

ATmega1280

EEPROM vs Flash

- **EEPROM:** permite ștergerea granulară de câte un octet *at a time*.
 - Scriere lentă, bună pentru modificări mici.
- **Flash:** pentru ștergere se citește un block, se pune într-un buffer, se schimbă octetul dorit.
 - Se șterge întregul block original și se înlocuiește cu cel din buffer.

Canale ADC

- **16 canale.**
- Rezoluție **10 biți** -> 2^{10} trepte, 1024 pași.
 - De obicei 0-5V, deci **4.9mV** per unitate.
- Un singur convertor intern, cu un mux care îi permite să asculte pe 16 pini.
 - **ADC0-ADC7** ->> PORTF
 - **ADC8-ADC15** ->> PORTK

DAC Dedicat

- **NU are.**
- Se poate simula o ieșire analogică cu un **PWM**.

Registri ATmega1280

- **R0-31** -- uz general 8 biți ->> direct conectați la ALU.
 - $0x0000$ - $0x001F$ în SRAM (mapare).
- Ultimii 6 registri (Pointeri):
 - **X** = R27(High) + R26(Low)
 - **Y** = R29(High) + R28(Low) -> *frame pointer*
 - **Z** = R31(High) + R30(Low) -> accesarea memoriei flash prin LPM sau ELPM
- **Registri IO:**
 - Mapati: $0x0020$ - $0x005F$
- **Registri IO extinși:**
 - Mapati: $0x0060$ - $0x01FF$

Calling Convention

- **Parametri:** R16-R23
- Restul prin **stack**.
- Return value
- 8 biți (uint8_t, char) -> r24
- 16 biți (uint16_t, int) -> r25:r24 (r25 = byte superior, r24 = byte inferior)
- 32 biți (uint32_t, long) -> r25:r24:r23:r22
- 64 biți -> r25 ... r18

☒ Tipuri speciale

- void -> nimic
- float (32-bit) -> r25:r24:r23:r22
- pointeri (16-bit pe AVR) -> r25:r24

Directiva Assembler: `RSEG CODE:CODE:NOROOT(2)`

- **RSEG** = Relocatable segment
- **CODE:** = Symbolic segment name
- **:CODE** = Tipul; dacă e code spune că trebuie pus în flash
- **:NOROOT** = spune că linkerul are voie să îl elimine dacă nu e folosit
- **(2)** = alinierea la puteri ale lui 2

Arhitectură Sistem

- **Magistrală Date:** 8 biți (Arhitectură RISC nativă pe 8 biți).
- **Magistrală Adrese:** 16 / 22 biți.
 - 16 biți pentru RAM.
 - Până la 22 biți pentru Flash (PC).
- **Flash (Program):** 128 KB. Organizată ca 64K x 16 biți (instrucțiunile sunt de 16/32 biți).
- **SRAM (Date):** 8 KB. Memorie volatilă pentru variabile și stivă.

- **EEPROM:** 4 KB. Memorie non-volatilă pentru parametri.

Mapare Memorie

Adresă	Tip	Descriere
0x0000 - 0x001F	Registre Lucru	Cele 32 de registre de uz general (R0 - R31).
0x0020 - 0x005F	I/O Registers	Porturi standard (64 de registre, accesibile prin IN/OUT).
0x0060 - 0x01FF	Extended I/O	Registre suplimentare (Timere, UART-uri noi).
0x0200 - 0x21FF	Internal SRAM	Zona dedicată pentru datele utilizatorului.

Registri Speciali

- **Program Counter (PC):** Are 17 biți pentru a putea adresa întreaga memorie Flash de 128 KB (adresele de salt sunt la nivel de word, nu de byte).
- **Stack Pointer (SP):** Este pe 16 biți, stocat în două registre de 8 biți (SPL și SPH).
- **Vectori Întreruperi:** Situați la începutul memoriei Flash (0x0000), fiecare vector ocupând 2 word-uri (4 bytes).

Configurație Pini

- **Total:** 100 pini.
- **Capsulă:** TQFP 100.
- **Organizare:** Port A..L.

Pini Alimentare și Referință

- **VCC (2 pini):** Alimentare digitală (+5V sau +3.3V).
- **GND (5 pini):** Masă (0V). Sunt mai mulți pentru a reduce zgomotul electric.
- **AVCC (1 pin):** Alimentare pentru convertorul ADC. Trebuie conectat la VCC (de preferat prin filtru LC).
- **AREF (1 pin):** Referință analogică externă pentru ADC.

