

ΜΙΚΡΟΠΕΞΕΡΓΑΣΤΕΣ-ΜΙΚΡΟΕΛΕΓΚΤΕΣ

Υλικό (hardware) και λογισμικό (software) των Η/Υ

Υλικό (hardware) είναι το σύνολο των συσκευών και των εξαρτημάτων (πλακέτες με ολοκληρωμένα κυκλώματα, μονάδες δίσκων, συστήματα απεικόνισης, πληκτρολόγια, mouse, κλπ), που απαρτίζουν τον Η/Υ.

Λογισμικό (software) είναι το σύνολο των προγραμμάτων με τα οποία οργανώνεται η λειτουργία του Η/Υ, ώστε να καθίσταται ευχερέστερη η χρήση του και να εκτελεί τις επιθυμητές εργασίες. Το λογισμικό διακρίνεται σε **λογισμικό συστήματος** και **λογισμικό εφαρμογών**.

Λογισμικό Συστήματος και Λογισμικό Εφαρμογών

Το **λογισμικό συστήματος** (*system software*) αποτελείται από το λειτουργικό σύστημα (operating system) και τα βοηθητικά προγράμματα (utility programs) όπως editors, library routines, compilers, κλπ.

Λογισμικό εφαρμογών (*application software*) είναι το λογισμικό που έχει σχεδιασθεί ώστε να βοηθάει τον χρήστη να εκτελεί μεμονωμένες ή πολλές σχετικές μεταξύ τους εξειδικευμένες εργασίες. Παραδείγματα λογισμικού εφαρμογών είναι τα λογιστικά φύλα, το λογισμικό εφαρμογών γραφείου, το λογισμικό γραφικών και πολυμέσων, κλπ.

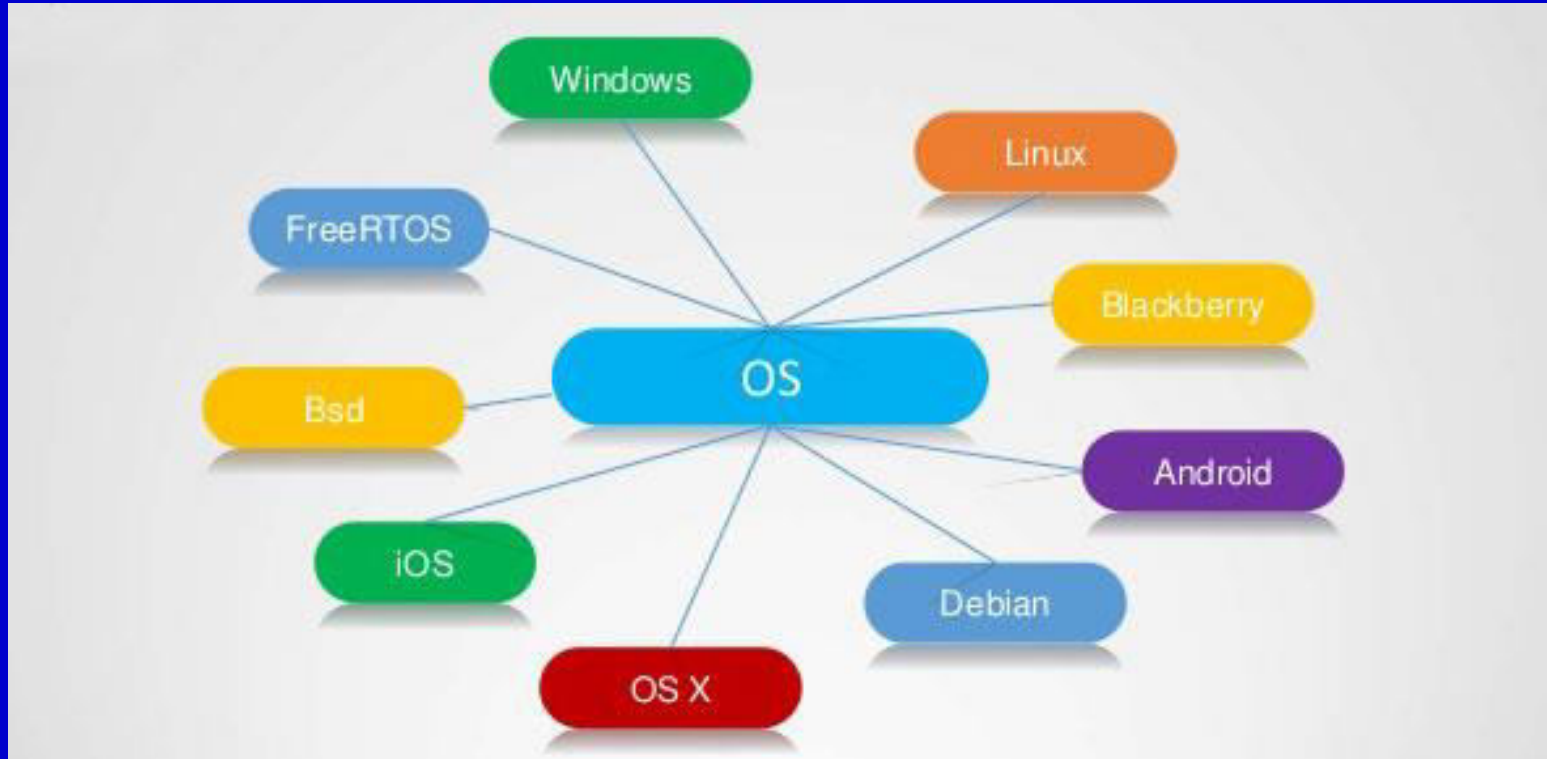
Λειτουργικό σύστημα υπολογιστικών συστημάτων

Λειτουργικό σύστημα (*operating system*) είναι το μέρος εκείνο του λογισμικού συστήματος το οποίο είναι υπεύθυνο για την καλύτερη κατανομή των εργασιών, την εκμετάλλευση του υλικού του υπολογιστή, καθώς και για την ευχερέστερη χρησιμοποίησή του.

Το λειτουργικό σύστημα προσθέτει στο υπολογιστικό σύστημα επιπλέον δυνατότητες και λειτουργίες. Για την είσοδο/έξοδο δεδομένων ή την διαχείριση μνήμης το λειτουργικό σύστημα λειτουργεί σαν ενδιάμεσο μεταξύ του υλικού και του λογισμικού των εφαρμογών, οι οποίες όταν εκτελούνται πολλές φορές καλούν το λειτουργικό σύστημα ή διακόπτονται από αυτό.

Παραδείγματα διαδεδομένων σύγχρονων λειτουργικών συστημάτων είναι τα Microsoft **Windows**, **Mac OS X**, **Unix**, **Linux**, **Android**, ...

Δημοφιλή λειτουργικά συστήματα



RTOS

Real-time systems are hardware and software systems subject to a "real-time constraint" from event to system response.

Real-time systems must guarantee response within specified time constraints. Real-time responses are often understood to be in the order of **milliseconds**, and sometimes **microseconds**.

A **real-time operating system (RTOS)** is an operating system (OS) intended to serve real-time applications that process data as it comes in, typically without buffer delays. Processing time requirements are measured in tenths of seconds or shorter increments of time.

They either are event driven or time sharing. Event driven systems switch between tasks based on their priorities, while time sharing systems switch between tasks based on clock interrupts. Most RTOS use a pre-emptive scheduling algorithm.

FreeRTOS

FreeRTOS is a real-time operating system kernel for embedded devices that has been ported to 35 microcontroller platforms. FreeRTOS is designed to be small and simple. The kernel itself consists of only three C files. To make the code readable, easy to port, and maintainable, it is written mostly in C, but there are a few assembly functions included where needed (mostly in architecture-specific scheduler routines).

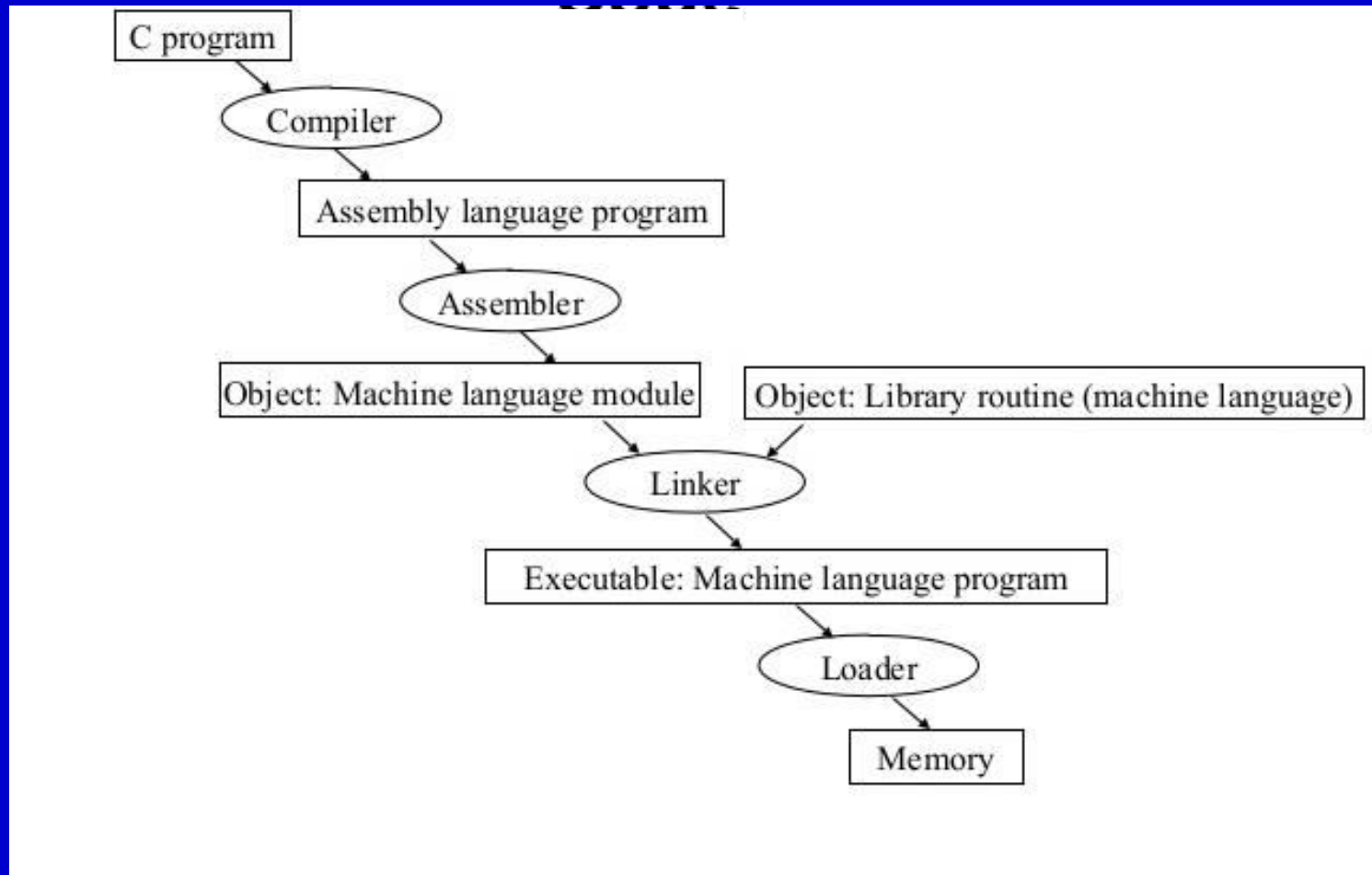
FreeRTOS provides methods for multiple threads or tasks, mutexes, semaphores and software timers. A tick-less mode is provided for low power applications. Thread priorities are supported. FreeRTOS applications can be completely statically allocated.

Mutex

In computer programming, a *mutual exclusion object* (**mutex**) is a program object that allows multiple program threads to share the same resource, such as file access, but not simultaneously.

When a program is started, a mutex is created with a unique name. After this stage, any thread that needs the resource must lock the mutex from other threads, while it is using the resource. The mutex is set to unlock when the data is no longer needed or the routine is finished.

Διαδικασία ανάπτυξης και εκτέλεσης προγραμμάτων



Βασικές μονάδες των Η/Υ

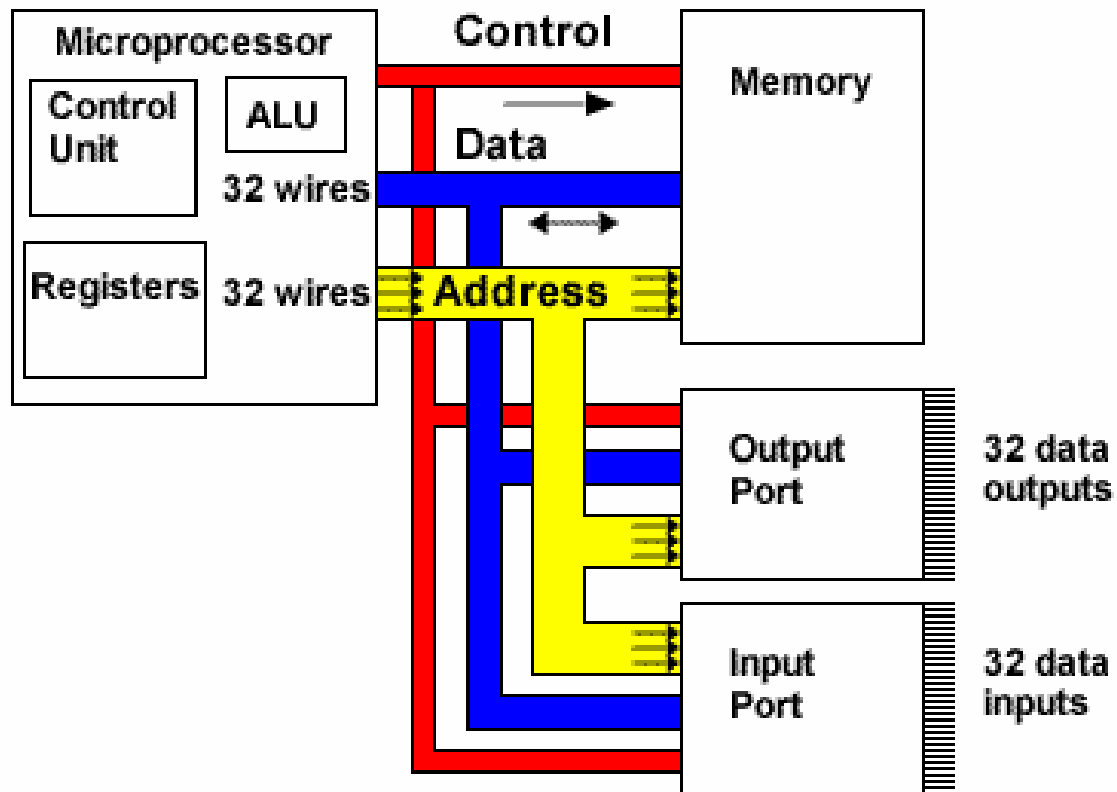
Central Processing Unit (CPU). Η μονάδα αυτή κάνει την επεξεργασία των δεδομένων και ταυτόχρονα ελέγχει την λειτουργία ολόκληρου του υπολογιστή. Ουσιαστικά ανακαλεί από την κύρια μνήμη εντολές σε γλώσσα μηχανής (σειρές από 0 και 1) και τις εκτελεί. Συχνά αναφέρεται και σαν επεξεργαστής (*processor*).

Main memory (Κύρια μνήμη). Είναι μνήμη άμεσα προσπελάσιμη από την CPU. Σε αυτήν αποθηκεύονται υπό μορφή σειρών από 0, 1 οι εντολές και τα δεδομένα των προγραμμάτων τα οποία εκτελούνται από την CPU.

Μονάδες Input/Output (I/O). Μεταφέρουν δεδομένα μεταξύ του υπολογιστή και εξωτερικών συσκευών.

Μηχανισμός διασύνδεσης. Μηχανισμός που εξασφαλίζει την επικοινωνία μεταξύ της CPU, της κύριας μνήμης και των μονάδων I/O.

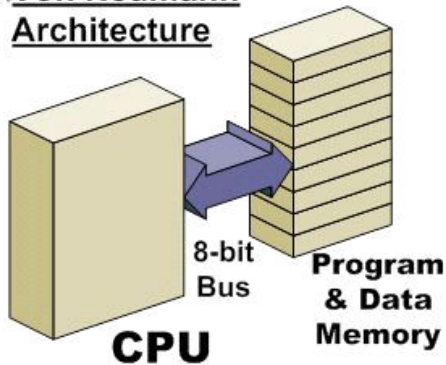
Δομή απλού υπολογιστικού συστήματος



Von Neumann and Harvard architectures



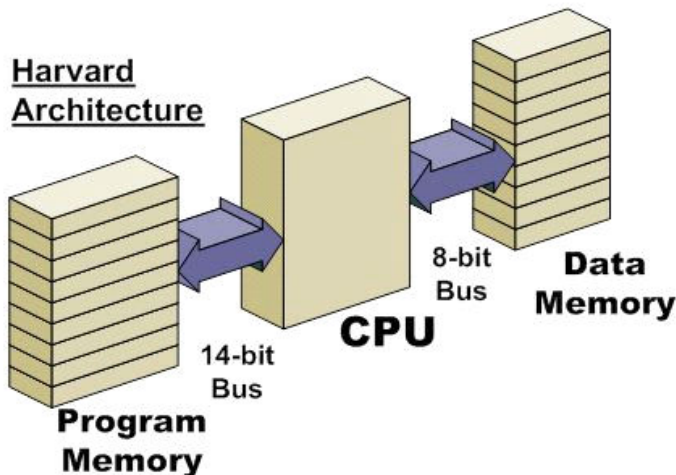
Von Neumann Architecture



- **Von Neumann Architecture:**

- Fetches instructions and data from a single memory space
- Limits operating bandwidth

Harvard Architecture



- **Harvard Architecture:**

- Uses two separate memory spaces for program instructions and data
- Improved operating bandwidth
- Allows for different bus widths

Βασικές υπομονάδες μιας κεντρικής μονάδας επεξεργασίας

Η κεντρική μονάδα επεξεργασίας (CPU) αποτελείται από το *datapath* και την μονάδα ελέγχου.

