

1.1 PREZENTAREA GENERALA A MICROCONTROLLERE-LOR PE 8 BITI DIN FAMILIA ATMEL – AVR

Microcontrolerele din familia AVR prezinta o organizare de tipul RISC executand o instructiune / ciclu masina. Prezinta unor blocuri interne ca : oscilator intern , timere , unitate UART , SPI , rezistoare pull-up , PWM (pulse width modulation) , ADC , comparatoare, determina utilizarea acestor microcontrolere intr-o gama foarte larga de aplicatii.

Instructiunile acestei familii de microcontrolere au fost proiectate pentru a reduce dimensiunea unui program scris in limbaj C sau in limbaj de asamblare.

Posibilitatea programarii memoriei FLASH si a memoriei EEPROM, determina ca aceste microcontrolere sa aiba o larga utilizare datorata costului mic de dezvoltare a unei aplicatii (timpul de proiectare scurt).

O alta calitate remarcabila a acestor microcontrolere este consumul foarte mic de energie. Domeniul tensiunilor de alimentare este cuprins intre 1.8 si 5V.Prezinta 6 moduri diferite de stand-by ceea ce ne asigura ca aceste microcontrolere nu vor consuma energie decat atunci cand este nevoie.

Controlul software al frecventei garanteaza o viteza maxima de executie atunci cand este nevoie , iar in restul timpului microcontrolerul poate trece in stand-by unde consumul de energie este minim.

Utilizarea acestor microcontrolere poate reduce semnificativ timpul de dezvoltare a unei aplicatii datorita prezentei pe acestea a unui bloc de depanare in timp real , circuitul aflandu-se chiar pe placa ce reprezinta aplicatia. Se pot face in timp real operatii de "watch" asupra unor registri , operatii de rulare pas cu pas , operatii de oprire in breakpoint.

Sunt produse microcontrolere de uz general precum si microcontrolere cu functii specializate.

Printre cele mai utilizate microcontrolere de uz general putem specifica : ATMEGA 8 , ATMEGA 16 , ATMEGA 128, ATMEGA 162. etc.

Microcontrolerele cu functii specializate se impart in mai multe catagorii :

- Lighting AVR : contin un procesor de semnal si este special conceput pentru controlul motoarelor de curent continuu.
- LCD AVR : contin un controler pentr adresarea dispozitivelor LCD
- CAN AVR : contin implementata hardware o interfata CAN.

CAP. 2 ATmega 16 ARHITECTURA

2.1 INTRODUCERE

ATmega 16 este un microcontroler CMOS de 8 – biti de mica putere bazat pe arhitectura RISC AVR imbunatatita.

Dispune de un set de 131 instructiuni si 32 de registre de uz general. Cele 32 de registre sunt direct adresabile de Unitatea Logica Aritmetica (ALU), permitand accesarea a doua registre independente intr-o singura instructiune. Se obtine astfel o eficienta sporita in executie (de pana la zece ori mai rapide decat microcontrolerele convetionale CISC).

ATmega16 este un microcontroler RISC pe 8 biți realizat de firma Atmel. Caracteristicile principale ale acestuia sunt:

- 16KB de memorie Flash reinscriptibilă pentru stocarea programelor
- 1KB de memorie RAM
- 512B de memorie EEPROM
- două numărătoare/temporizatoare de 8 biți
- un numărător/temporizator de 16 biți
- conține un convertor analog – digital de 10 biti, cu intrări multiple
- conține un comparator analogic
- conține un modul USART pentru comunicație serială (port serial)
- dispune de un cronometru cu oscilator intern
- oferă 32 de linii I/O organizate în patru porturi (PA, PB, PC, PD).

Structura internă generală a controlerului este prezentată în Figura 1. Se poate observa că există o magistrală generală de date la care sunt conectate mai multe module:

- unitatea aritmetică și logică (ALU)
- registrele generale
- memoria RAM și memoria EEPROM
- liniile de intrare (porturile – I/O Lines) și celelalte blocuri de intrare/ieșire. Aceste ultime module sunt controlate de un set special de registre, fiecare modul având asociat un număr de registre specifice.

Memoria Flash de program împreună cu întreg blocul de extragere a instrucțiunilor, decodare și execuție comunică printr-o magistrală proprie, separată de magistrala de date menționată mai sus. Acest tip de organizare este conform principiilor unei arhitecturi Harvard și permite controlerului să execute instrucțiunile foarte rapid.

Modul Power-down salvează conținutul registrelor, dar blochează Oscilatorul, dezactivând toate celelalte funcții ale chip-ului până la următoarea Întrerupere Externă sau Reset hardware. În modul Power-save, timer-ul asincron continuă să meargă, permițând user-ului să mențină o bază de timp în timp ce restul dispozitivului este oprit.