Pini de Sistem și Control (5 pini)

- **RESET:** Pin activ pe "LOW" pentru restartarea microcontrolerului.
- **XTAL1 & XTAL2:** Pini pentru conectarea cristalului de cuarț (oscilator extern).
- **PEN:** Pin pentru activarea modului de programare serială (doar în anumite condiții de testare).

Organizare memorie 8086/8088

- **TPA (Transient Program Area) = 640Kb**
 - Rulează programe și MS DOS – 0 - 640K .
- **System Area = 384Kb**
 - 640K - 1Mb .
 - Conține: VRAM, BIOS, ROM.
 - **TPA + System Area = Memorie Convențională.**
- **Memorie Extinsă:** tot ce e peste 1Mb.

Harta Memorie TPA

- **Low addr** -> interrupt vectors.
- **High addr** -> MSDOS program.

Semnale Control

- **MWTC / MRDC = MemoryWrite Command / MemoryRead Command.**
- **IORC / IOWC (Input/Output Read/Write Command).**

Specificații 8086 / 80286

- **8086** – CPU speed 5MHz, Memorie 16Ko.
- **80286** – Memorie 16 Mb.

Registri 8086

- **AX, BX, CX, DX** (16bit) - Registri generali.
- **Pointeri și Index:**
 - **SP** = Stack Pointer
 - **BP** = Base Pointer
 - **SI** = Source Index
 - **DI** = Destination Index

Registri Segmentare

- **CS** = Code Segment
- **DS** = Data Segment
- **SS** = Stack Segment
- **ES** = Extra Segment

Calcul Adresă Fizică

- **A.F** = (Segment << 4 + offset) => adresă fizică 20biți.
- Fiecare segment are o dimensiune maximă de **64Kb** (registru de offset are 16biți).

ARM7TDMI

- **32-bit.**
- **Interfață ETM** = Embedded Trace Macrocell.
 - Real time tracing for debug.
- **ARM 32-bit RISC:**
 - 16-bit **Thumb** instruction set for increased code density.
 - Three stage pipeline.
 - 32-bit ALU.

ARM Cortex-M3

- Arhitectură **Harvard**.
- **DAP** = Debug Access Port = interfațare între debugger intern și magistrale interne.
- **NVIC** = Nested Vectored Interrupt Controller.
- **WIC** = Wakeup Interrupt Controller -> trezire din deep sleep.
 - Rămâne pornit chiar și în deep sleep.
- **DWT** = Data Watchpoint and Trace unit = Implements hardware breakpoints and provides instruction execution statistics.
- **FPB** = Flash patch and breakpoint unit = Implements 6 program breakpoints and 2 literal data fetch breakpoints.
- **TPA** = Trace Port Analyzer.

Debug & Trace

- **ETM** = Embedded Trace Macrocell.
- **ETB** = Embedded Trace Buffer = Round Buffer in RAM.

Facilități Hardware

- **Atomic bit manipulation with Bit Banding:**
 - Fiecare bit dintr-o zonă de memorie de 1Mb are o adresă a lui de 32biți (alias).
 - Există 2 astfel de regiuni: una pentru memorie și una pt periferice.
- **Performanță Aritmetică:**
 - 32bit multiply = **1 cycle**.
 - Hw divide = between **2 and 12 cycles**.

Memorie

- Memory map preconfigurat.
- Memory space adresabil -> up to **6Go**.
- **Memory Protection Unit (MPU):**
 - Optional component for separation of processing tasks and data protection.
 - Up to 8 regions of protection; each of which can be divided into 8 sub-regions.
 - Region sizes between 32 bytes to the entire 4 gigabytes of addressable memory.

Sleep Modes

- **Sleep Now:** după instr curentă procesorul intră în sleep.
- **Sleep on exit:** sleep după o întrerupere.

Adresare și Memorie

ADRESA LOGICĂ = SELECTOR + OFFSET

- Selectorul este un index în tabela de descriptori.
- **Tabel descriptori (GDT sau LDT)**: Local Descriptor Table sau Global Descriptor Table.
- Se citește `BASE + Offset` din descriptor => Rezultă **adresa liniară**.

Memorie Virtuală:

- `virtual addr -> PAGE` (este convertită) și **LINE** (rămâne la fel).