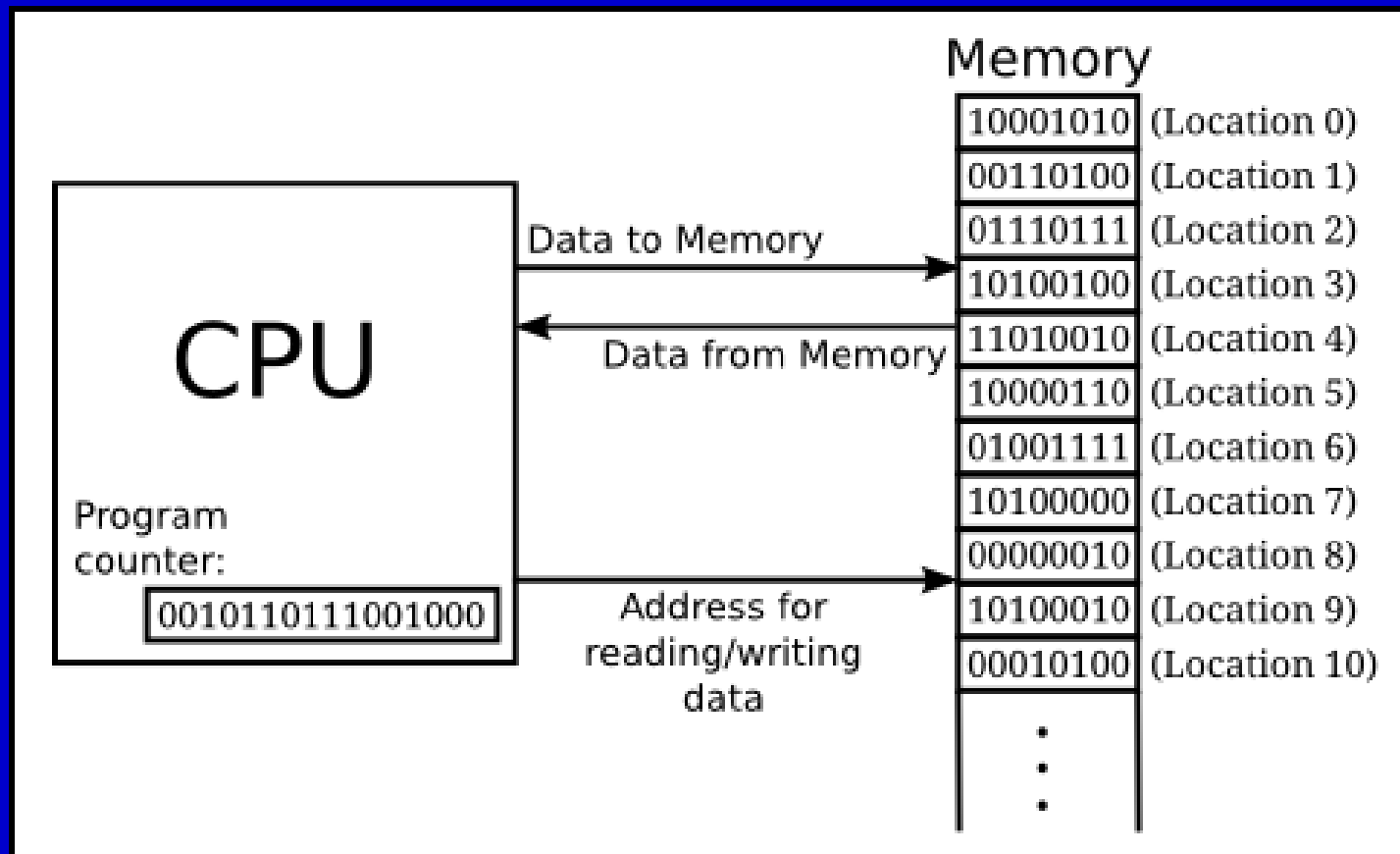
Το *datapath* αποτελείται από την αριθμητική λογική μονάδα (ALU), καταχωρητές, μονάδα επεξεργασίας αριθμών κινητής υποδιαστολής (floating point units ή FPU) και πολλές φορές την δικιά τους μονάδα ελέγχου που είναι διαφορετική από την κύρια μονάδα ελέγχου της CPU.

Η *control unit* (μονάδα ελέγχου) είναι σύνθετη μονάδα που με βάση τις εντολές γλώσσας μηχανής των προγραμμάτων που εκτελούνται ελέγχει την λειτουργία της CPU αλλά και ολόκληρου του υπολογιστή

Κύρια λειτουργία της CPU

Η *κύρια λειτουργία της CPU* είναι να ανακαλεί από την κύρια μνήμη ακολουθίες εντολών σε γλώσσας μηχανής (προγράμματα) και να τις εκτελεί. Οι εντολές γλώσσας μηχανής είναι κατάλληλες σειρές από 0 και 1. Κάθε εντολή προσδιορίζει την λειτουργία που θα εκτελεστεί από την CPU, καθώς και τα δεδομένα που θα χρησιμοποιηθούν. Προσδιορίζει επίσης άμεσα ή έμμεσα και την επόμενη εντολή που θα εκτελεστεί. Για την επικοινωνία με τις άλλες μονάδες του Η/Υ οι CPU διαθέτουν δίαυλο δεδομένων, δίαυλο διευθύνσεων, καθώς και γραμμές ελέγχου.

Λειτουργία της CPU

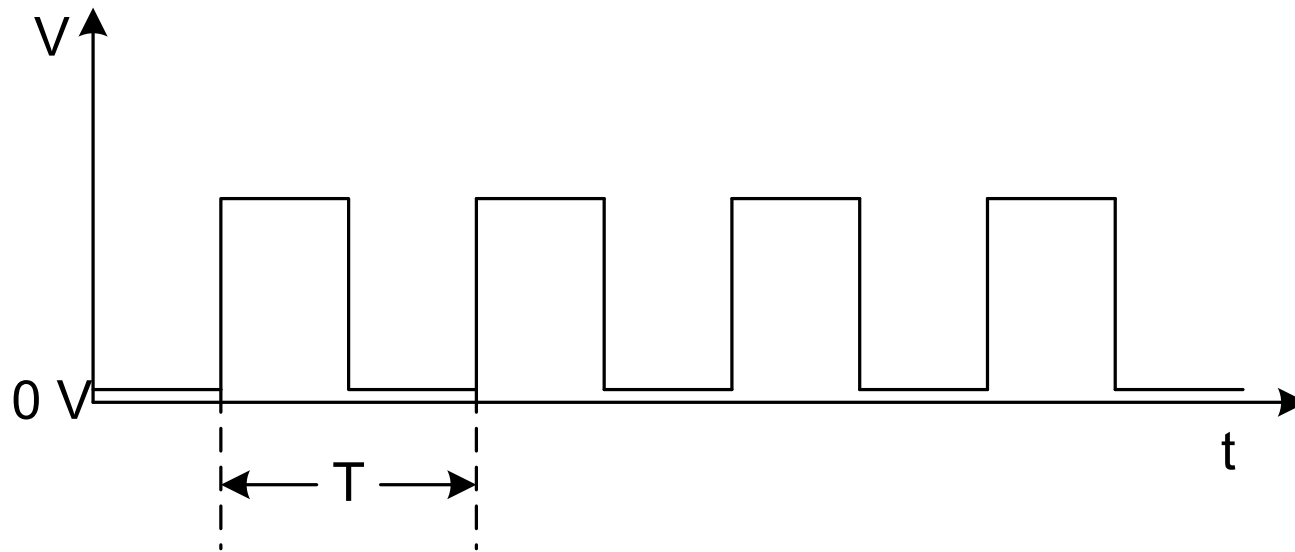


Συχνότητα και κύκλος ωρολογίου των CPU

Η CPU είναι ένα ψηφιακό ηλεκτρονικό κύκλωμα η λειτουργία του οποίου βασίζεται σε ρολόι που παράγει τετραγωνικούς παλμούς η *συχνότητα* των οποίων (δηλαδή ο αριθμός των παλμών ανά δευτερόλεπτο) καθορίζεται από έναν *κρύσταλλο χαλαζία* (Quartz).

Ο όρος *κύκλος ωρολογίου* αναφέρεται στον χρόνο που διαρκεί μία πλήρης εναλλαγή του σήματος ωρολογίου, δηλαδή στην περίοδο των ωρολογιακών παλμών. Για παράδειγμα μία CPU 200 MHz έχει ένα ρολόι που στέλνει $200 \times 10^6 = 200000000$ παλμούς το δευτερόλεπτο.

Ωρολογιακοί παλμοί



Για την μέτρηση της συχνότητας του ωρολογίου χρησιμοποιούνται μονάδες όπως KHz, MHz, GHz. Αναφέρατε την σχέση των μονάδων αυτών με το Hz.

$$1 \text{ KHz} = 10^3 \text{ Hz}$$

$$1 \text{ MHz} = 10^6 \text{ Hz}$$

$$1 \text{ GHz} = 10^9 \text{ Hz}$$

Κατά τη μέτρηση των χρόνων εκτέλεσης στοιχειωδών λειτουργιών των Η/Υ χρησιμοποιούνται μονάδες όπως ms, μ s, ns, ps. Αναφέρατε την σχέση των μονάδων αυτών με το second.

$$1 \text{ ms} = 10^{-3} \text{ sec (millisecond)}$$

$$1 \mu\text{s} = 10^{-6} \text{ sec (microsecond)}$$

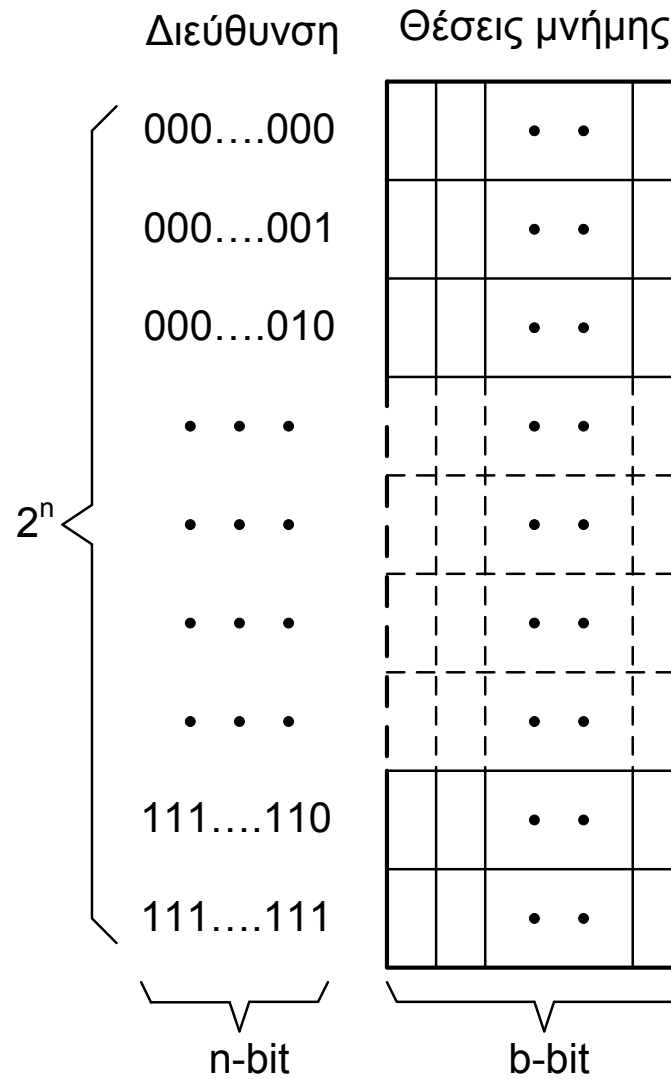
$$1 \text{ ns} = 10^{-9} \text{ sec (nanosecond)}$$

$$1 \text{ ps} = 10^{-12} \text{ sec (picosecond)} \quad (\text{sec: second})$$

Κύρια μνήμη

Η κύρια μνήμη (*main memory*) μπορεί να θεωρηθεί ότι αποτελείται από διαδοχικές θέσεις στις οποίες αποθηκεύονται πληροφορίες οι οποίες είναι συνδυασμοί δυαδικών ψηφίων. Η κάθε θέση της κυρίας μνήμης προσδιορίζεται από έναν αριθμό που καλείται *φυσική διεύθυνση* αυτής και μπορεί να προσπελασθεί άμεσα από την CPU. Στην συνέχεια δίδεται η λογική οργάνωση μιας μνήμης $2^n \times b$ -bit. Οι διευθύνσεις είναι των n bit, δηλαδή η μνήμη έχει 2^n θέσεις, ενώ οι θέσεις είναι των b bit.

Λογική οργάνωση
κύριας μνήμης



Η χωρητικότητα των μνημών μετρείται σε:

Kbyte ή KB, $1 \text{ KB} = 2^{10} \text{ byte} = 1024 \text{ byte}$,

Mbyte ή MB, $1 \text{ MB} = 2^{20} \text{ byte}$,

Gbyte ή GB, $1 \text{ GB} = 2^{30} \text{ byte}$,

Tbyte ή TB, $1 \text{ TB} = 2^{40} \text{ byte}$.

Ή αντίστοιχα σε:

Kbit ή Kb, $1 \text{ Kb} = 2^{10} \text{ bit} = 1024 \text{ bit}$,

Mbit ή Mb, $1 \text{ Mb} = 2^{20} \text{ bit}$,

Gbit ή Gb, $1 \text{ Gb} = 2^{30} \text{ bit}$,

Tbit ή Tb, $1 \text{ Tb} = 2^{40} \text{ bit}$.

Ημιαγωγικές μνήμες

Οι ημιαγωγικές μνήμες τυχαίας προσπέλασης είναι διδιάστατες διατάξεις αποθήκευσης δυαδικών ψηφίων (0 ή 1) στις οποίες κάθε φορά γίνεται προσπέλαση μιας σειράς από bit. Οι μνήμες αυτές κατασκευάζονται υπό μορφή ολοκληρωμένων κυκλωμάτων.

Διακρίνονται σε μνήμες ανάγνωσης/εγγραφής που έχει επικρατήσει να ονομάζονται RAM (Random Access Memories) και μνήμες μόνο για ανάγνωση που ονομάζονται ROM (Read Only Memories).

Τύποι RAM

Οι *στατικές μνήμες RAM (SRAM)* αποτελούνται από κύτταρα μνήμης τα οποία λειτουργούν όπως τα flip-flop. Τα δυαδικά δεδομένα που αποθηκεύονται σε αυτές διατηρούνται όσο τροφοδοτούνται με ηλεκτρική τάση.

Οι *δυναμικές μνήμες (DRAM)* αποτελούνται από κύτταρα μνήμης τα οποία βασίζονται σε πυκνωτές και αποθηκεύουν τα δεδομένα υπό μορφή φορτίων. Η παρουσία, ή η απουσία φορτίου στον πυκνωτή ερμηνεύονται ανάλογα σαν 0 ή 1. Επειδή οι πυκνωτές των κυττάρων μνήμης εκφορτίζονται με την πάροδο του χρόνου οι μνήμες αυτές απαιτούν περιοδική ανανέωση του περιεχομένου τους.

Τύποι ROM

Mask ROM. Στις μνήμες αυτές τα δεδομένα εγγράφονται κατά την διαδικασία κατασκευής τους.

PROM (Programmable ROM). Οι PROM είναι ROM στις οποίες μπορούν να εγγραφούν δεδομένα μόνο μία φορά από τους χρήστες τους. Για να εγγραφούν δεδομένα σε μία PROM χρειάζεται μία ειδική συσκευή που λέγεται *PROM programmer* ή *PROM burner*.

EPROM (Erasable PROM). Οι EPROM είναι τύπος PROM οι οποίες μπορούν να προγραμματισθούν από τους χρήστες τους πάνω από μία φορά. Για να προγραμματισθούν πρέπει πρώτα να σβήσει το περιεχόμενό τους με έκθεσή σε υπεριώδες (ultraviolet) φως για συγκεκριμένο χρονικό διάστημα. Αυτό δικαιολογεί την ύπαρξη υάλινου παραθύρου στα chip των EPROM.

EEPROM. Οι μνήμες EEPROM (Electrically Erasable PROM) είναι παρόμοιες με τις EPROM, αλλά τα περιεχόμενά τους σβήνονται ηλεκτρικά. Η διαγραφή και η εγγραφή δεδομένων γίνεται ανά byte.

Flash Memory. Τύπος EEPROM που σβήνει και επανεγγράφεται πολύ γρήγορα (flash). Διακρίνονται σε τύπου NAND και τύπου NOR. Οι μνήμες flash τύπου NAND μπορεί να διαβαστούν και να γραφούν ανά block που γενικά είναι μικρότερα από την χωρητικότητά τους. Οι σύγχρονες μνήμες flash NOR διαιρούνται σε διαγραφόμενα τμήματα (blocks). Η λειτουργία διαγραφής γίνεται ανά block. Οι μνήμες flash τύπου NOR επιτρέπουν μία απλή λέξη (byte) να γραφεί σε μία διαγεγραμμένη θέση ή να διαβαστεί ανεξάρτητα.

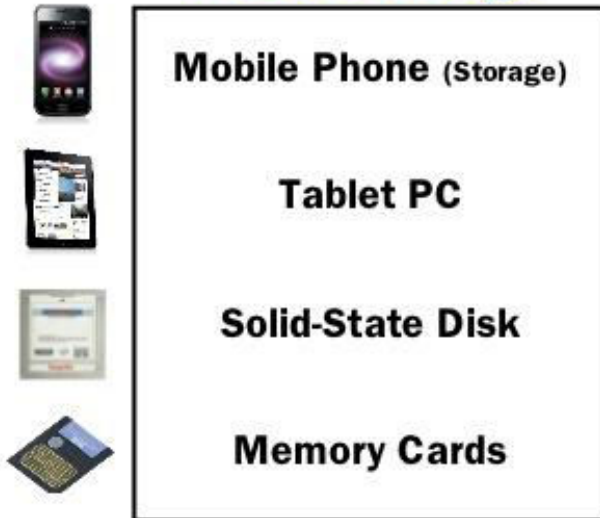
SEMICONDUCTOR MEMORY CLASSIFICATION

Read-Write Memory		Non-Volatile Read-Write Memory	Read-Only Memory
Random Access	Non-Random Access	EPROM E ² PROM	Mask-Programmed Programmable (PROM)
SRAM DRAM	FIFO LIFO Shift Register CAM	FLASH	

NAND & NOR application

NAND

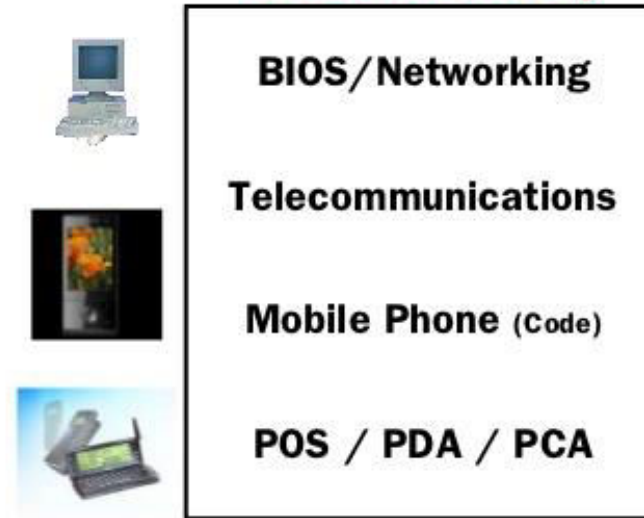
Mass Storage



| Memory Technology |
• **Low Cost and High Density**
• **Page Mode Program**
| Flash Device Technology |

NOR

Code Memory



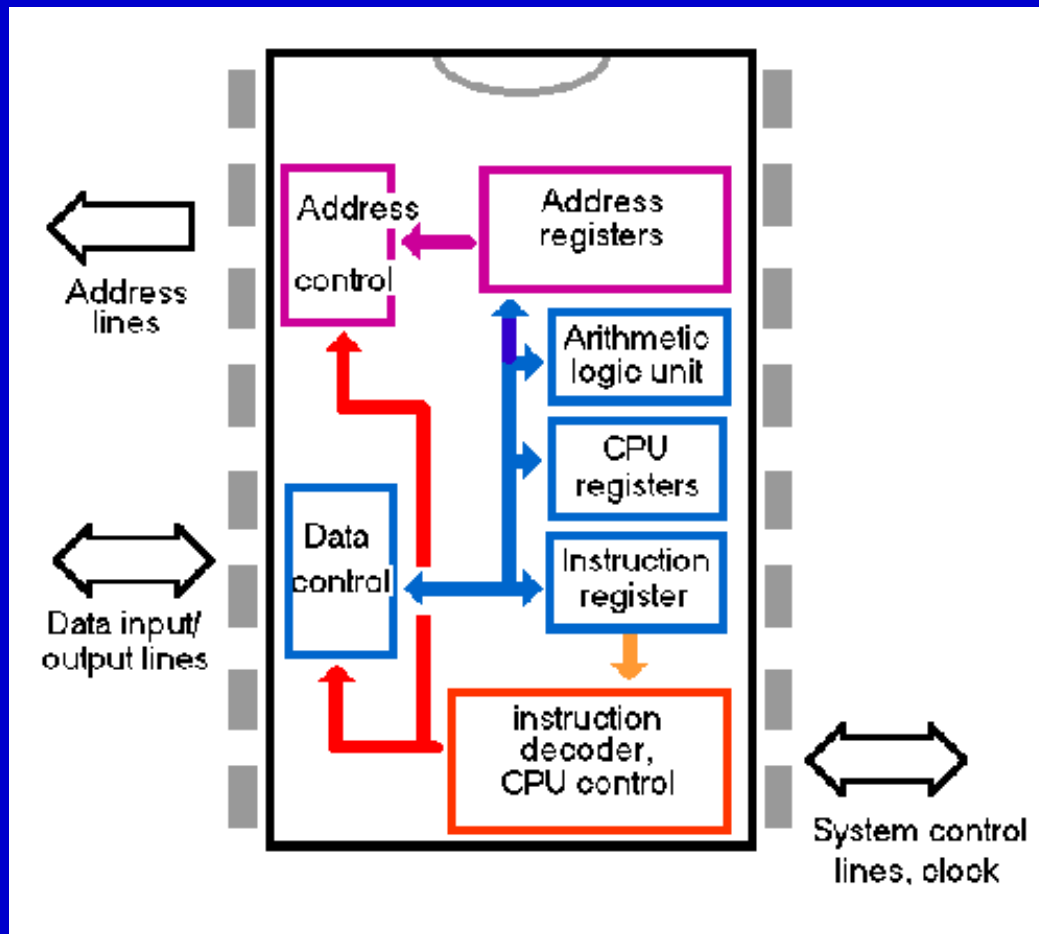
• **Fast Random Access**
• **Fast Read Speed**

Microprocessor

A ***microprocessor*** incorporates the functions of a computer's central processing unit (CPU) on a single integrated circuit (IC). The microprocessor is a multipurpose, clock driven, register based, programmable electronic device which accepts digital or binary data as input, processes it according to instructions stored in its memory, and provides results as output.