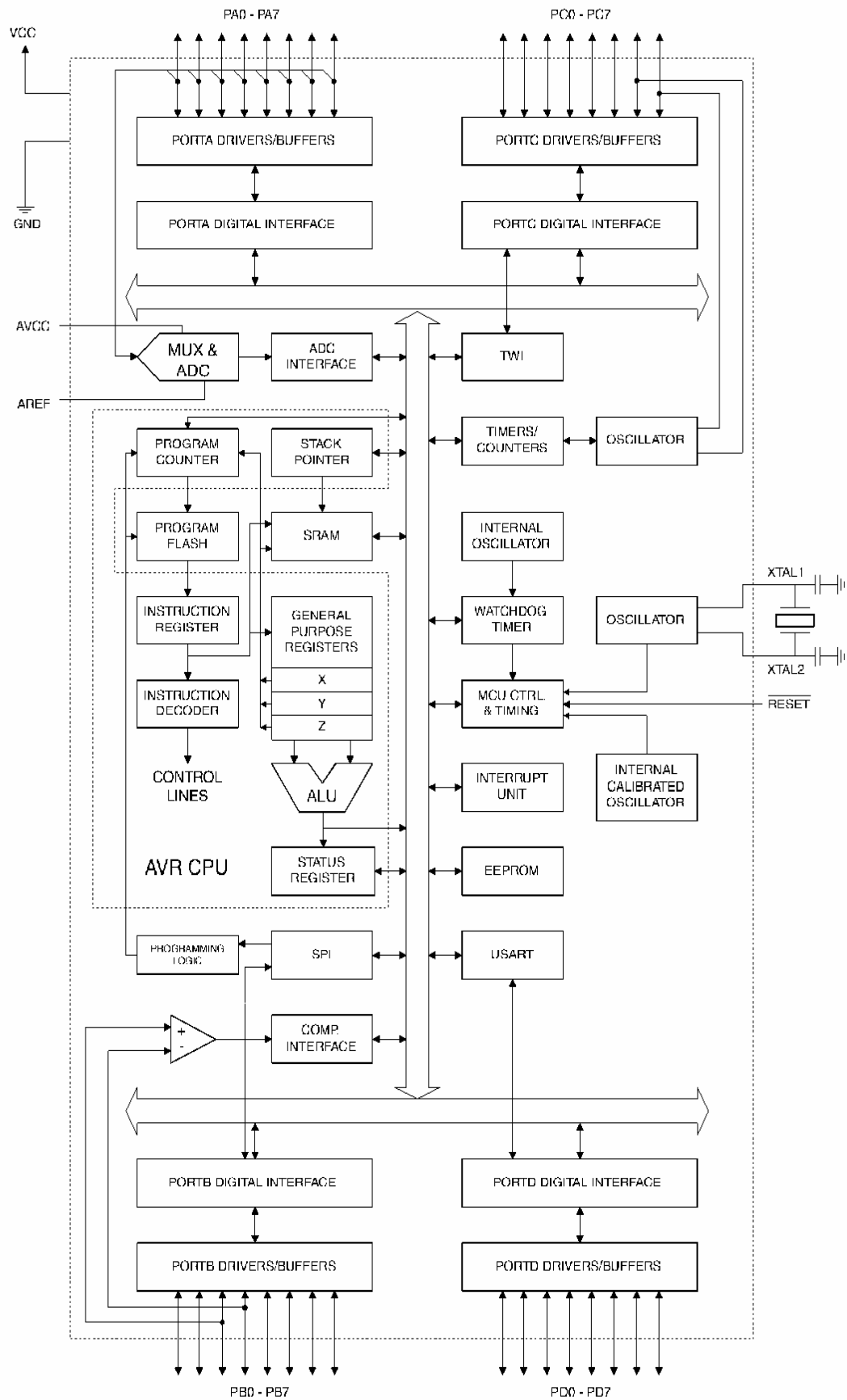
În modul Standby, Oscilatorul funcționează în timp ce restul dispozitivului este oprit. Acest lucru permite un start foarte rapid combinat cu un consum redus de energie. În modul standby extins (Extended Standby Mode), atât Oscilatorul principal cât și timer-ul asincron continuă să funcționeze.