Microcontrolere (8051)

- 8051 verifică întreruperile doar la final de **S5P1**.
- *Notă: VEZI OVERFLOW FLAGS DETECTION!!!*

Arhitectura ARM (v5 & v8)

ARMv5

- Introduce excepții.
- **7 moduri execuție**: 1 user mode și 6 de privilegiu.
- **IRQ** = Interrupt Request.
- **FIQ** = Fast Interrupt Request (prioritate maximă).

ARMv8

- **NMI exception** = Non Maskable Interrupt.

Late Arriving (Gestionare Întreruperi)

- Apare **IRQ A** și **IRQ B**.
- IRQ A apare prima -> începe să se salveze contextul pe *stack frame*.
- Apare **IRQ B**, care este mai prioritară.
- Se termină procesul de salvare pe stack, dar se intră direct în **IRQ B**.

Stack (Stivă)

- **Utilizare implicită**: apeluri `PUSH / POP`.
- **Descrescător**: stiva crește spre adrese mici (în jos).
- **Plin / Gol (Full/Empty)**:
 - **Full**: SP (Stack Pointer) arată către ultimul element populat.
 - **Empty**: SP arată către prima locație liberă.

ARM: Registre și Calling Convention

- **r0 – r3**: parametri + valori de retur (*caller-saved*).
- **r4 – r11**: registre generale (*callee-saved*).
- **r13 (sp)**: Stack Pointer.
- **r14 (lr)**: Link Register (adresa de retur).
- **r15 (pc)**: Program Counter.

Returnarea valorilor:

- **32 biți**: `r0`.
- **64 biți**: `r1` (high) - `r0` (low).

ARMv5 Specifics:

- **Thumb Mode**: `PUSH ≈ STMFD sp!` și `POP ≈ LDMFD sp!`.
- Modul de utilizare al stivei: **Full Descending**.

80286 mem

- **Magistrală de Date (Data Bus):** 16bit
- **Magistrală de Adrese (Address Bus):** 24bit
- **Memorie Fizică Maximă:** 16Mo - 2²⁴
- **Memorie Virtuală (per task):** 1Go
- **Dimensiune Maximă Segment:** 64 Ko (OFFSET pe 16 biti)
- **Spațiu de Adresare I/O:** 64Ko

Mapare Memorie

Adrese	Descriere
00000h – 9FFFFh	Memorie Convențională (RAM utilizabil pentru programe) - 640 KB
A0000h – BFFFFh	Memorie Video (VRAM) - 128 KB
C0000h – DFFFFh	ROM BIOS pentru periferice / adaptoare - 128 KB
E0000h – FFFFFh	BIOS-ul Sistemului (Motherboard BIOS) - 128 KB
100000h – FFFFFFFh	Memorie Extinsă (accesibilă doar în Mod Protejat sau via drivere XMS) - Până la 15 MB restanți.

8086 mem

- **Magistrală de Date (Data Bus):** 16bit
- **Magistrală de Adrese (Address Bus):** 20bit
- **Memorie Fizică Maximă:** 1Mo - 2²⁰
- **Memorie Virtuală (per task):** Nu are
- **Dimensiune Maximă Segment:** 64Ko - 2¹⁶ (OFFSET pe 16 biti)
- **Spațiu de Adresare I/O:** 64Ko

Tabelă Memorie

Interval Adrese (Hex)	Dimensiune	Utilizare Tipică
00000h – 003FFh	1 KB	IVT (Interrupt Vector Table). Tabela vectorilor de întrerupere (esențială pentru funcționarea CPU).
00400h – 004FFh	256 Bytes	BIOS Data Area. Variabile de sistem (ex: poziția cursorului, starea tastaturii).
00500h – 9FFFFh	~640 KB	Memorie Convențională. Spațiul liber pentru sistemul de operare (DOS) și programele utilizatorului.
A0000h – BFFFFh	128 KB	Memorie Video (VRAM). Rezervată pentru adaptoarele grafice (CGA, EGA, VGA text).
C0000h – EFFFFh	192 KB	ROM Extensions. BIOS-uri pentru plăci de extensie, controllere de hard disk, rețea etc.
F0000h – FFFFFh	64 KB	System BIOS. Codul de pornire al plăcii de bază. La pornire, procesorul sare la adresa FFFF0h.

8051 mem

- **Magistrală de Date:** 8bit
- **Magistrală de Adrese:** 16bit
- **Memorie Program (ROM/Flash):** 64Ko (4 interni, restul externi)
- **Memorie Date Externă (XRAM):** 64Ko
- **Memorie Date Internă (IRAM):** 128o (00h - 7Fh)
- **Regiștri Speciali (SFR):** 128o

CODE

Adresă	Dimensiune	Descriere
0000h – 0FFFh	4 KB	ROM Intern.