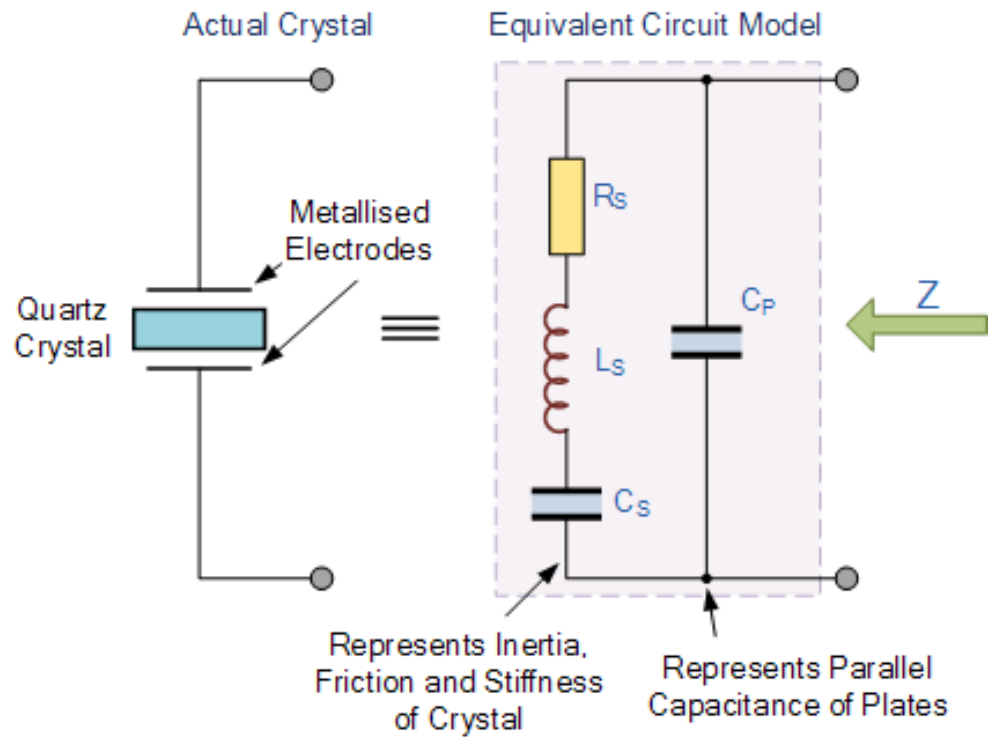
Microprocessors contain both combinational logic and sequential digital logic. Microprocessors operate on numbers and symbols represented in the binary numeral system.

Δομή μικροεπεξεργαστή

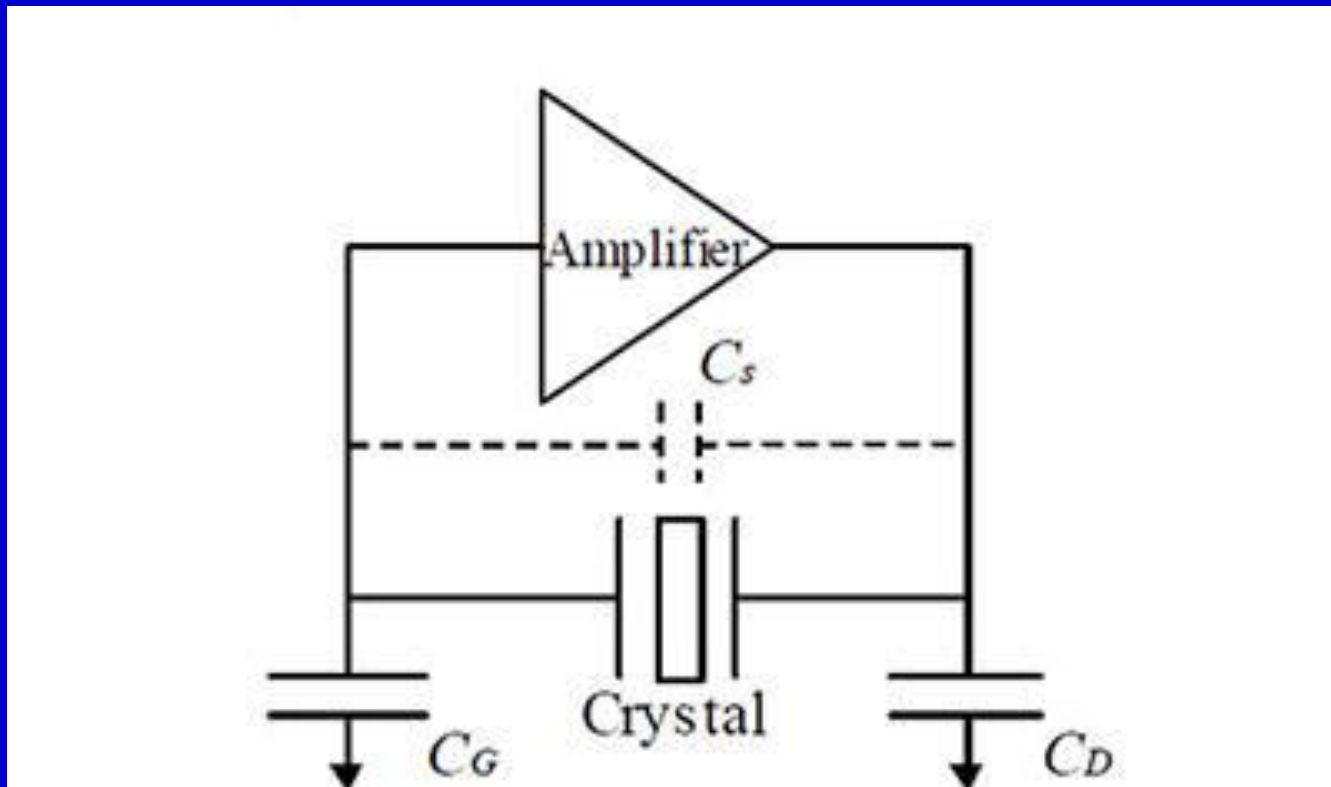


Κρύσταλλος χαλαζία

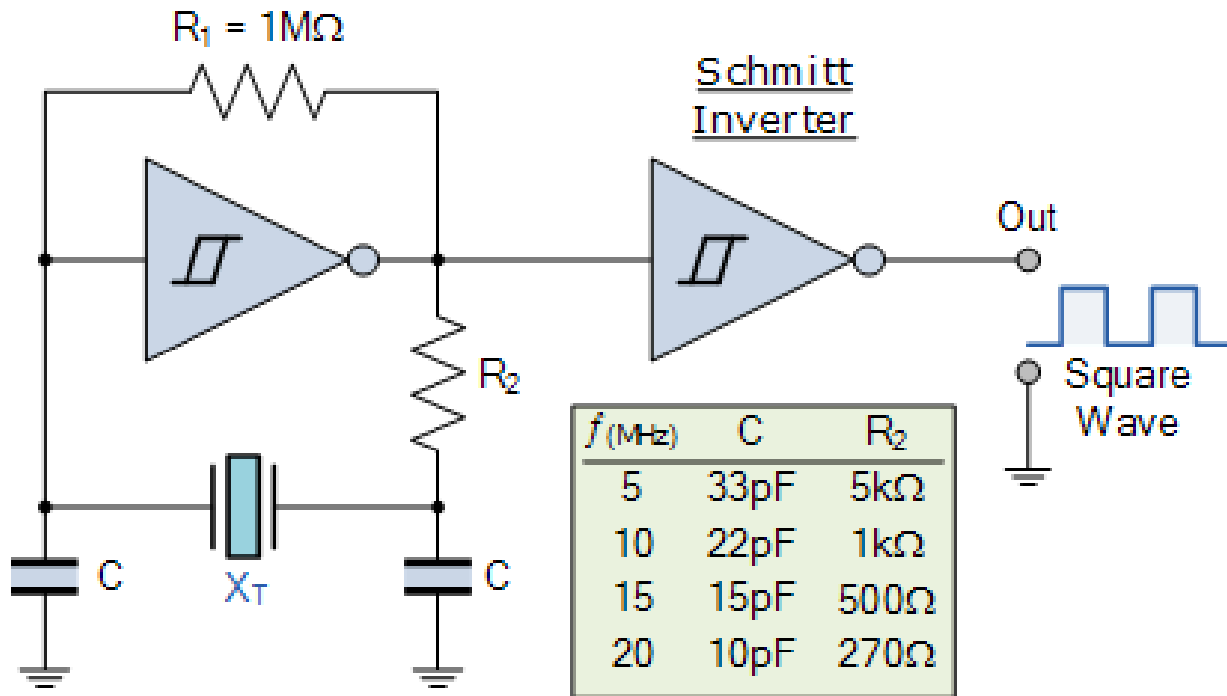
Σύμβολο και ισοδύναμο κύκλωμα



Αρχή ταλαντωτή βασισμένου σε κρύσταλλο

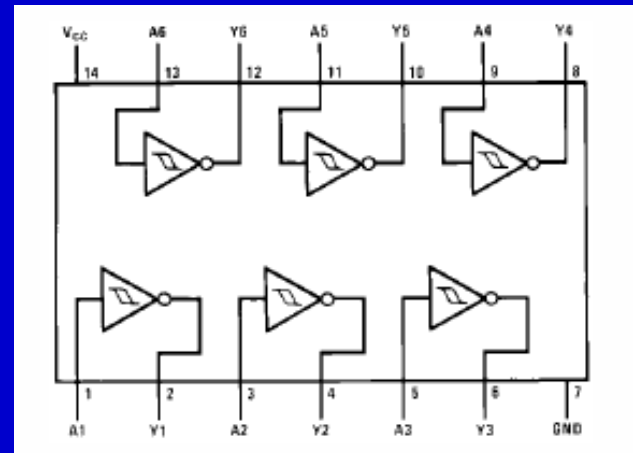
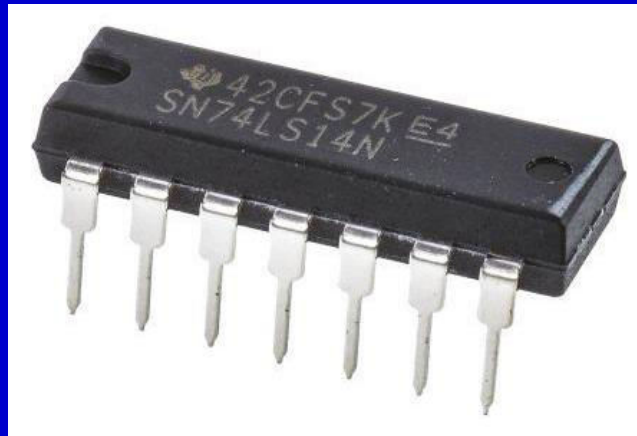


Crystal oscillator



74LS14: Schmitt trigger Inverter

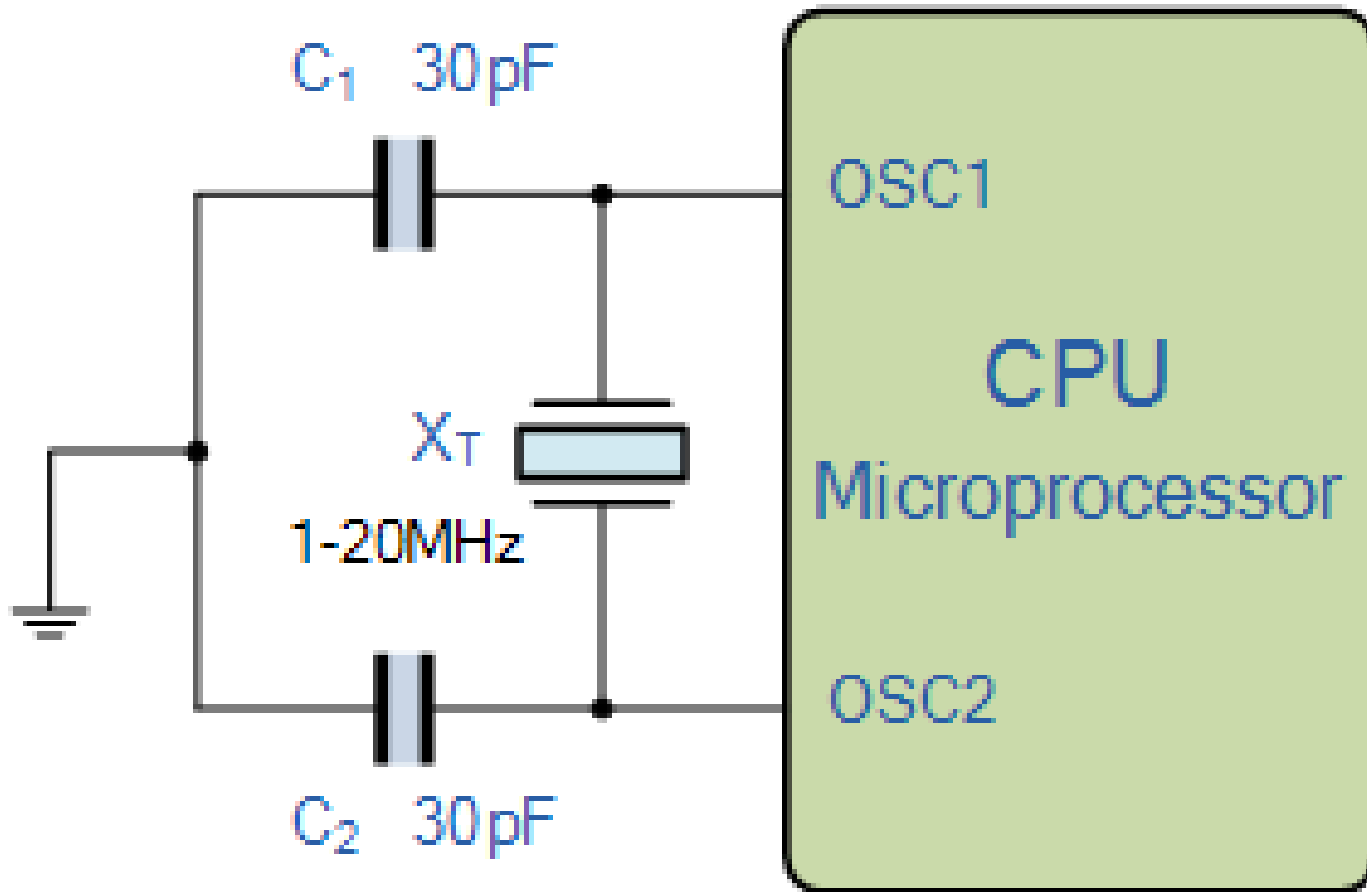
74LS14



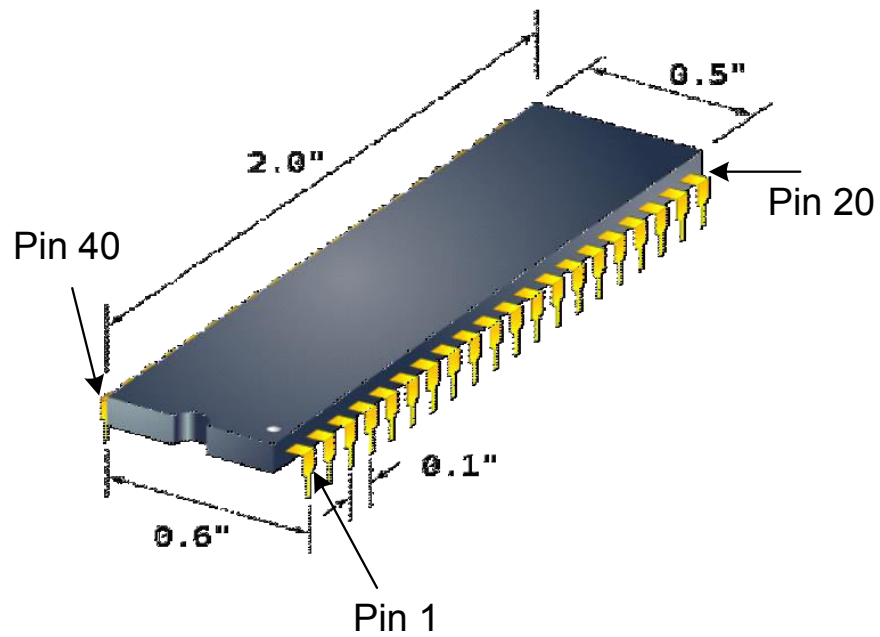
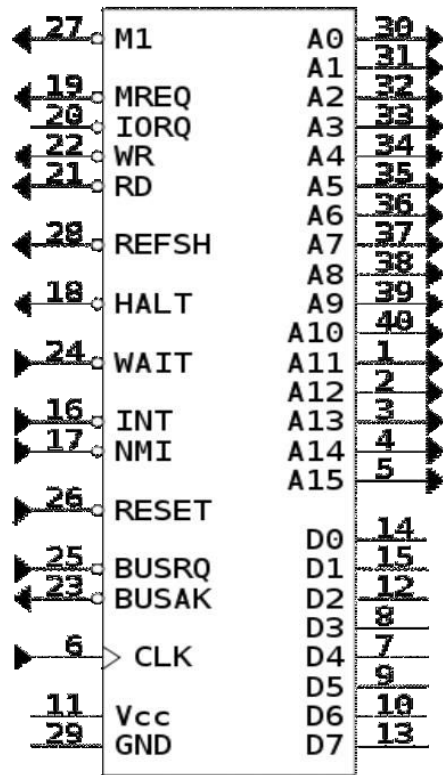
Symbol	Parameter	Min	Nom	Max	Units
V_{CC}	Supply Voltage	4.75	5	5.25	V
V_{T+}	Positive-Going Input Threshold Voltage (Note 2)	1.4	1.6	1.9	V
V_{T-}	Negative-Going Input Threshold Voltage (Note 2)	0.5	0.8	1	V
HYS	Input Hysteresis (Note 2)	0.4	0.8		V
I_{OH}	HIGH Level Output Current			-0.4	mA
I_{OL}	LOW Level Output Current			8	mA
T_A	Free Air Operating Temperature	0		70	°C

Note 2: $V_{CC} = 5V$.