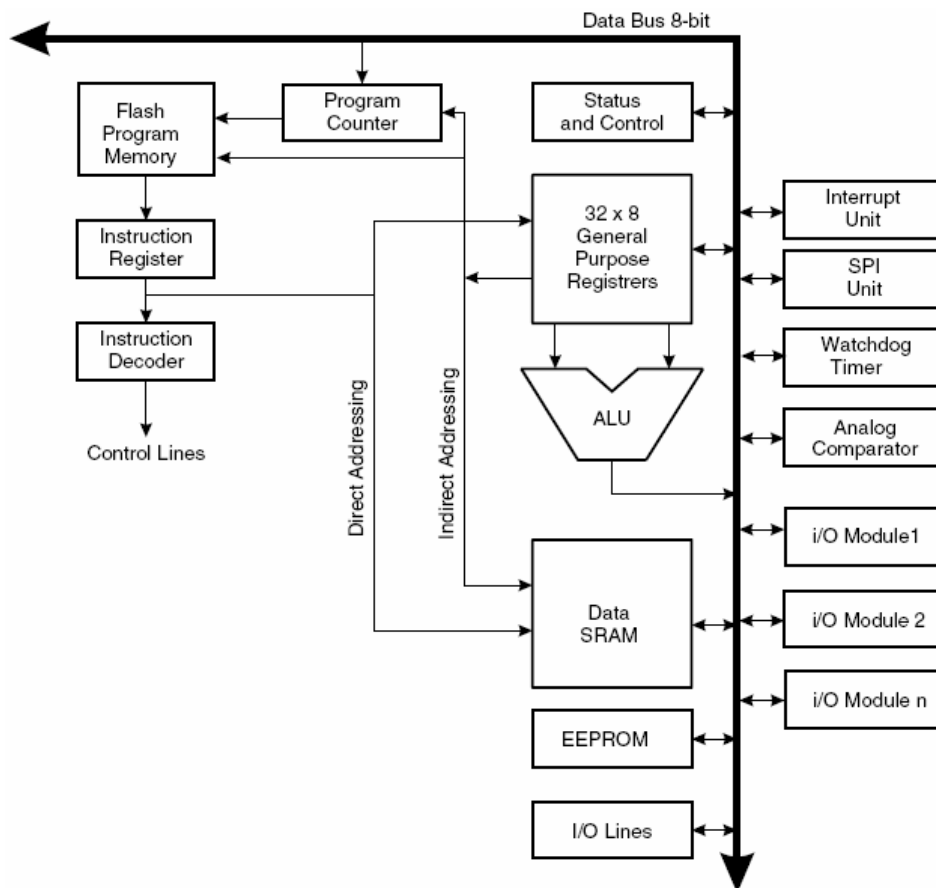
Memoria flash (On-chip) permite să fie reprogramată printr-o interfață serială SPI, de către un programator de memorie nonvolatilă convențională, sau de către un program de boot On-chip ce rulează pe baza AVR. Programul de boot poate folosi orice interfață să încarce programul aplicație în memoria Flash de aplicație.

Combinând un CPU RISC de 8 biți cu un Flash In-system auto – programabil pe un chip monolitic, ATmega 16 este un microcontroler puternic ce oferă o soluție extrem de flexibilă și cu un cost redus în comparație cu multe altele de pe piață.

ATmega 16 AVR este susținut de o serie completă de instrumente de program și de dezvoltare a sistemului, care include: compilatoare C, macroasamblare, programe debug/simulare etc.

Diagrama bloc :





ATmega16 conține 32 de registre de uz general și 64 de registre speciale pentru modulele I/O. Aceste registre sunt mapate la adrese din memoria RAM cuprinse între 0000h și 005Fh.

Descrierea pinilor

VCC – Sursa de curent

GND – Masa

Port A (PA7 .. PA0)

Port-ul A serveste drept port de intrari analogice pentru Convertorul A/D.

Port-ul A serveste de asemenea si ca un port bidirectional I/O de 8 biti, in cazul in care Convertorul A/D nu este folosit. Pinii de port pot fi conectati optional la VCC prin rezistori interni, (selectati pentru fiecare bit). Buffer-ele de iesire ale Portului A au caracteristici de amplificare .

Port B (PB7.. PB0)

Portul B este un port I/O de 8 biti bidirectional cu rezistori interni (optional).

Buffer-ele de iesire ale Port-ului B au caracteristici de amplificare.

Port-ul B indeplineste de asemenea functii speciale ale microcontrolerului ATmega 16

Port C (PC7...PC0)

Portul C este un port I/O de 8 biti bidirectional cu rezistori interni (optional).

Buffer-ele de iesire ale Port-ului C au caracteristici de amplificare.

Daca interfata JTAG (de depanare) este activata , rezistorii pinilor PC5(TDI), PC3(TMS) si PC2(TCK) vor fi activati, chiar daca are loc o resetare.

Port-ul C indeplineste de asemenea functii ale interfetei JTAG si alte functii speciale ale ATmega 16.

