-R4, ARM Cortex-R5, ARM Cortex-R7		Real-time
ARMv7-A	32	ARM Cortex-A5, ARM Cortex-A7, ARM Cortex-A8, ARM Cortex-A9, ARM Cortex-A12, ARM Cortex-A15	Krait, Scorpion, PJ4/Sheeva, Apple A6/A6X (Swift)	Application
ARMv8-A	64/32	ARM Cortex-A53, ARM Cortex-A57 <sup>[22]</sup>	X-Gene, Denver, Apple A7 (Cyclone)	Application
ARMv8-R	32	No announcements yet		Real-time

# The ARM Register Set



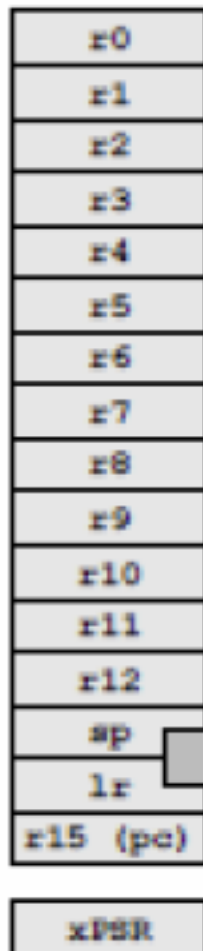
## lr

link register (lr) is a special-purpose register which holds the address to return to when a function call completes. This is more efficient than the more traditional scheme of storing return address on a call stack, sometimes called a machine stack.

The link register does not require the writes and reads of the memory containing the stack which can save a considerable percentage of execution time with repeated calls of small subroutines.

## Καταχωρητές των επεξεργαστών ARM Cortex-M

Main



Changes from standard ARM architecture:

- Stack-based exception model
- Only two processor modes
- Thread Mode for User tasks\*
- Handler Mode for OS tasks and exceptions\*
- Vector table contains addresses

\*Only SP changes between modes

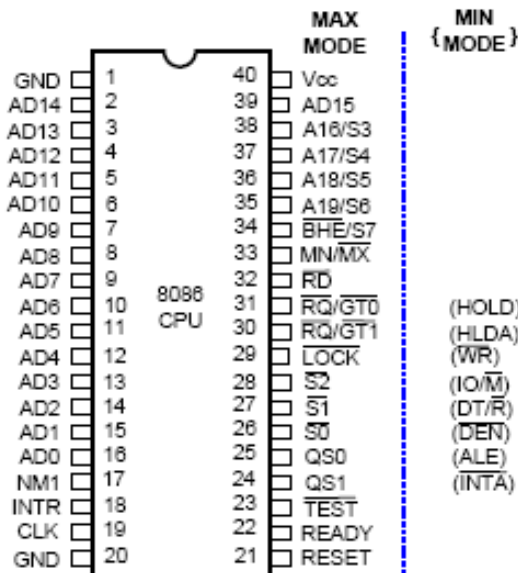
Process

sp

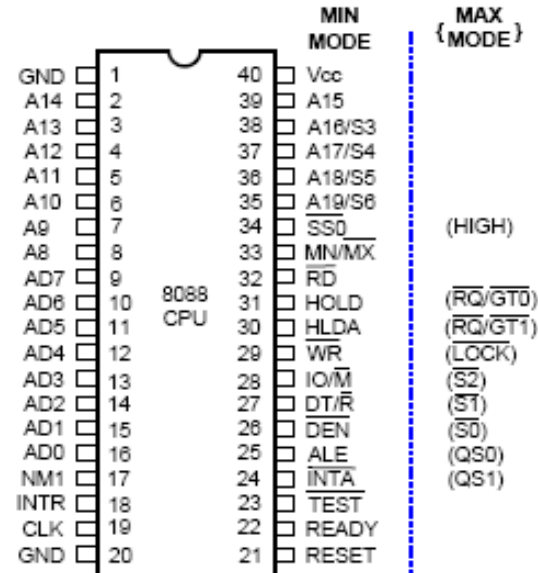
## Εντολές ADD του ARM

- For example an add instruction takes the form:
  - `ADD r0,r1,r2 ; r0 = r1 + r2 (ADDAL)`
- To execute this only if the zero flag is set:
  - `ADDEQ r0,r1,r2 ; If zero flag set then...`  
`; ... r0 = r1 + r2`