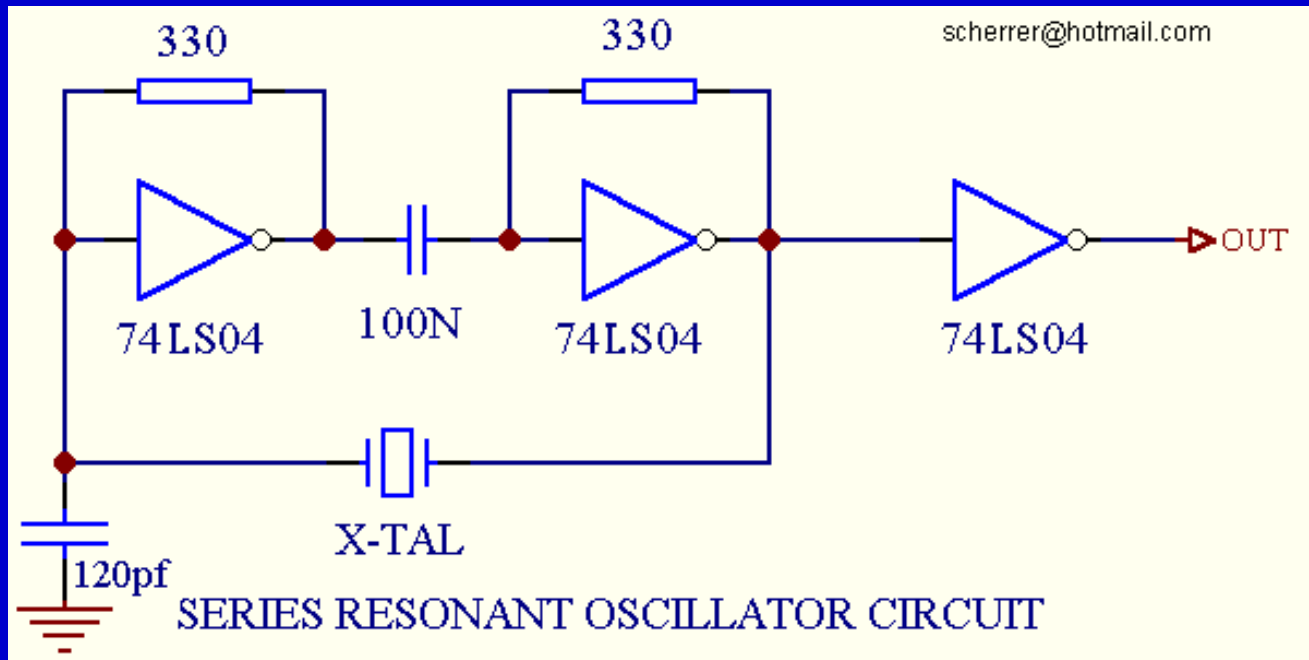
Μικροεπεξεργαστής μόνο με εξωτερικό κρύσταλλο



Μικροεπεξεργαστής Z80



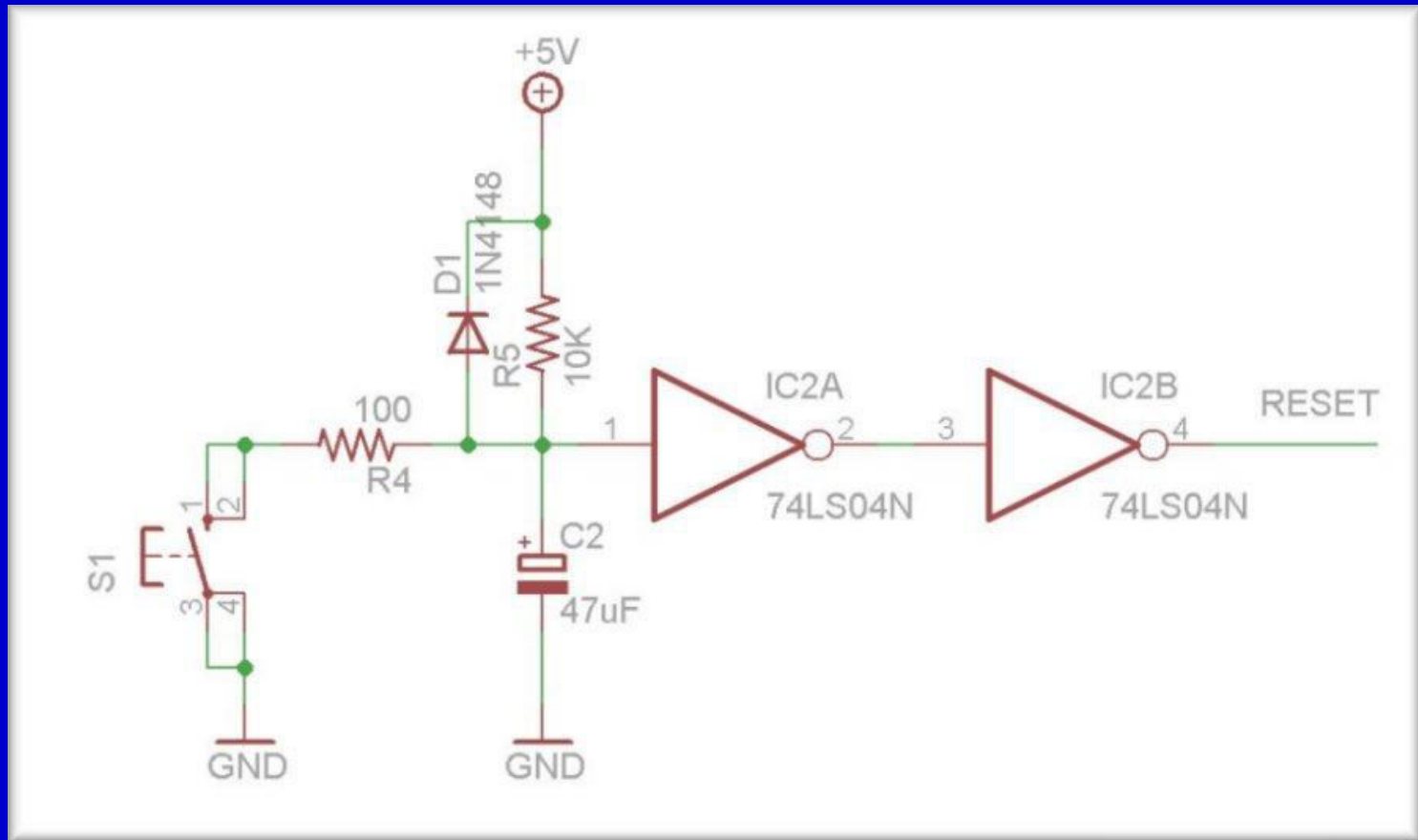
Κύκλωμα ωρολογίου του Z80



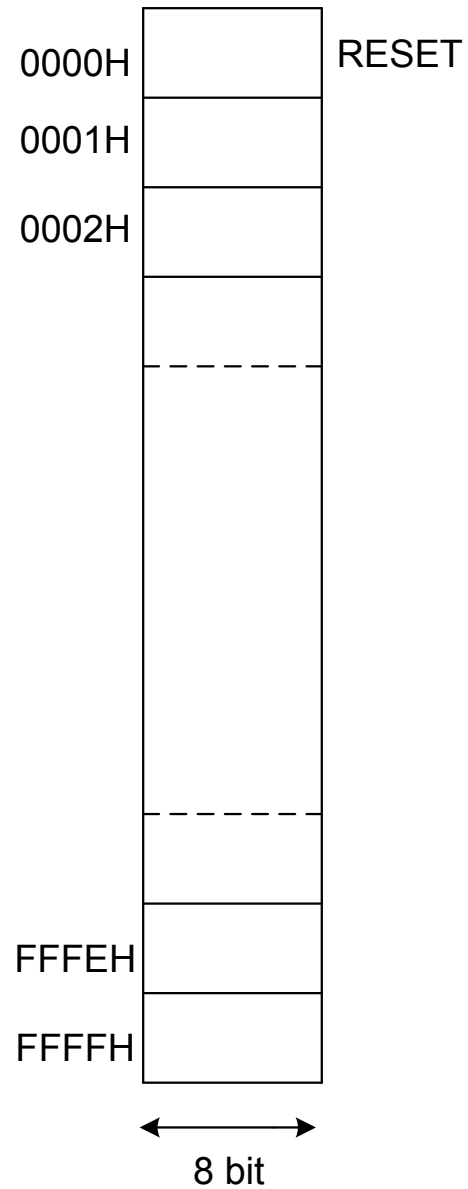
Σήμα Reset του Z80

Για την αρχικοποίηση του μΕ Z80 χρησιμοποιείται το σήμα RESET του οποίου η ενεργοποίηση **μηδενίζει τον PC (PC=0000H) και εκτελείται η εντολή που βρίσκεται σε αυτή την θέση.** Το σήμα RESET για να προκαλέσει αρχικοποίηση του Z80 πρέπει να παραμείνει στο 0 για τρεις κύκλους του ωρολογίου. Για την παραγωγή του χρησιμοποιείται ειδικό κύκλωμα.

Κύκλωμα Reset του Z80



Χάρτης μνήμης του Z80



Hardware interrupts

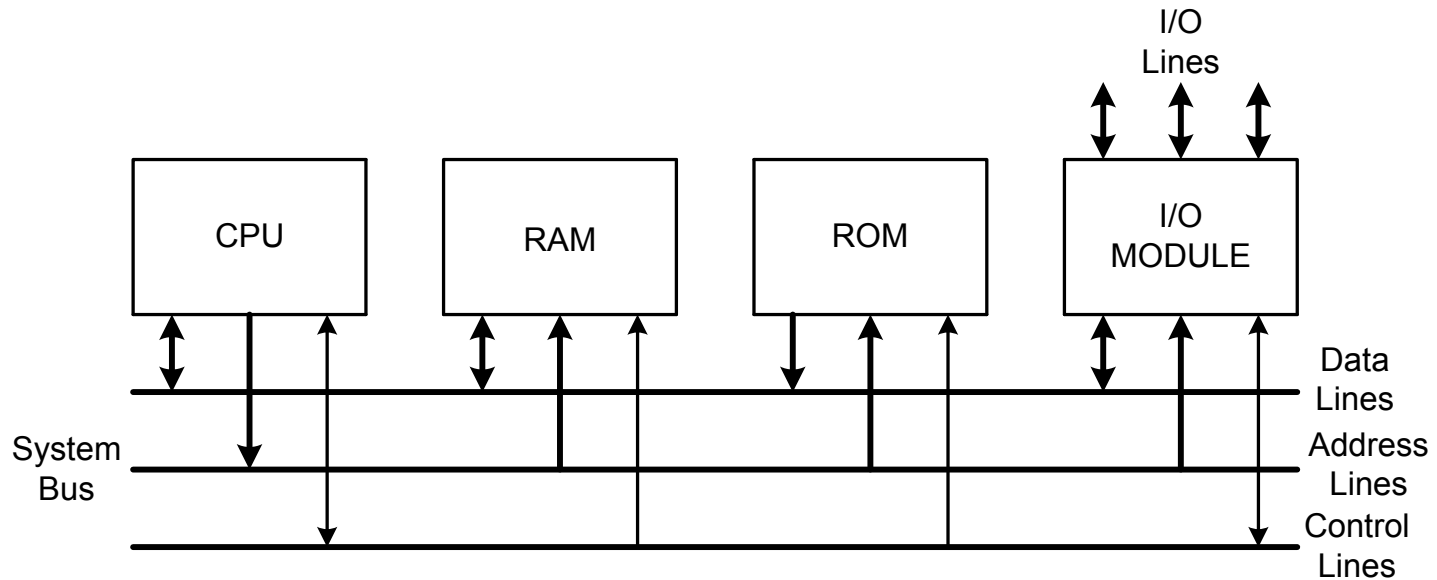
Τα *hardware interrupts* (σήματα διακοπής) είναι σήματα ελέγχου των CPU των οποίων η ενεργοποίηση διακόπτει την εκτέλεση του προγράμματος που εκτελείται εκείνη την στιγμή και προκαλεί την εκτέλεση ενός "interrupt service" προγράμματος. Μετά το πέρας του interrupt service προγράμματος η CPU συνεχίζει την εκτέλεση του προγράμματος του οποίου η εκτέλεση διακόπηκε από το interrupt. Συνήθως τα σήματα διακοπής προέρχονται από μονάδες I/O.

Σήματα Interrupts του Z80

Ο Z80 αναγνωρίζει εάν είναι ενεργοποιημένο το σήμα διακοπής INT στο τέλος της εκτέλεσης κάθε εντολής. Ο Z80 προγραμματίζεται ώστε να αποκρίνεται στο σήμα διακοπής με τρεις διαφορετικούς τρόπους (mode 0, mode 1, mode 2).

Σε **mode 1** στο τέλος της εκτέλεσης κάποιας εντολής όταν το σήμα διακοπής είναι ενεργοποιημένο δεν εκτελείται η επόμενη εντολή αλλά η εντολή που βρίσκεται στην διεύθυνση **0038H** ($PC \leftarrow 0038H$), ενώ ο PC αποθηκεύεται στην κορυφή της μνήμης stack και η τιμή του SP μειώνεται κατά 2.

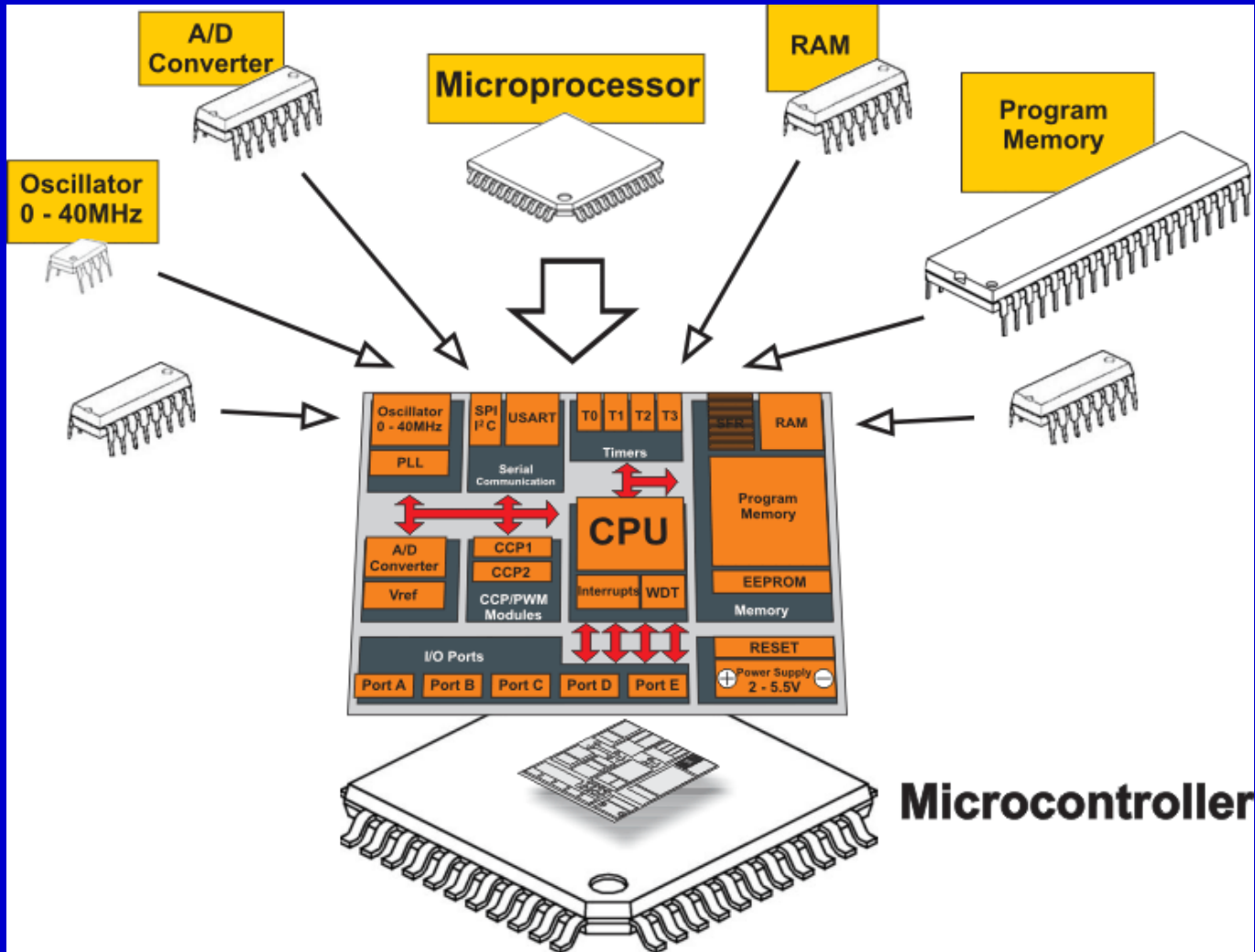
Διασύνδεση των βασικών μονάδων των Η/Υ μέσω διαύλων



Μικροελεγκτές

A ***microcontroller*** (or MCU, short for *microcontroller unit*) is a small computer on a single integrated circuit (SoC) containing a processor core, memory, and programmable input/output peripherals. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips.