Port D (PD7...PD0)

Portul D este un port I/O de 8 biti bidirectional cu rezistori interni conectati optional la VCC (selectati pentru fiecare bit). Buffer-ele de output ale Port-ului D au caracteristici de amplificare. Port-ul D indeplineste de asemenea functii speciale ale ATmega 16.

Reset

Un nivel scazut la acest pin mai mare ca durata decat o valoare prestabilita, va genera o initializare.

XTAL 1: Intrare pentru amplificatorul inversor al Oscilatorului;

XTAL 2: Iesire pentru amplificatorul inversor al Oscilatorului.

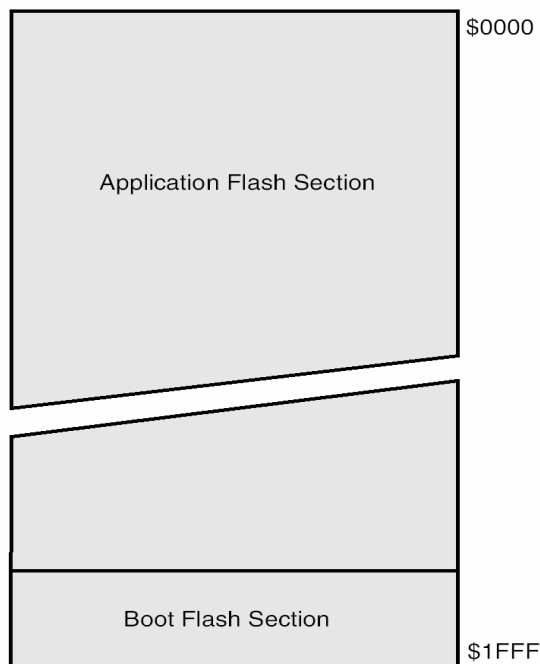
AVCC: AVCC este pin de alimentare pentru Port-ul A si Convertorului A/D. Trebuie conectat extern la Vcc, chiar daca ADC nu este folosit. Daca ADC este folosit , ar trebui conectat la Vcc printr-un filtru trece -jos.

AREF :AREF este pinul de referinta analogica pentru Convertorul A/D

2.2 ORGANIZAREA MEMORIEI

ATmega 16 AVR are doua spatii de memorie principala, spatiul pentru Memoria de Date si pentru Memoria de Program. In plus, ATmega16 are si o memorie nevolatila EEPROM pentru memorarea datelor. Toate cele trei tipuri de memorie sunt cu adresare liniara.

Program Memory Map



ATmega 16 contine o memorie flash reprogramabila (In-system On-chip) de 16 Ko pentru programe. Deoarece toate comenzile pentru AVR sunt de 16 si 32 biti, Flash-ul este organizat ca 8Kx16. Pentru securitatea software-ului , spatiul pentru memoria de programe

Flash este impartit in doua sectiuni: sectiunea de program boot si sectiunea pentru programe de aplicatie.

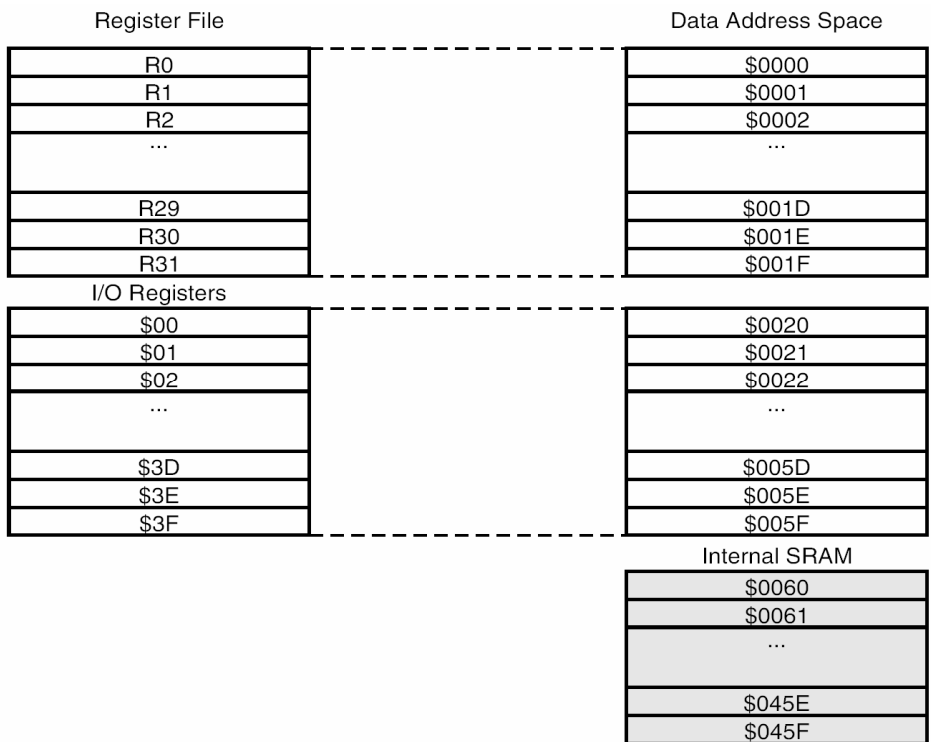
Memoria Flash suporta cel puțin 10000 de cicluri de scriere/ stergere. Counter-ul programului de la ATmega 16 (PC) are o lungime de 13 biti, ceea ce permite adresarea unei memorii de 8*1024 locatii de 16 biti.