3.1. Αναφέρατε το μήκος του διαύλου διευθύνσεων, το μέγεθος του διαύλου δεδομένων και το μέγιστο μέγεθος της κύριας μνήμης που μπορεί να προσπελάσουν οι επεξεργαστές 8086 8088. Οι επεξεργαστές 8086, 8088 έχουν πολύ- πλεγμένους διαύλους διευθύνσεων και δεδομένων. Για παράδειγμα η ακίδα AD7 χρησιμοποιείται για την παραγωγή του σήματος διεύθυνσης A7 και συνέχεια σαν σήμα δεδομένων D7.



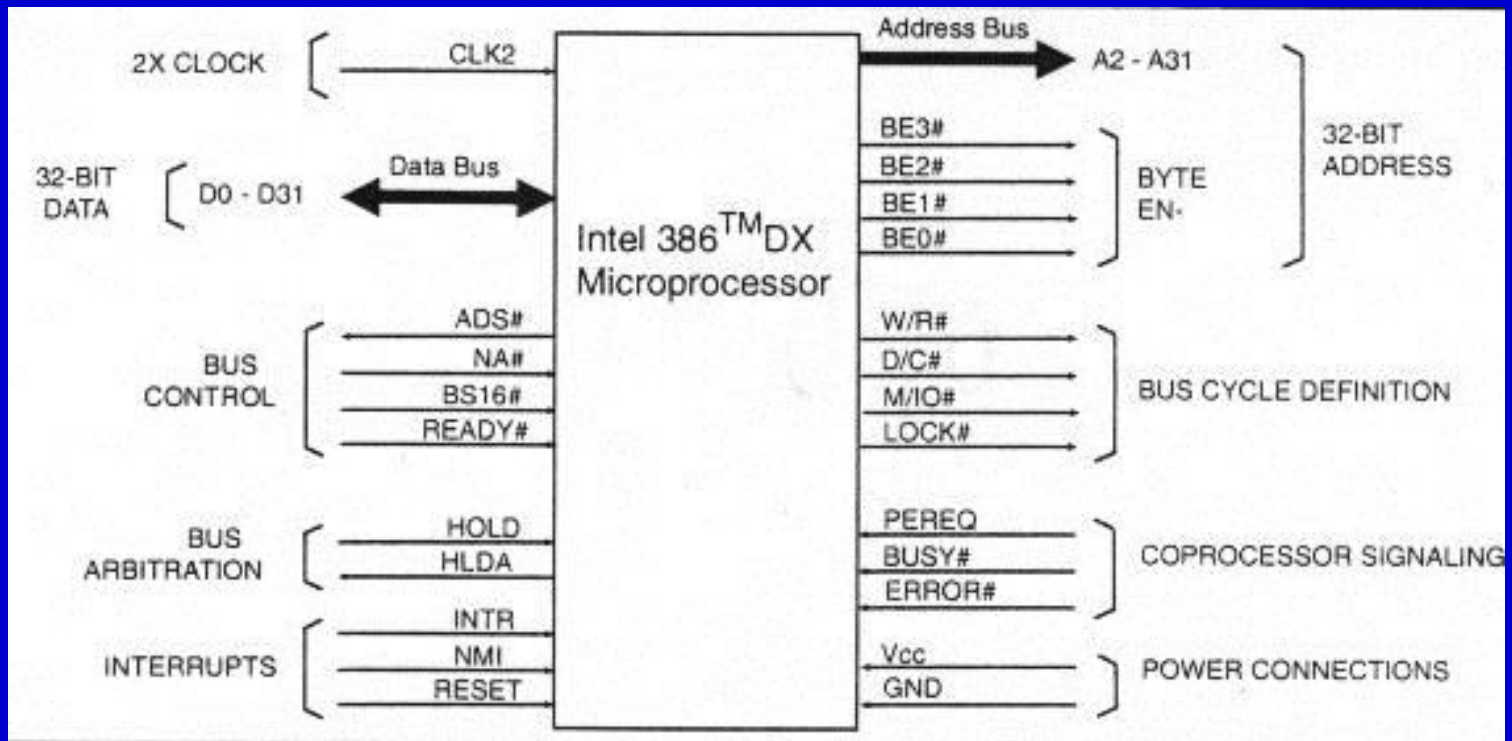
8086 pin diagram



8088 pin diagram

2. Σε έναν επεξεργαστή 8088 ο IP περιέχει τον δυαδικό αριθμό του οποίου η κωδικοποίηση στο δεκαεξαδικό σύστημα είναι 12A3H και ο CS δυαδικό αριθμό του οποίου η κωδικοποίηση είναι 111BH. Να βρεθεί η διεύθυνση της μνήμης στην οποία θα γίνει προσπέλαση ανάκλησης εντολής. Το H στο τέλος κάθε αριθμού σημαίνει ότι είναι σε δεκαεξαδική μορφή.

3. Μελετήστε το λειτουργικό διάγραμμα ακίδων του  $\mu\text{E}$  80386. Οι γραμμές διευθύνσεων A0, A1 χρησιμοποιούνται για την παραγωγή των σημάτων BE0#  $\rightarrow$  BE3#. Να υπολογισθεί το μέγιστο μέγεθος της κύριας μνήμης που μπορεί να προσπελάσει ο  $\mu\text{E}$  80386. Να υπολογιστεί επίσης το μήκος του διαύλου δεδομένων.



4. Μελετήστε την μετάφραση σε γλώσσα μηχανής του προγράμματος σε γλώσσα C που δίδεται στην συνέχεια. Εντοπίστε την στήλη με τις διευθύνσεις, την στήλη με την γλώσσα μηχανής και τις στήλες με την γλώσσα Assembly.