Microcontroller architecture



Αρχιτεκτονικές μικροελεγκτών

Atmel AVR

Microchip's PIC

Intel's 8051

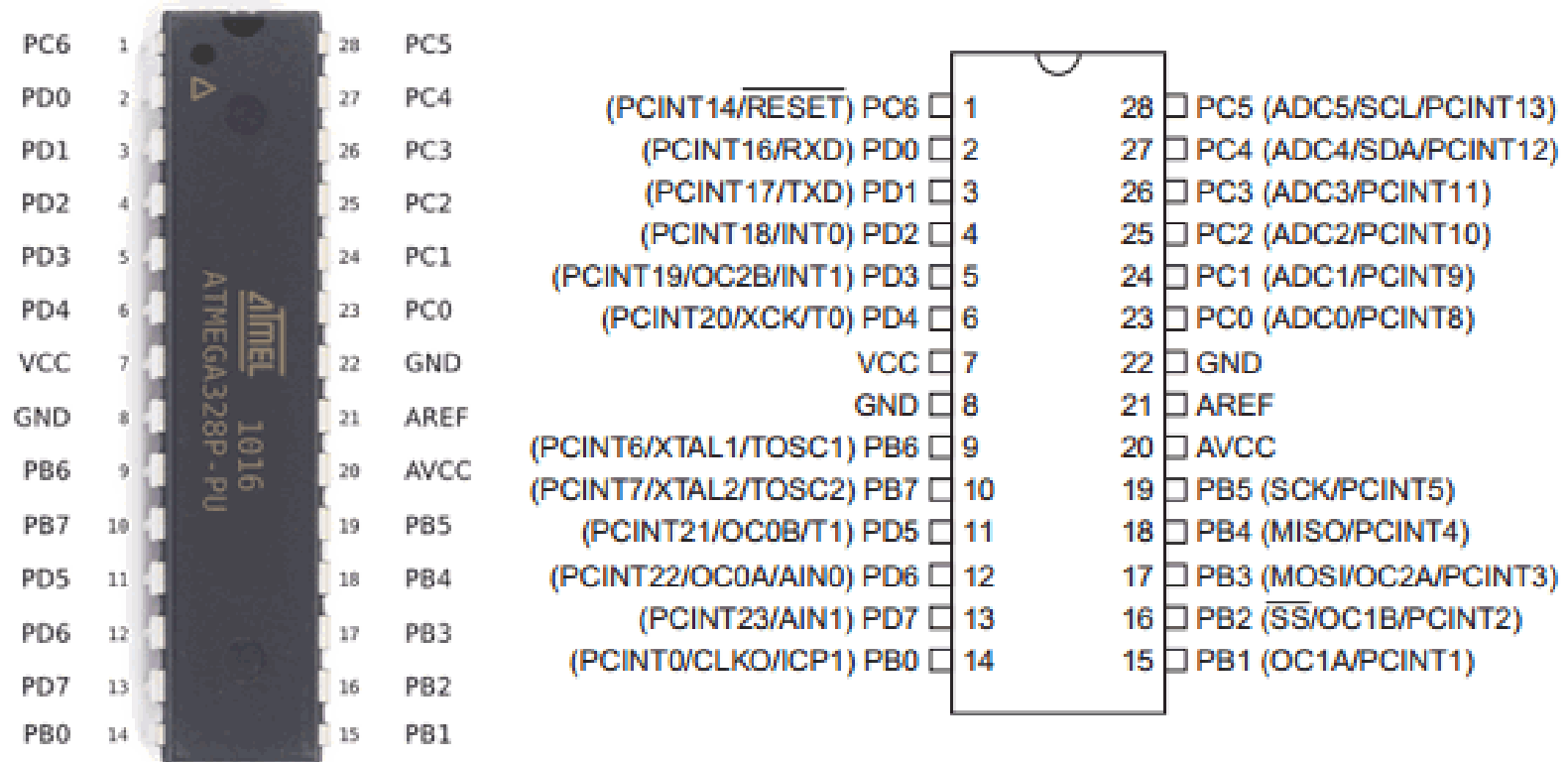
ARM

ARM architecture (32-bit)

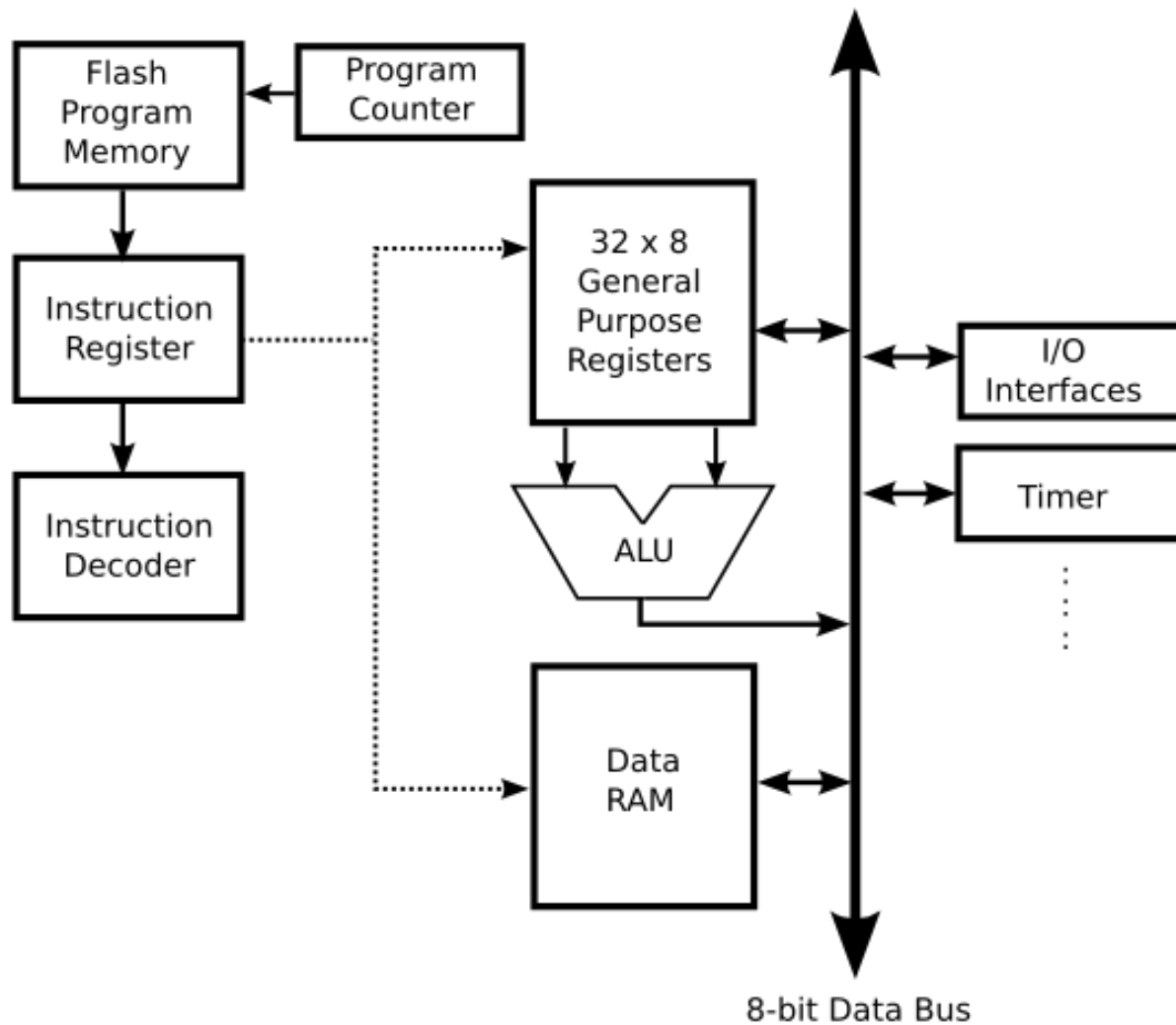
ARM64 (64/32-bit)

Μικροελεγκτής ATmega328P

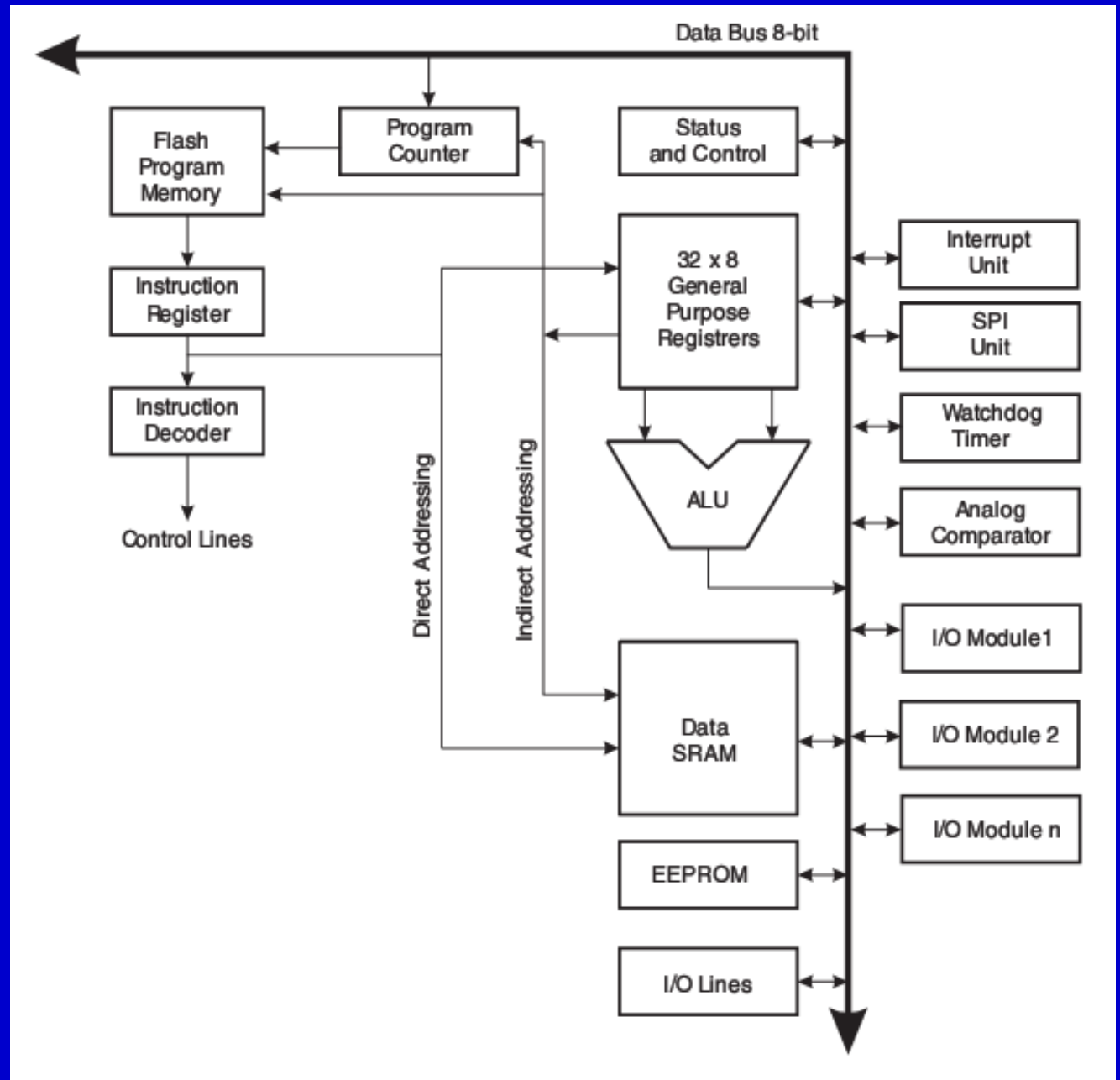
ATmega328P pin mapping



Δομή μικροελεγκτή AVR



Δομή του μικροελεγκτή AVR



Καταχωρητές του μικροελεγκτή AVR

	7	0	Addr.	
General Purpose Working Registers		R0	\$00	
		R1	\$01	
		R2	\$02	
		...		
		R13	\$0D	
		R14	\$0E	
		R15	\$0F	
		R16	\$10	
		R17	\$11	
		...		
		R26	\$1A	X-register Low Byte
		R27	\$1B	X-register High Byte
		R28	\$1C	Y-register Low Byte
		R29	\$1D	Y-register High Byte
		R30	\$1E	Z-register Low Byte
		R31	\$1F	Z-register High Byte

12.4 Interrupt Vectors in ATmega328 and ATmega328P

Table 12-6. Reset and Interrupt Vectors in ATmega328 and ATmega328P

VectorNo.	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x0000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B

Εντολή πρόσθεσης του AVR

ADD

Adds two registers without the C Flag and places the result in the destination register Rd.

Operation: $Rd, \leftarrow Rd + Rr$

Syntax: ADD Rd, Rr

Operands: $0 \leq d \leq 31$ $0 \leq r \leq 31$

Flags: H, S, V, N, Z, C

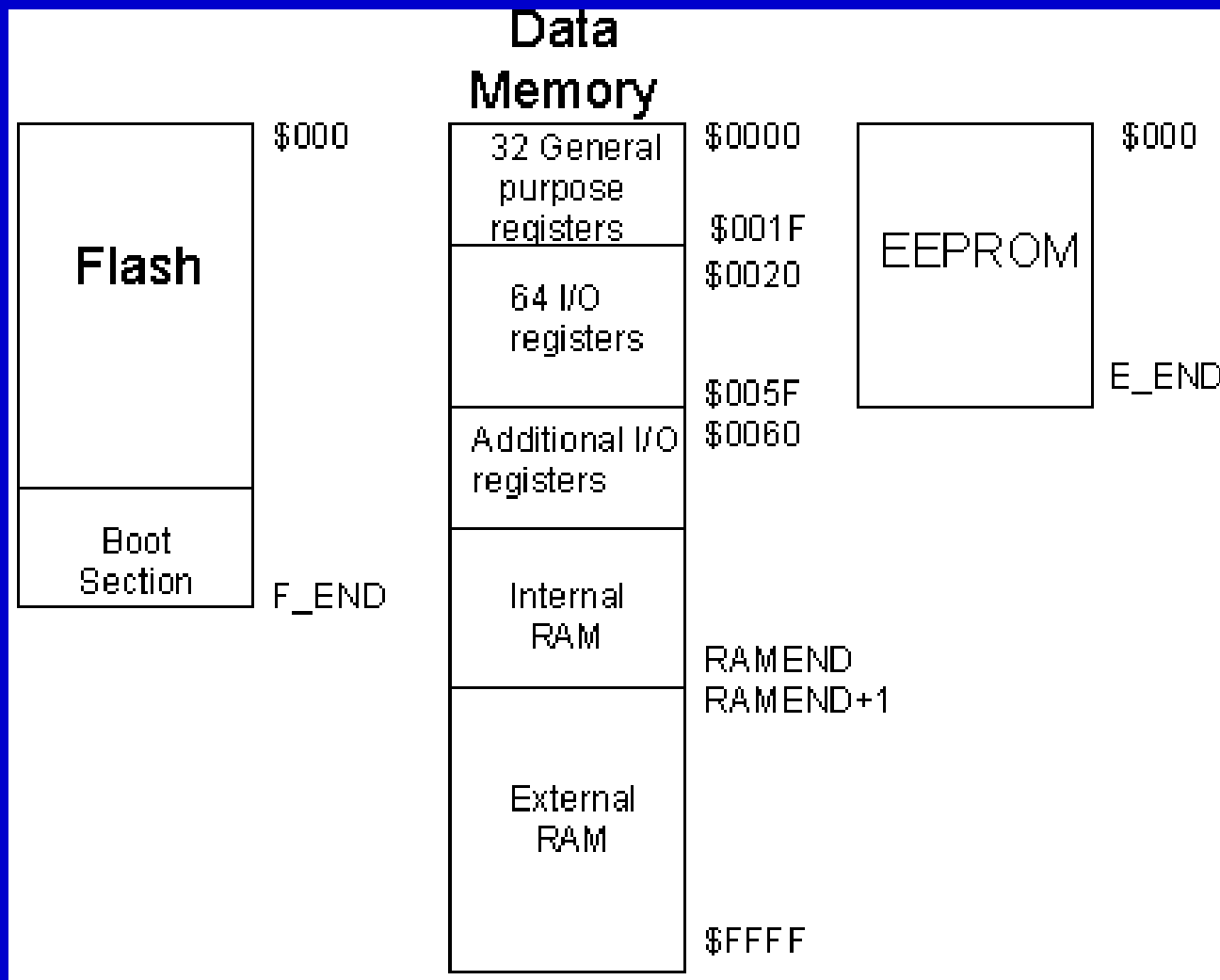
Words: 1

Cycles: 1

16-bit Opcode:

0000	11rd	dddd	rrrr
------	------	------	------

Χάρτης μνήμης των μικροελεγκτών AVR



Memory of AVR ATmega328

Flash memory: The programs are stored here. The ATmega328 has **32 KB** of Flash program memory. This memory supports at least **10 000 writes** or upload cycles.

RAM: Its content disappears when you turn off the power, but it can be read and written really fast. Every normal variable in your sketch is held in RAM while your sketch runs. The ATmega328 has **2 KB** of RAM.

EEPROM: it's an older technology to implement rewritable non-volatile memory. It's normally used to store settings and other parameters between resets. The ATmega328 has **1 KB** of EEPROM. This memory supports at least **100 000 writes**. You can use the EEPROM library in the stock Arduino library to read from and write to this memory.

Χαρακτηριστικά των μικροελεγκτών AVR

Comparison Between Processors

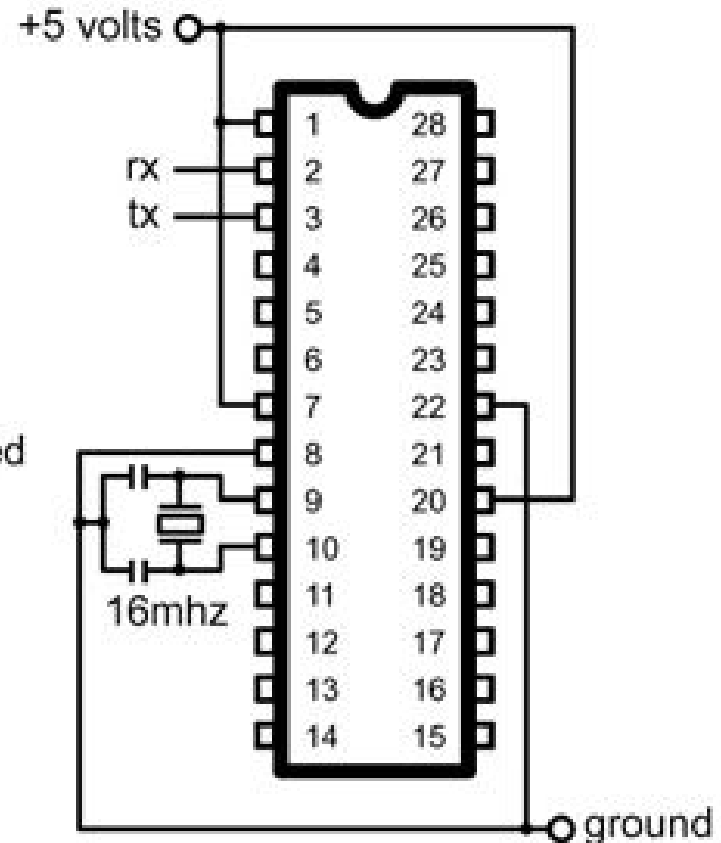
The ATmega48A/PA/88A/PA/168A/PA/328/P differ only in memory sizes, boot loader support, and interrupt vector sizes. [Table 2-1](#) summarizes the different memory and interrupt vector sizes for the devices.