Memoria de date SDRAM

Figura de mai jos arata cum este organizata memoria SDRAM ATmega 16. Primele 96 de locatii se refera la Fisierul de Registre, si urmatoarele 1024 de locatii sunt dedicate datelor interne SDRAM.

Registreele generale 26, 27, 28, 29, 30, 31 pot fi utilizate cu denumiri specifice:

- R26: X octet inferior R27: X octet superior
- R28: Y octet inferior R29: Y octet superior
- R30: Z octet inferior R31: Z octet superior



MEMORIA DE DATE EEPROM

ATmega 16 contine 512 octeti de memorie de date EEPROM. Este organizata ca spatiu separat de date, in care pot fi cititi si scrisi biti individuali. EEPROM-ul are o durata de viata de cel puțin 10,000 de cicluri scriere/stergere.

Accesul citire/scriere EEPROM

Registrii de acces EEPROM sunt in spatiul I/O.

Cand se citeste EEPROM, CPU este oprit timp de patru perioade de ceas inainte ca urmatoarea comanda sa fie executata. Cand se scrie EEPROM, CPU este oprit timp de doua perioade de ceas inainte ca urmatoarea comanda sa fie executata.

Registreele de adresa EEPROM – EEARH si EEARL

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	-	EEAR8	EEARH
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

Bitii 9 – 15 sunt biti rezervati in Atmega 16 si vor lua intotdeauna valoarea 0.

Bitii 0 - 8 sunt biti de adresa (total 9 biti, deci se adreseaza 0.5 KB)

Registrii de adresa de mai sus, **EEARH** si **EEARL** – specifica adresa EEPROM pentru cele 512 locatii ale spatiului EEPROM. Locatiile de memorie se adreseaza liniar de la 0 la 511.

Pentru operatia de scriere a EEPROM, registrul **EEDR** contine date care sa fie scrise in EEPROM la adresa data de registrul **EEAR**. Pentru operatia de citire a EEPROM, **EEDR** contine date citite de pe EEPROM la adresa data de **EEAR**.

Registrul de date EEPROM – EEDR

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Registrul de control EEPROM – EECR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

Biti 7..4 - acesti biti sunt biti rezervati la ATmega 16 si au valoarea zero.

Bit 3 - EERIE: Activarea EEPROM Ready Interrupt

Scrierea in EERIE a unui 1 logic, activeaza functia lui EEPROM Ready Interrupt ("pregatit de intrerupere"). Scrierea in EERIE a unui 0 logic dezactiveaza intreruperea . Functia EEPROM Ready Interrupt genereaza o intrerupere constanta cand EEWL este **sters**.

Bit 2 – EEMWE : EEPROM Master Write Enable (activarea functiei principale de scriere EEPROM)

Bitul EEMWE determina daca setarea lui EEWL la unu genereaza scrierea lui EEPROM. Cand este setat EEMWE, setarea lui EEWL va produce scriere date in EEPROM la adresa selectata. Daca EEMWE este zero, atunci setarea lui EEWL nu va avea nici un efect.

Bitul 1 – EEWL : EEPROM Write Enable (activarea scrierii EEPROM)

Semnalul de activare a scrierii EEPROM EEWL este semnalul de scriere a EEPROM. Cand adresa si datele sunt setate corect , bitul EEWL trebuie sa fie scris la unu pentru ca valoarea sa fie scrisa pe EEPROM. Bitul EEMWE trebuie sa fie scris la unu inainte ca unu logic sa fie scris pe EEWL , altfel nu va avea loc nici o scriere a EEPROM. Urmatoarea procedura trebuie urmata cand se scrie EEPROM (ordinea pasilor 3 si 4 nu este esentiala):

1. Asteptati pana cand EEWL devine zero
2. Asteptati pana cand SPMEN din SPMCR devine zero
3. Scrieti noua adresa EEPROM pe/la EEAR (optional)
4. Scrieti noile date EEPROM pe/ la EEDR(optional)
5. Scrieti unul logic pe/la bitul EEMWE in timp ce se scrie EEWL in EECR
6. In patru cicluri de ceas dupa ce s-a setat EEMWE , scrieti unul logic pe /la

EEWL.