```

Address      Machine Code  Assembly
PUBLIC      _main
_TEXT       SEGMENT
_a$ = -4
_b$ = -8
_c$ = -12
_main       PROC NEAR; COMDAT
; 3 :{
00000       55             push  ebp
00001       8b ec          mov   ebp, esp
00003       83 ec 4c       sub   esp, 76           ; 0000004cH
00006       53             push  ebx
00007       56             push  esi
00008       57             push  edi
00009       8d 7d b4       lea  edi, DWORD PTR [ebp-76]
0000c       b9 13 00 00 00 mov   ecx, 19           ; 00000013H
00011       b8 cc cc cc cc mov   eax, -858993460   ; cccccccH
00016       f3 ab          rep  stosd
; 4 : int a, b, c;
; 5 : a=b+c;
00018       8b 45 f8       mov   eax, DWORD PTR _b$[ebp]
0001b       03 45 f4       add   eax, DWORD PTR _c$[ebp]
0001e       89 45 fc       mov   DWORD PTR _a$[ebp], eax
; 6 :}
00021       5f             pop   edi
00022       5e             pop   esi
00023       5b             pop   ebx
00024       8b e5         mov   esp, ebp
00026       5d             pop   ebp
00027       c3             ret   0
_main       ENDP
_TEXT       ENDS
END

```

5. Να υπολογισθεί στο δεκαδικό σύστημα η τιμή του αριθμού του οποίου η αναπαράσταση σε floating-point format δίδεται στην συνέχεια.