Table 2-1. Memory Size Summary

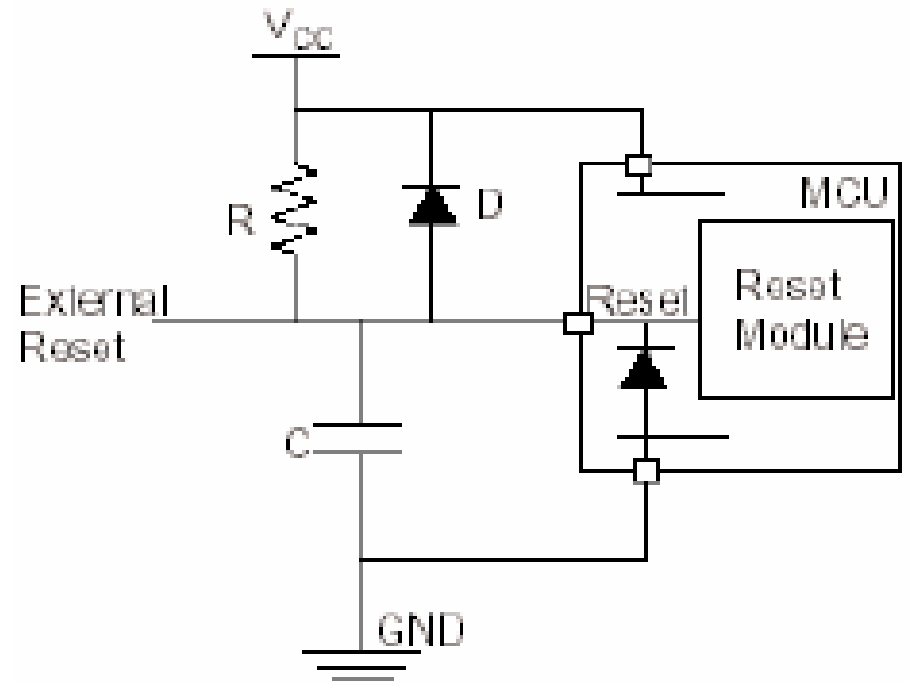
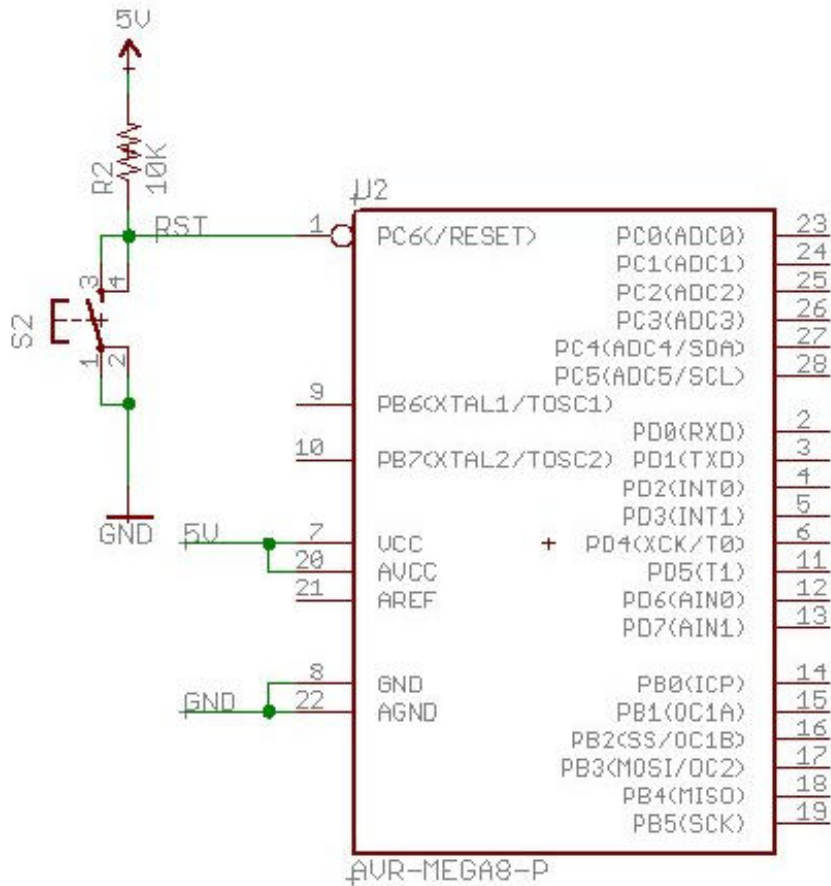
Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48A	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega48PA	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega88A	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega88PA	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega168A	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega168PA	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega328	32KBytes	1KBytes	2KBytes	2 instruction words/vector
ATmega328P	32KBytes	1KBytes	2KBytes	2 instruction words/vector

Μικροελεγκτής AVR

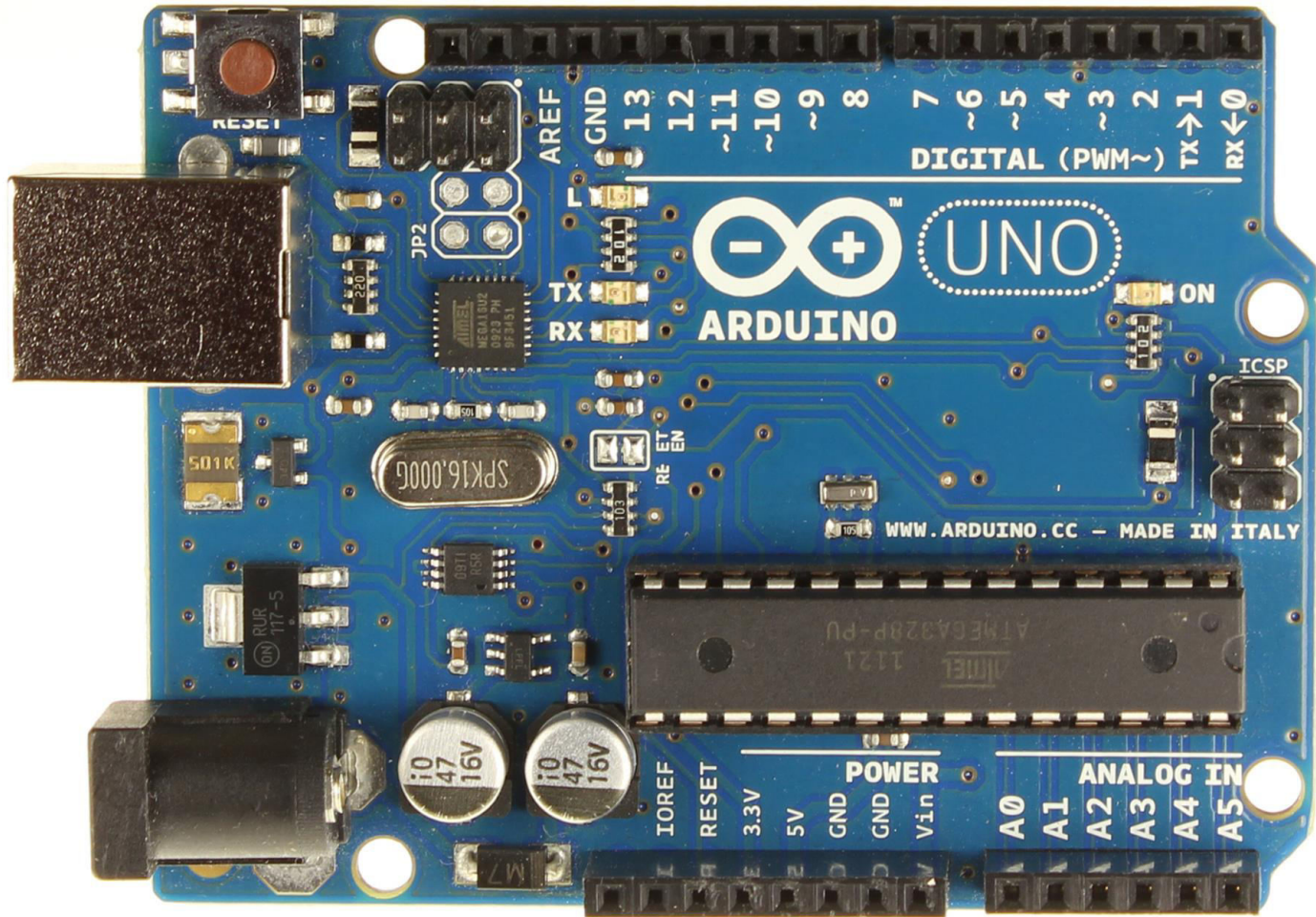
reset	1	28	analog 5
pin 0 rx	2	27	analog 4
pin 1 tx	3	26	analog 3
pin 2	4	25	analog 2
pin 3 pwm	5	24	analog 1
pin 4	6	23	analog 0
+5 volts	7	22	ground
ground	8	21	not connected
crystal	9	20	+5 volts
crystal	10	19	pin 13
pin 5 pwm	11	18	pin 12
pin 6 pwm	12	17	pin 11 pwm
pin 7	13	16	pin 10 pwm
pin 8	14	15	pin 9 pwm



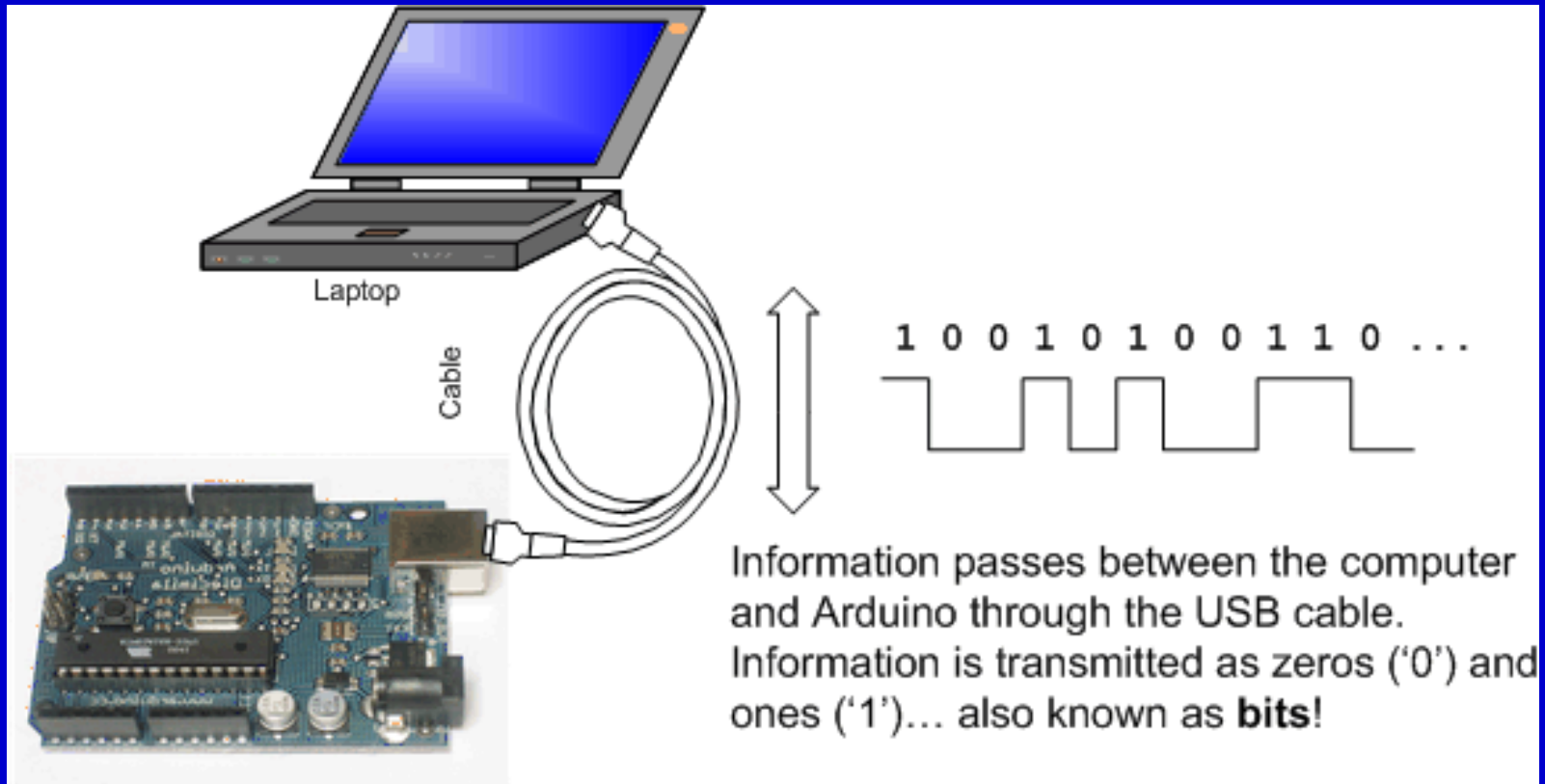
Κύκλωμα Reset των επεξεργαστών AVR



Arduino Uno



Σύνδεση του Arduino με Η/Υ



ARDUINO SOFTWARE

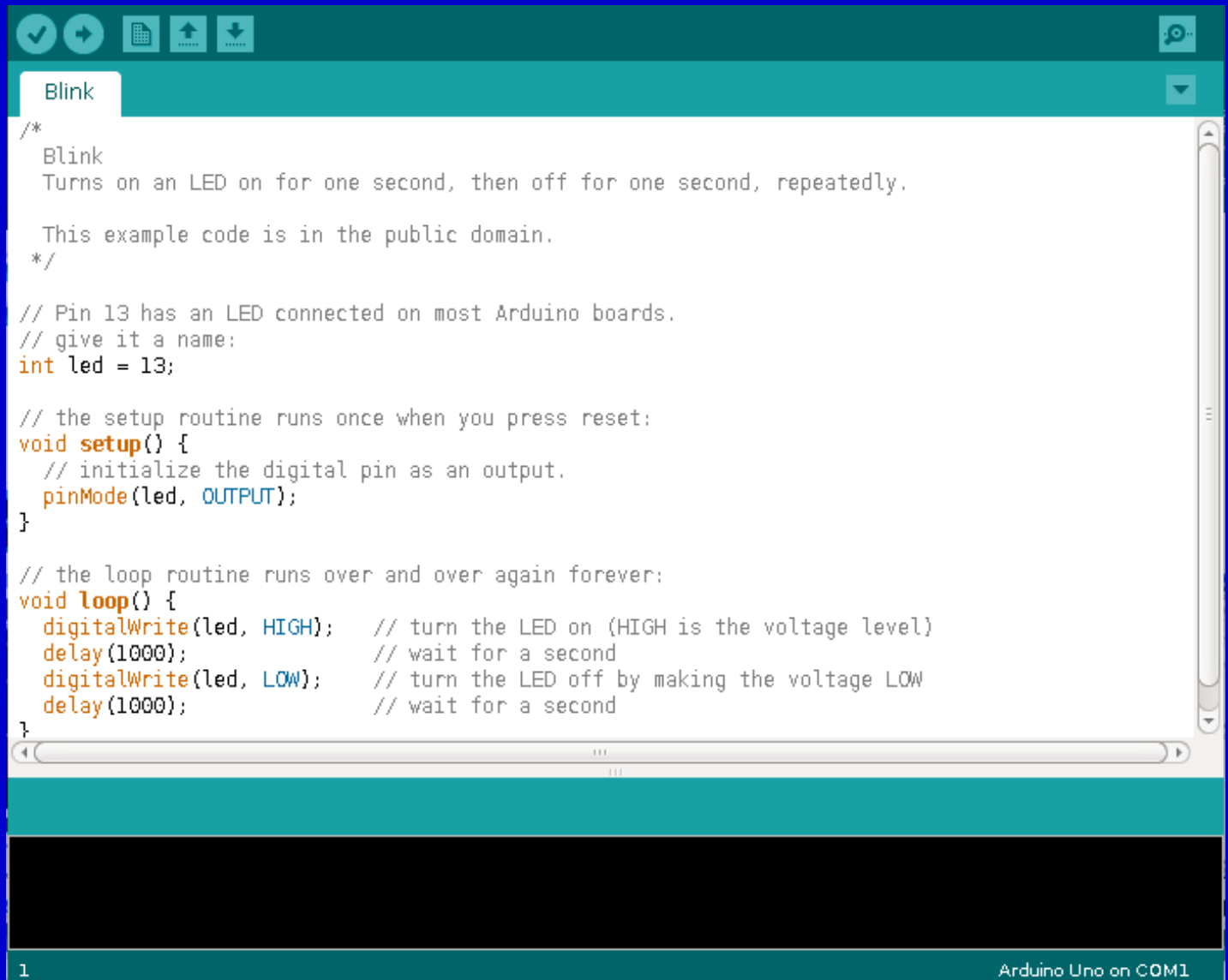
ARDUINO BOOTLOADER

(Must be uploaded into the Atmel controller to make it compatible with the arduino IDE)

ARDUINO IDE

(Installed in the computer and used to communicate with and program the arduino board)

Arduino IDE



The screenshot shows the Arduino IDE interface. At the top, there is a toolbar with icons for checking, running, saving, and uploading. Below the toolbar is a tab labeled "Blink". The main text area contains the following code:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

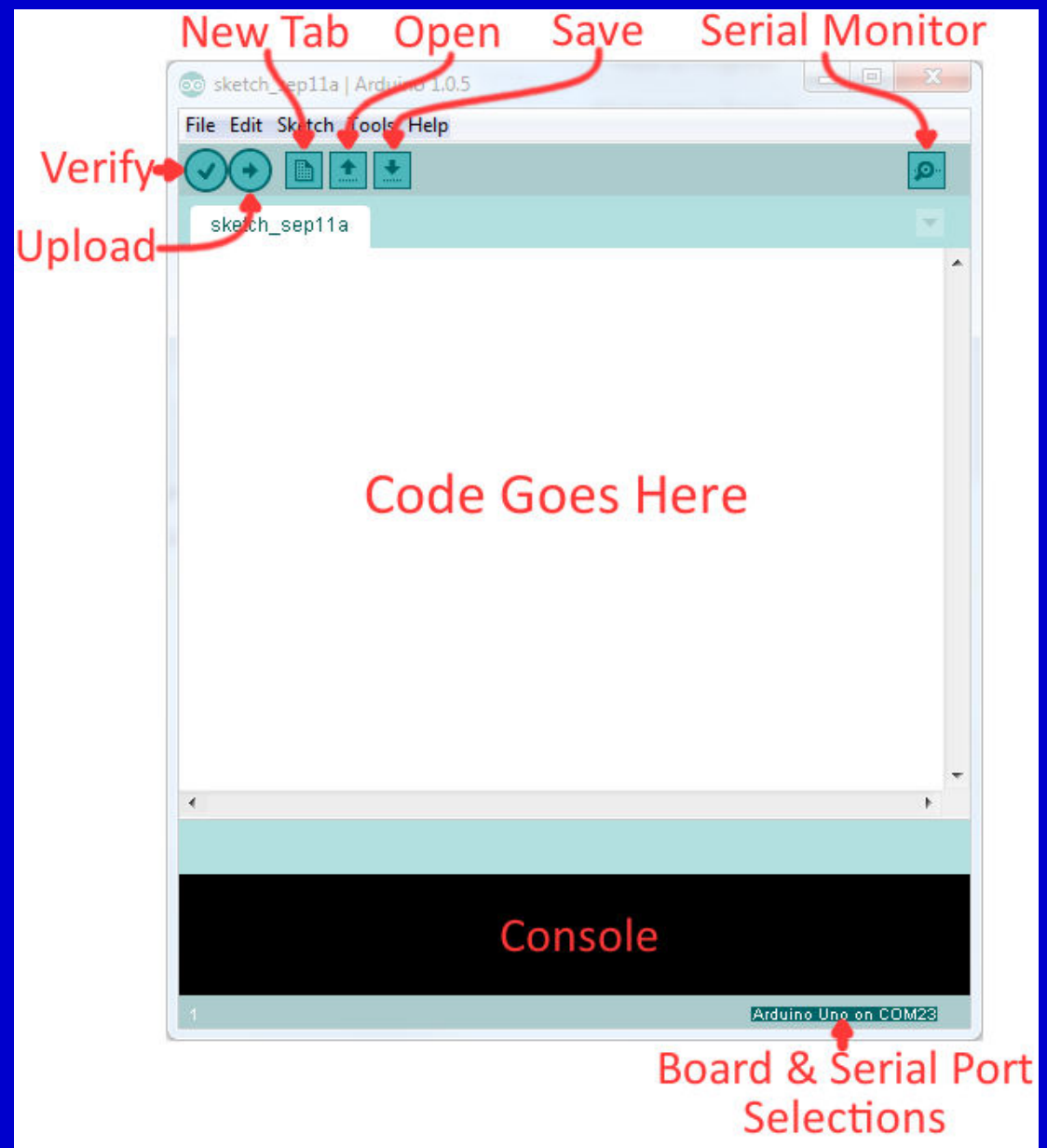
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