EEPROM nu poate fi programat in timp ce CPU scrie memoria flash. Software-ul trebuie sa verifice daca programarea memoriei flash este completa inainte de a initia o noua scriere a EEPROM. Pasul doi este relevant doar in cazul in care programul software contine un Boot Loader ("activator de boot") care permite CPU sa programeze memoria flash. Daca memoria flash nu este niciodata updatata de catre CPU, atunci pasul doi poate fi omis.

Atentie: o intrerupere intre pasii 5 si 6 va anula ciclul de scriere ,caci activarea Master a EEPROM va fi anulata. Daca o rutina de accesare a EEPROM intrerupe o alta accesare EEPROM, atunci registri EEAR si EEDR vor fi modificati, astfel cauzand anularea accesului intrerupt al EEPROM

Bit 0 – EERE : Activarea citirii EEPROM

Cand se seteaza adresa corecta a registrului EEAR, bitul EERE trebuie sa fie scris pe unu logic ca sa declanseze citirea EEPROM. Accesul la citirea EEPROM se face cu o comanda si datele cerute sunt disponibile imediat. Cand EEPROM este citit, CPU este oprit timp de patru cicluri inainte ca urmatoarea comanda sa fie executata.

Oscilatorul calibrat este folosit la cronometrarea accesarilor EEPROM. Tabelul 1 arata timpii normali de programare a accesarilor EEPROM din CPU.

Table 1. EEPROM Programming Time

Symbol	Number of Calibrated RC Oscillator Cycles ⁽¹⁾	Typ Programming Time
EEPROM write (from CPU)	8448	8.5 ms

SPATIUL DE MEMORIE I/O

Toate I/O-urile de la ATmega 16 si perifericele sunt plasate in spatiul I/O. Locatiile I/O sunt accesate de catre comenzile IN si OUT , transferand datele dintre cei 32 de registrii de lucru si spatiul I/O. Registrii I/O cuprinsi intre valorile adreselor \$00 - \$1F sunt direct accesabile folosind comenzile SBI si CBI. La acesti registrii valoarea bitilor unici poate fi verificata utilizand comenzile SBIS si SBIC.

Cand se utilizeaza instructiunile specifice IN si OUT , trebuie folosite adresele I/O din zona \$00 - \$3F. Cand se adreseaza/aceseaza registrii I/O ca spatiu de date cu instructiunile LD si ST , trebuie adaugat la aceste adrese, \$20, adica salt peste zona registrelor de uz general.

2.3 UNITATEA TIMER DE 16 BITI

Unitatea timer /counter de 16 biti face posibila cronometrarea precisa a executiei programului (managementul evenimentelor), generarea de forme de unda si masurarea timpului semnalelor.