Adresă	Dimensiune	Descriere
100h – FFFFh	60 KB	BOM Extern.

DATA

Adresă	Tip	Descriere
00h – 1Fh	Bancuri de Regiștri	32 bytes împărțiți în 4 bancuri (Bank 0-3) a câte 8 regiștri (R0-R7). Doar un banc este activ la un moment dat.
20h – 2Fh	Zonă Adresabilă pe Biți	16 bytes (128 biți total). Fiecare bit de aici poate fi modificat individual (instrucțiuni SETB, CLR).
30h – 7Fh	Scratchpad RAM	80 bytes. Zonă de memorie generală pentru variabilele utilizatorului și stivă (Stack).
80h – FFFh	SFR (Special Function Registers)	Aici nu este RAM fizic general, ci sunt porturile și perifericele (P0-P3, TMOD, SBUF, ACC, B, SP). Se accesează doar prin adresare directă.

Detecție Overflow și Flag-uri

Tip Aritmetică	Flag Principal	Când se activează (Set = 1)	Semnificație
Fără Semn (Unsigned)	CF (Carry)	Transport din MSB (la adunare) sau Împrumut (la scădere).	Rezultatul depășește valoarea maximă (ex: > 255 pe 8 biți).
Cu Semn (Signed)	OF (Overflow)	Bitul de semn s-a schimbat neașteptat.	Rezultatul a depășit intervalul (ex: > +127 sau < -128 pe 8 biți).
Oricare	ZF (Zero)	Rezultatul este exact 0.	Operanzii au fost egali (la scădere/comparare).
Cu Semn	SF (Sign)	Rezultatul are bitul MSB = 1.	Rezultatul este considerat negativ.

Reguli Logice pentru Adunare și Scădere

Operație	Regula pentru OF = 1 (Signed Overflow)	Regula pentru CF = 1 (Unsigned Carry)
A + B	Pos + Pos = Neg OR Neg + Neg = Pos	Apare transport (Carry-out) din bitul cel mai semnificativ.
A - B	Pos - Neg = Neg OR Neg - Pos = Pos	Scazi un număr mai mare din unul mai mic (Împrumut).

2. Instrucțiuni de Salt Condiționat (Jumps)

Aceste instrucțiuni sunt folosite imediat după `CMP` sau o operație aritmetică.

Aritmetică FĂRĂ SEMN (Unsigned)

Mnemonic	Condiție	Flag-uri	Descriere
JB / JNAE	Below / Not Above or Equal	CF = 1	Mai mic
JAE / JNB	Above or Equal / Not Below	CF = 0	Mai mare sau egal
JA / JNBE	Above / Not Below or Equal	CF = 0 AND ZF = 0	Mai mare

Mnemonic	Condiție	Flag-uri	Descriere
JL / JNGE	Below or Equal / Not Above	ZF = 1	Mai mic sau egal

Aritmetică CU SEMN (Signed)

Mnemonic	Condiție	Flag-uri	Descriere
JL / JNGE	Less / Not Greater or Equal	SF ≠ OF	Mai mic
JGE / JNL	Greater or Equal / Not Less	SF = OF	Mai mare sau egal
JG / JNLE	Greater / Not Less or Equal	ZF = 0 AND SF = OF	Mai mare
JLE / JNG	Less or Equal / Not Greater	ZF = 1 OR SF ≠ OF	Mai mic sau egal

3. Detectarea prin Logică de Bit (Hardware level)

Dacă analizezi transporturile la nivel de bit în ALU:

- **CF** = Transportul ieșit din ultimul bit (MSB).
- **OF** = (Transportul intrat în ultimul bit) **XOR** (Transportul ieșit din ultimul bit).
 - *Formula:* $OF = C_{in_MSB} \text{ XOR } C_{out_MSB}$

4. Exemplu Rapid (8 biți)

Operație: $127 + 1$

- **Binar:** $01111111 + 00000001 = 10000000$
- **Interpretare Fără Semn:** $127 + 1 = 128$ (Corect, încapă în 255) **CF = 0**
- **Interpretare Cu Semn:** $+127 + 1 = -128$ (Eroare! Am adunat două pozitive și a ieșit negativ) **OF = 1**