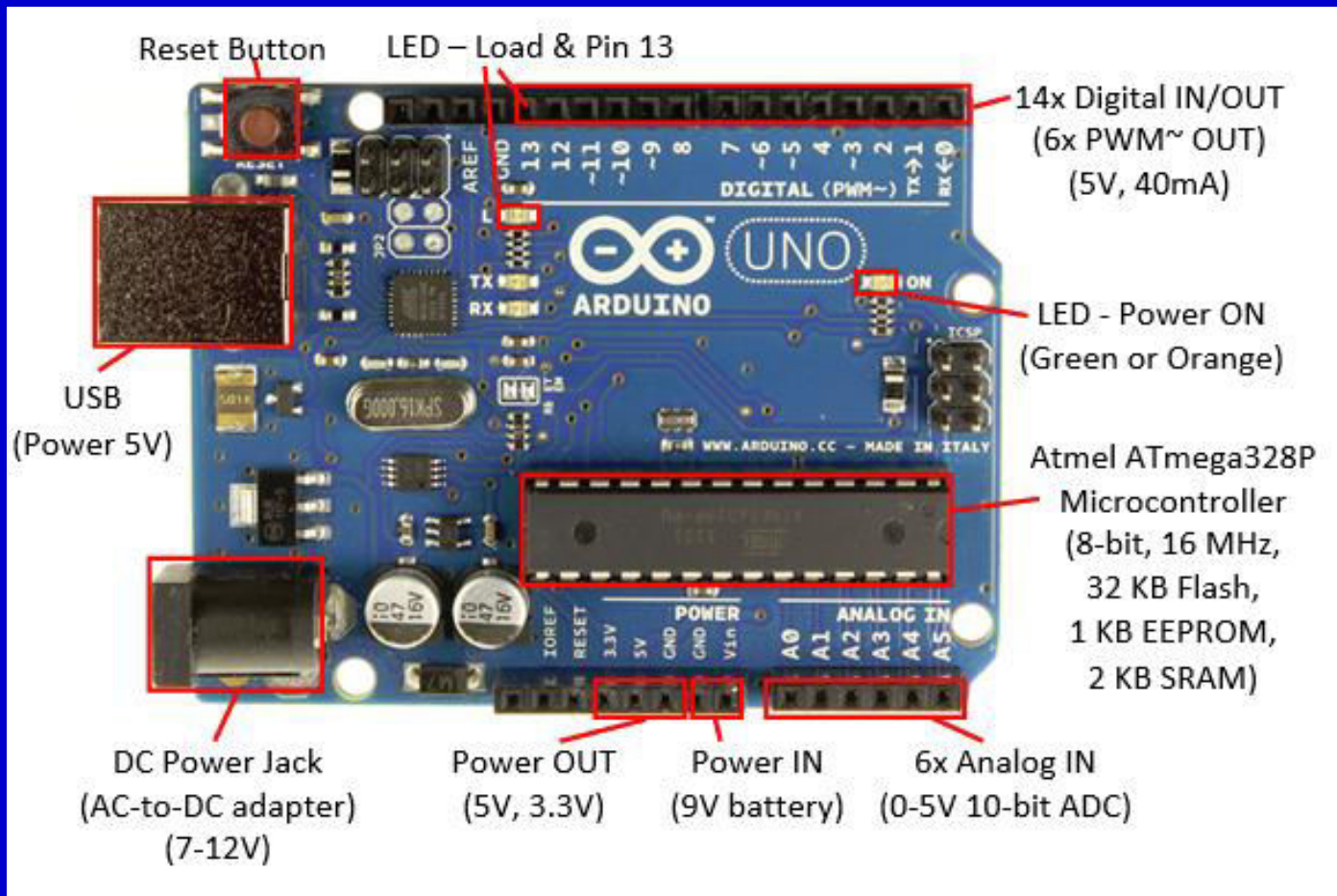
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

At the bottom of the IDE, there is a status bar showing "1" on the left and "Arduino Uno on COM1" on the right.

Arduino IDE



Σύστημα ανάπτυξης Arduino βασισμένο στον μικροελεγκτή ATMEL AVR



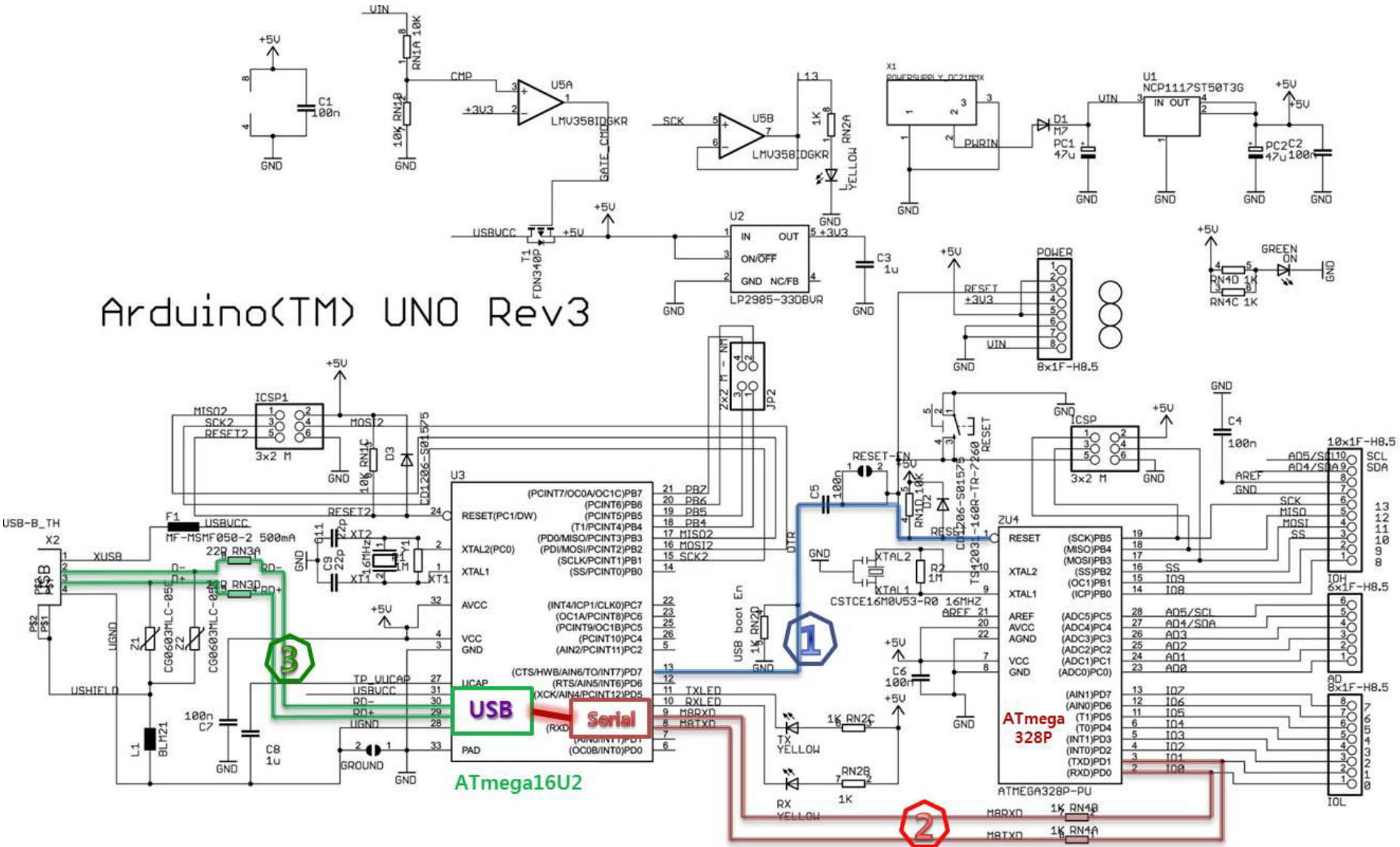
ΑΝΤΙΣΤΟΙΧΙΑ ΤΩΝ ΑΚΙΔΩΝ ΤΟΥ AVR ΜΕ ΤΙΣ ΠΟΡΤΕΣ ΤΟΥ Arduino

ATMega328P and Arduino Uno Pin Mapping

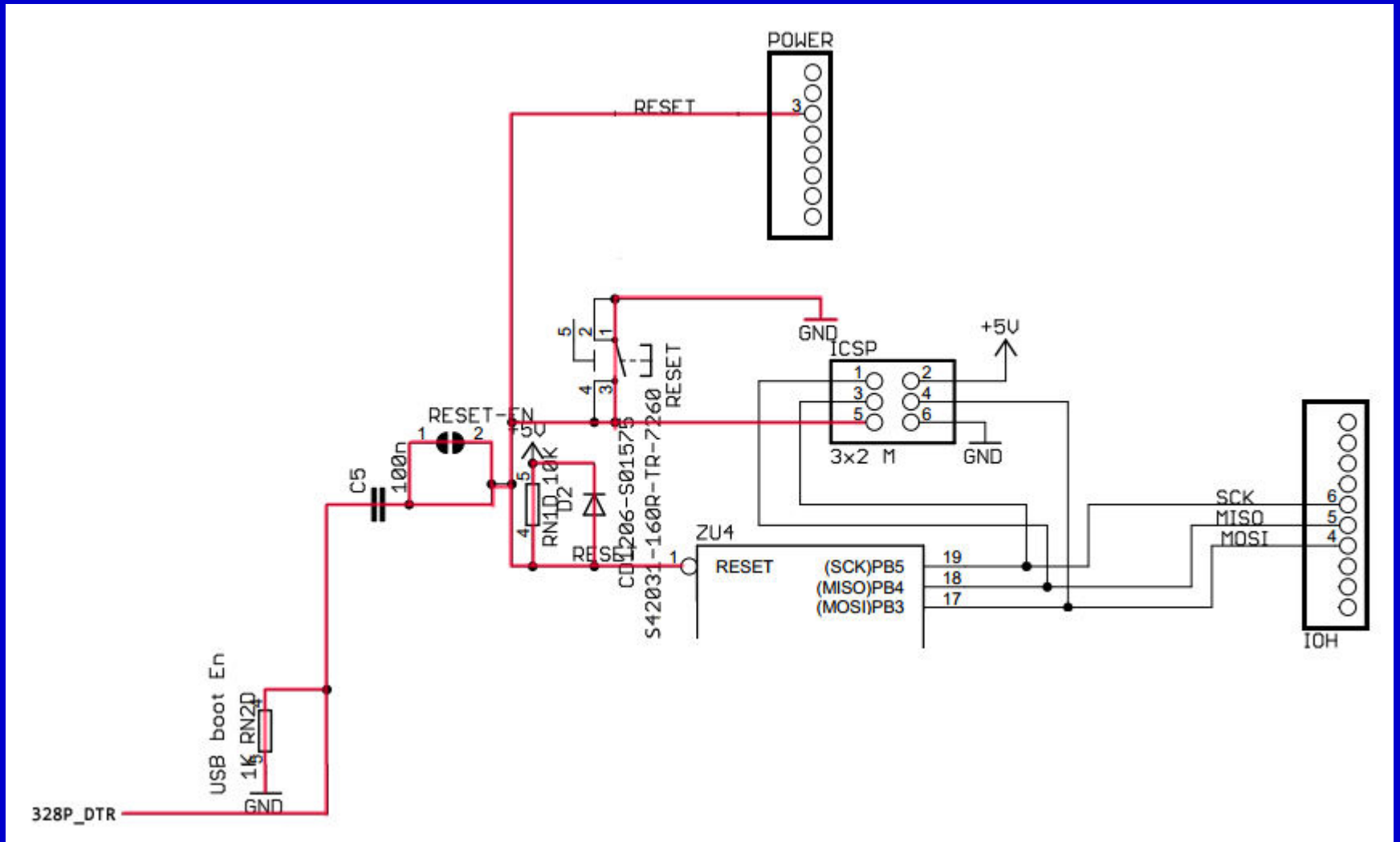
Arduino function					Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