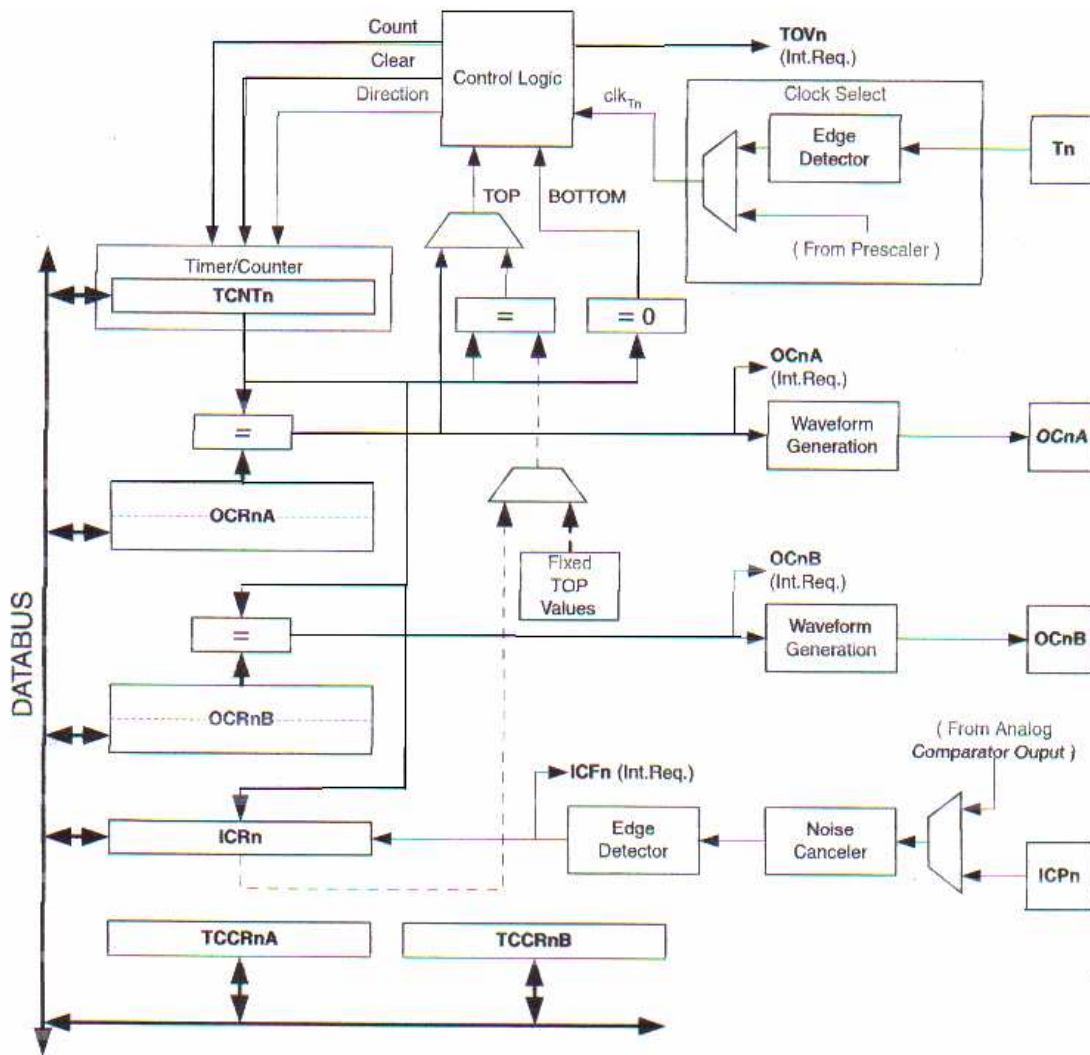
Principalele caracteristici sunt:

- Design true 16bit(permite 16bit PWM)
- Doua unitati de independente de comparare output
- Unitate de captura input
- Un anulator de zgomot pe intrare
- Stergerea cronometrului la potrivirea compararii(auto reincarcare)
- Puls de corectare a fazelor cu modulator (PWM)
- Perioada PWM variabila
- Generator de frecventa
- Counter extern de evenimente
- Patru surse de intrerupere independente

Majoritatea registrilor si referintelor de biti din aceasta sectiune sunt prezentate intr-o forma generala.Un "n" mic inlocuieste numarul timerului si un "x" mic inlocuieste unitatea de comparare de output. Atunci cand se foloseste registrul sau bitul definit intr-un program , forma precisa trebuie folosita.

O diagrama simplificata a timerului este prezentata in schema de mai jos.Plasarea exacta a pinilor I/O se regaseste in figura de mai jos. Registrii I/O accesibili de pe CPU , inclusiv bitii si pinii I/O sunt prezentati bolduiti.

Diagrama bloc :



REGISTRII

TCNT1, OCR1A/B si ICR1 sunt toti registrii de 16 biti. Proceduri speciale trebuie urmate cand se acceseaza registrii de 16 biti.

Registrii de control T/C(TCCR1A/B) sunt registrii de 8 biti si nu au restrictii de acces CPU. Semnalele de cerere de intrerupere sunt vizibile la TIFR (Registrul semnal de intrerupere timer). Toate intreruperile sunt mascate individual in TMISK. TIFR SI TMISK nu sunt prezentati in figura deoarece acesti registrii sunt folositi si de alte unitati timer.

T/C poate fi masurat intern , prin intermediul prescalei, sau de catre o sursa externa de ceas la pinul T1. Block-ul logic de selectare a ceasului controleaza ce sursa de ceas sau limita de T/C sa utilizeze pentru a creste(sau a descreste) valoarea sa. T/C este inactiv cand nu este

selectata nici o sursa de ceas. Output-ul de la selectarea logica a ceasului este numita cronometrarea ceasului(clkT1).

Registrii de comparare a output-ului cu buffere duble (OCR1A/B) sunt comparati cu valoarea T/C in permanenta. Rezultatul comparatiei poate fi folosit de catre Generatorul de Unde pentru a genera un PWM sau o frecventa variabila de output la pinul de Comparare a Output-ului(OC1A/B). Operatia de comparare va seta si semnalul de potrivire-comparare (OCF 1A/B) care poate fi folosit pentru a genera o cerere de intrerupere de comparare output.