1	11110011	11000000000000000000
---	----------	----------------------

**Λύση**

$$11110011 = 243 \quad E = 243 - 127 = 116$$

$$N = (-1)^1 \times 1.11 \times 2^{116}$$

$$N = -1.75 \times 2^{116}$$

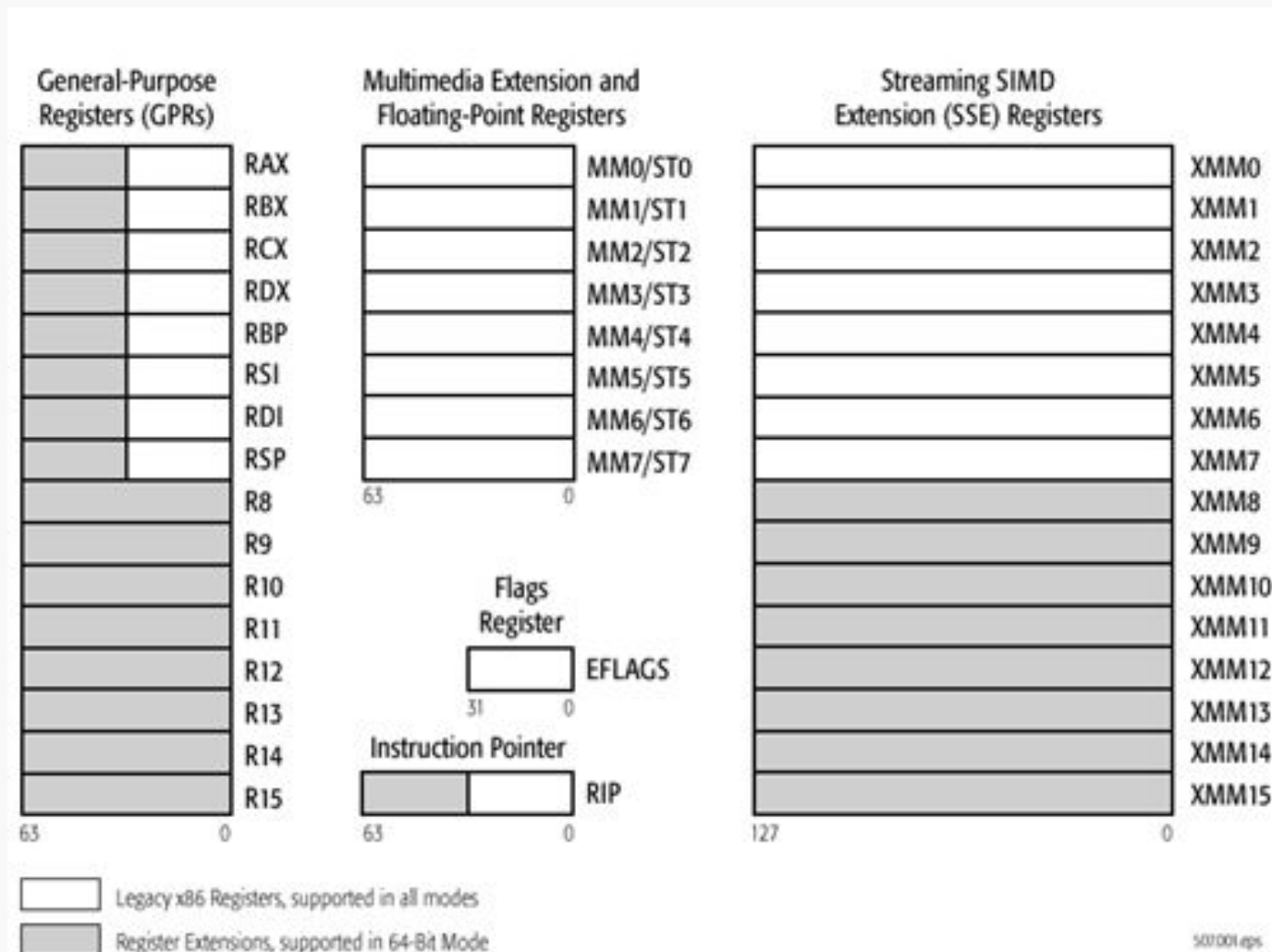
6. Σε έναν επεξεργαστή με αρχιτεκτονική SIMD εκτελείται η εντολή VADD (Vector ADD ) που δίδεται στην συνέχεια
- $$\text{VADD } A, B ; A \leftarrow A+B$$

Το περιεχόμενο των καταχωρητών δίδεται στην συνέχεια

A	5	6	9	2
B	1	4	1	1

Περιγράψτε το περιεχόμενο των καταχωρητών A, B μετά την εκτέλεση της εντολής. Πόσες ALU απαιτούνται να λειτουργούν ταυτόχρονα για την εκτέλεση της εντολής.

7. Στο μοντέλο προγραμματισμού x86-64 που δίδεται στην συνέχεια σημειώστε τους index registers, τον stack pointer, και τον program counter



8. Σε έναν επεξεργαστή ARM των 32 bit οι καταχωρητές R1, R2, R3 περιέχουν τα δεδομένα 0x00000001, 0x0000000A και 0x0000000B, εκτελείται η εντολή

ADD R1, R2, R3

Ποιο είναι το περιεχόμενο των καταχωρητών R1, R2, R3 μετά την εκτέλεση της εντολής.

```
- ADD r0, r1, r2 ; r0 = r1 + r2 (ADDAL)
```