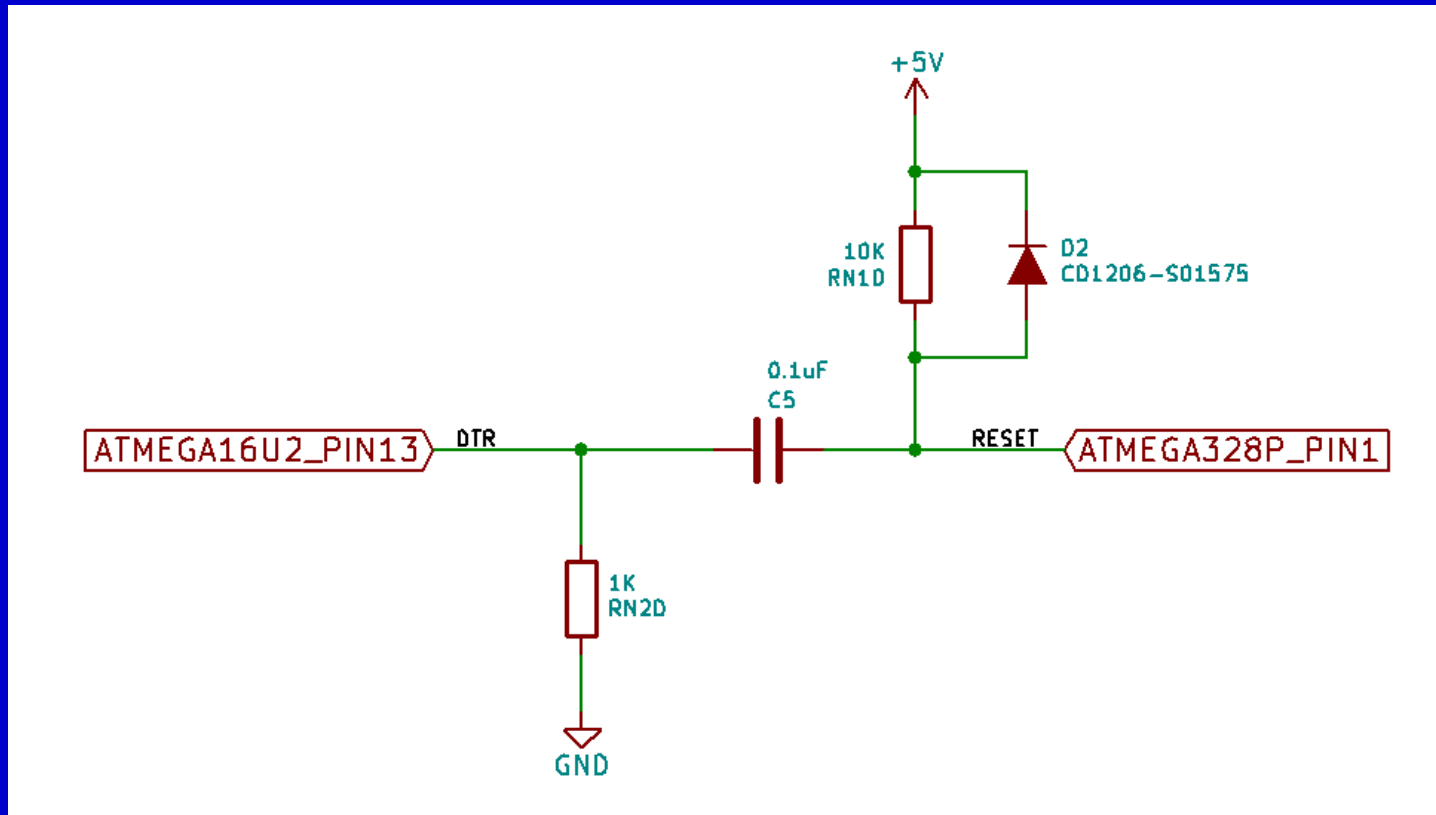
Arduino(TM) UNO Rev3



Κύκλωμα Reset του Arduino



Κύκλωμα Reset του Arduino



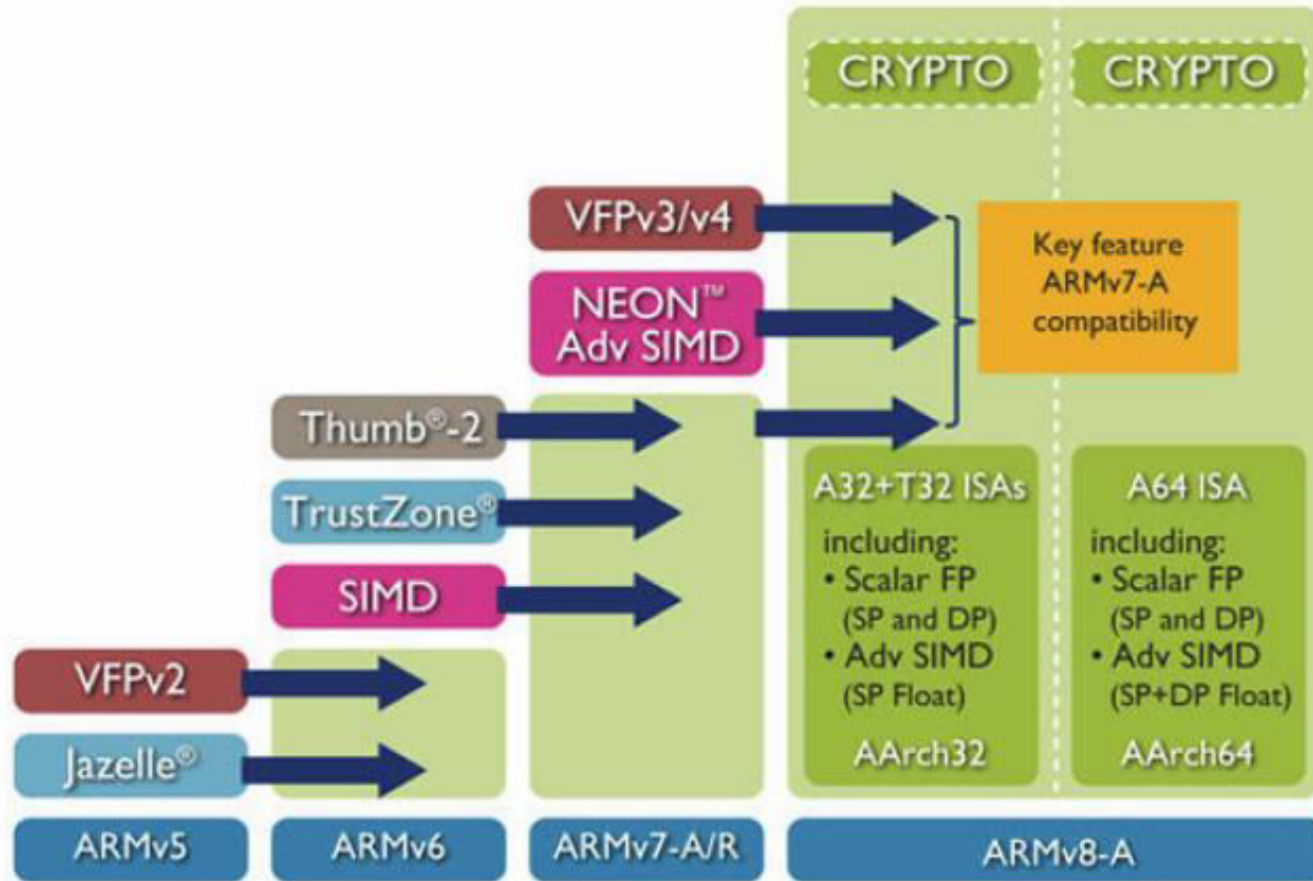
ARM

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

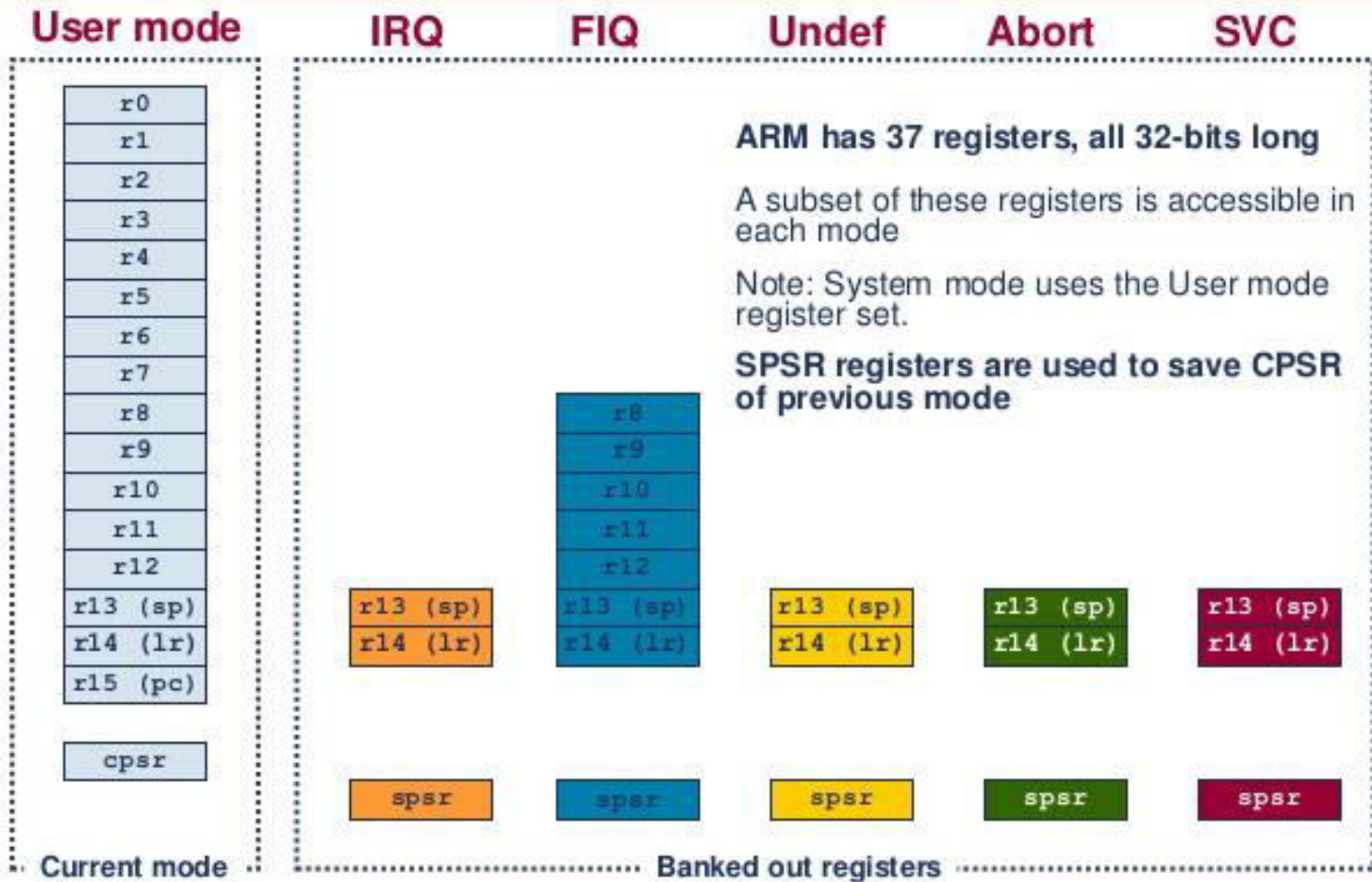
(From arm.com)



- 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

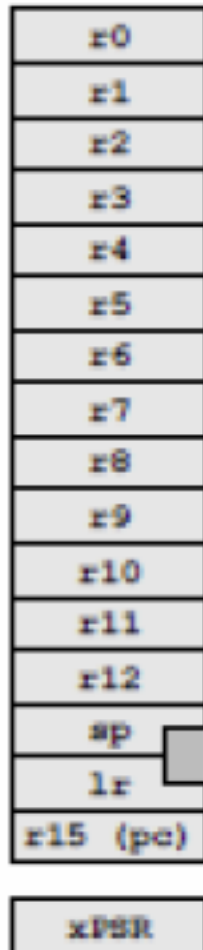
Architecture	Bit width	Cores designed by ARM Holdings	Cores designed by 3rd parties	Cortex profile
ARMv1	32/26	ARM1		
ARMv2	32/26	ARM2, ARM3	Amber	
ARMv3	32	ARM6, ARM7		
ARMv4	32	ARM8	StrongARM, FA526	
ARMv4T	32	ARM7TDMI, ARM9TDMI		
ARMv5	32	ARM7EJ, ARM9E, ARM10E	XScale, FA626TE, Feroceon, PJ1/Mohawk	
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 ^[22]	X-Gene, Denver, Apple A7 (Cyclone)	Application
ARMv8-R	32	No announcements yet		Real-time

The ARM Register Set



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

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`

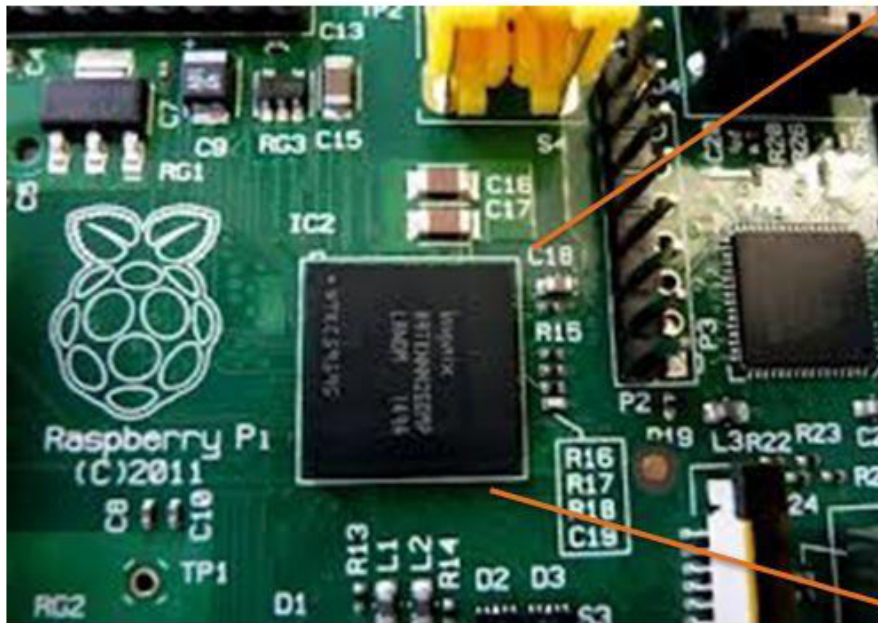
Raspberry Pi



Raspberry Pi – Hardware

Broadcom BCM2835

CPU core is an ARM11



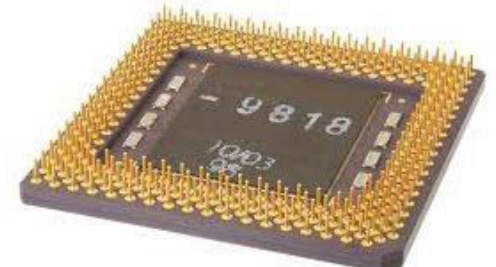
GPU

+

CPU

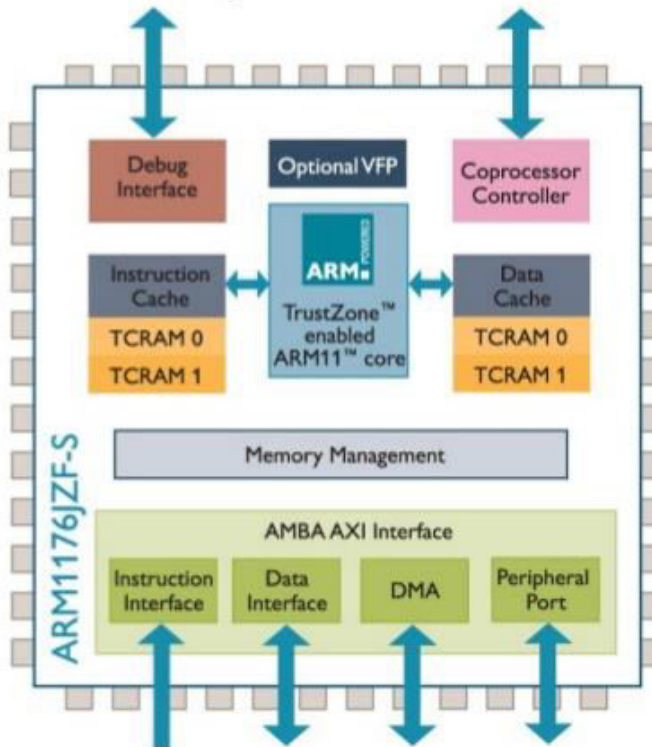
+

RAM

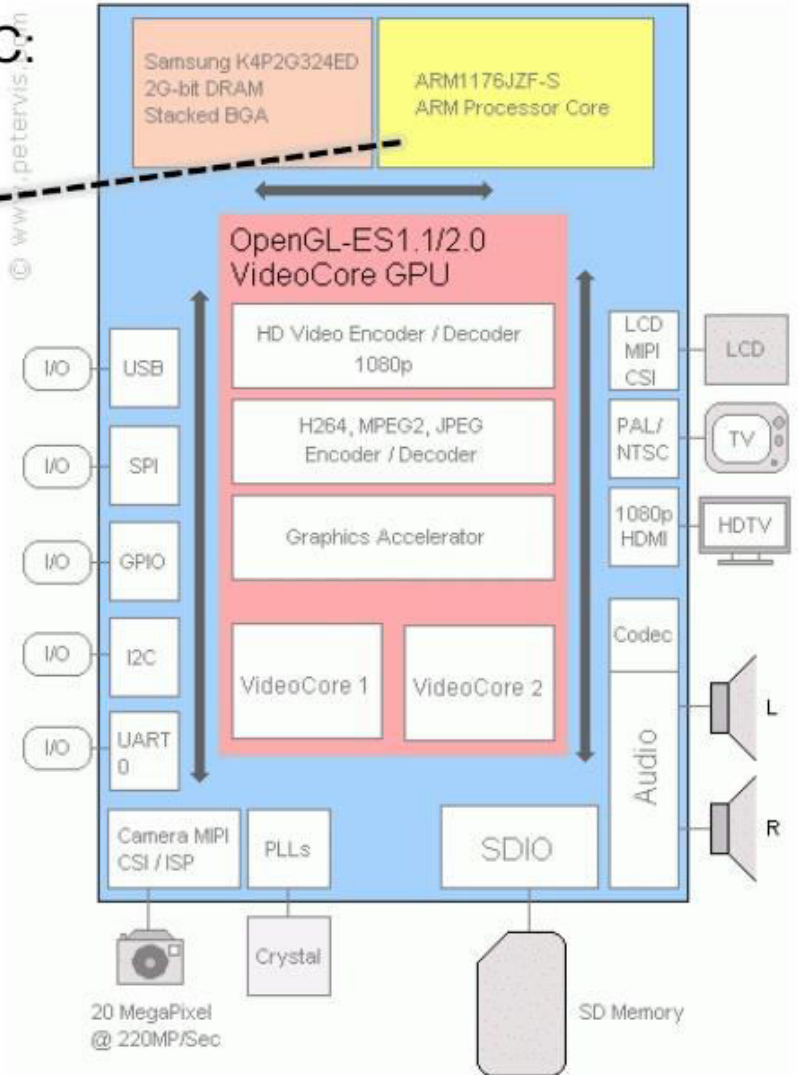


ARM based BCM2835 main chip of Raspberry Pi

Réseaux sur puce : Network On chip NoC:



ARM1176JZ



SOC: BCM2835 Block Diagram

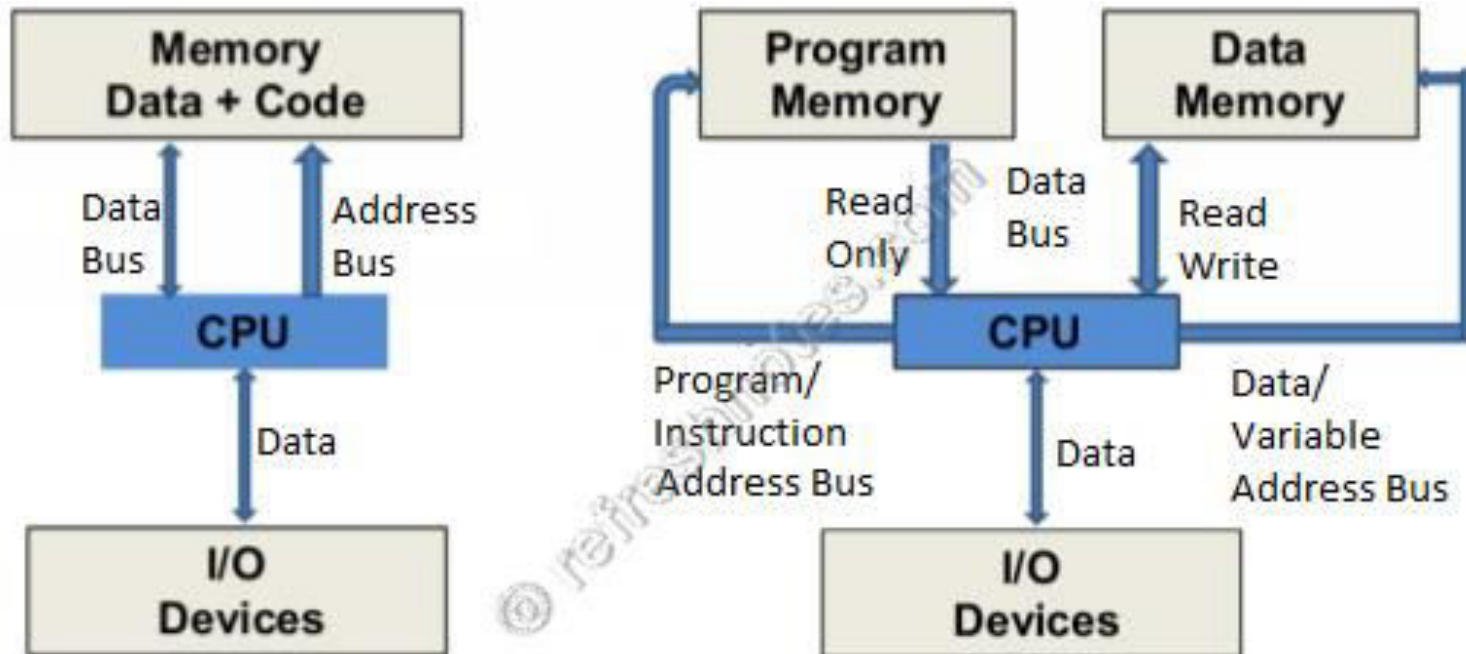
- 2.1.** Μία CPU έχει δίαυλο δεδομένων εύρους 8 bit και δίαυλο διευθύνσεων εύρους 16 bit. Ποιο θα είναι το μέγιστο μέγεθος (σε KByte ή MByte) της κύριας μνήμης που μπορεί να διαθέτει ένα υπολογιστικό σύστημα βασισμένο σε αυτή.
- 2.2.** Μία CPU έχει δίαυλο δεδομένων εύρους 8 bit και δίαυλο διευθύνσεων εύρους 24 bit. Ποιο θα είναι το μέγιστο μέγεθος (σε KByte ή MByte) της κύριας μνήμης που μπορεί να διαθέτει ένα υπολογιστικό σύστημα βασισμένο σε αυτήν.

2.3. Μία CPU έχει συχνότητα ωρολογίου $f=10$ MHz. Ποια είναι η διάρκεια του κύκλου του ωρολογίου της.

$$T=1/f=1/10\text{MHz}=1/10^7\text{Hz}=(1/10^7)\text{sec}=(100/10^9)\text{sec}=
=100\times 10^{-9}\text{ sec}=100\text{ ns}$$

2.4. Μία CPU έχει συχνότητα ωρολογίου 10 MHz και σε κάθε κύκλο εκτελείται μία εντολή γλώσσας μηχανής. Πόσο θα διαρκέσει η εκτέλεση ενός προγράμματος που απαιτεί την εκτέλεση 1000 εντολών γλώσσας μηχανής.

2.5. Από τις αρχιτεκτονικές CPU που περιγράφονται στην συνέχεια ποια είναι von Neumann και ποια Harvard.



2.6. Στο σύστημα Arduino που δίδεται στην συνέχεια εντοπίστε τον μικροελεγκτή, τον κρύσταλλο του ταλαντωτή και το ολοκληρωμένο που μετατρέπει τα σειριακά σήματα σε USB.



2.7. Στο σύστημα Arduino που δίδεται στην συνέχεια σημειώστε τις ψηφιακές εισόδους/εξόδους, τις αναλογικές εισόδους και τις σειριακές εισόδους/εξόδους.



2.8. Σε έναν επεξεργαστή AVR των 8 bit οι καταχωρητές R1, R2 περιέχουν τα δεδομένα 0x0A και 0x0B κτελείται η εντολή
ADD R1, R2

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

ADD

Adds two registers without the C Flag and places the result in the destination register Rd.

Operation: $Rd \leftarrow Rd + Rr$

Syntax: **ADD Rd, Rr**

Operands: $0 \leq d \leq 31$ $0 \leq r \leq 31$

Flags: H, S, V, N, Z, C

Words: 1

Cycles: 1

2.9. Σε έναν επεξεργαστή ARM των 32 bit οι καταχωρητές R1, R2, R3 περιέχουν τα δεδομένα 0x000000001, 0x00000000A και 0x00000000B, 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)
```

2.10. Στο σύστημα Raspberry που ακολουθεί σημειώστε το ολοκληρωμένο κύκλωμα που περιέχει τον επεξεργαστή ARM, την θέση της πόρτας GPIO, και την θύρα HDMI.