Registrul de captura input poate captura valoarea T/C la o comanda data extern fie pe pinul de captura input sau pe pinii analog de comparatie. Unitatea de captura input include o unitate de filtru digitala pentru a micsora posibilitatile de a captura zgomot.

Valoarea TOP, sau valoarea maxima T/C poate fi definita in cateva moduri de operare fie de catre registrul OCR1A, de catre registrul ICR1 sau de catre un set de valori fixate. Cand se foloseste OCR1A ca valoare TOP intr-un mod PWM, Registrul OCR1A nu poate fi utilizat pentru a genera PWM output. In acest caz valoarea TOP va dubla bufferele permitand valorii sa fie schimabata intr-un ciclu de operare. Daca se cere o valoare TOP stabilita , registrul ICR1 poate fi utilizat ca o alternativa , permitand astfel ca OCR1A sa fie folosit ca output PWM.

DEFINITII

BOTTOM = counterul ajunge la capat cand devine 0x0000.

MAX = Counterul ajunge la maxim cand devine 0xFFFF(.....)

TOP = Counterul ajunge la valoarea TOP cand devine egal cu cea mai mare valoare din secventa de numarare. Valoarea TOP poate fi desemnata astfel incat sa fie una din aceste valori fixe: 0x00FF, 0x01FF, sau 0x003FF, sau valoarea memorata la Registrul OCR1A sau ICR. Aceasta desemnare depinde de modul de operare.

COMPATIBILITATE

T/C de 16 biti a fost updatat si imbunatatit fata de versiunile precedente ale T/C de 16 biti AVR. Acest T/C de 16 biti este perfect compatibil cu versiunile precedente in ceea ce priveste:

- Toate T/C de 16 biti relationeaza locatii de adrese de registrii I/O, inclusiv registrii de intrerupere timer;
- Locatiile bit din interiorul tuturor registrilor T/C , inclusiv registrii de intrerupere ai timerului;
- Vectorii de intrerupere.

Urmatorii biti de control au numele schimbat , dar prezinta aceeasi functionalitate si locatie de registru:

- PWM 10 s-a schimbat in WGM10

- PWM 11 s-a schimbat in WGM11
- CTC 1 s-a schimbat in WGM 12

Urmatorii biti sunt adaugati la registrii de control ai T/C de 16 biti:

- FOC1A si FOC1B sunt adaugati la TCCR1A
- WGM 13 este adaugat la TCCR1B

T/C de 16 biti are unele imbunatatiri care vor afecta compatibilitatea in unele cazuri speciale.

ACCESAREA REGISTRILOR DE 16 BITI

TCNT1.OCR1A/B si ICR1 sunt registrii de 16 biti care pot fi accesati de catre AVR CPU cu ajutorul bus-ului de date de 8 biti. Registrul de 16 biti trebuie sa fie accesat de byte folosind doua operatii fie de scriere, fie de citire. Fiecare timer de 16 biti are un singur registru de 8 biti pentru stocarea temporara a High byte de la accesul de 16 biti. Acelasi registru temporar este folosit de catre toti registrii de 16 biti in cadrul fiecarui timer de 16 biti. Accesarea byte-ului sczut declanseaza operatia de citire sau scriere de 16 biti. Cand byte-ul jos al unui registru de 16 biti este scris de catre CPU , byte-ul inalt stocat in registrul temporar si cel jos scris sunt ambii copiat in intr-un registru de 16 biti in acelasi ciclu de ceas. Cand un byte jos al unui registru de 16 biti este citit de catre CPU , byte-ul inalt al registrului de 16 biti este copiat intr-un registru temporar in aceleasi cicluri de ceas in care este citit si byte-ul jos.

Nu toate accesarile de 16 biti folosesc registrul temporar pentru byte-ul inalt. Citirea registrilor de 16 biti OCR1A/B nu implica folosirea registrului temporar.

Pentru a executa o scriere de 16 biti , byte-ul high trebuie sa fie scris inaintea byte-ului low. Pentru o operatie de citire 16 biti , byte-ul low trebuie sa fie citit inainte de byte-ul high.

Urmatoarele exemple de cod exemplifica accesarea registrilor timer de 16 biti presupunand ca nici o intrerupere nu updateaza registrul temporar. Acelasi principiu poate fi utilizat pentru a accesa direct registrii OCR1 A/B si ICR1. De retinut ca atunci cand se utilizeaza "C", compilatorul manevreaza accesul de 16 biti.

2.5 PORTURILE I/O

ATmega16 dispune de 32 de linii de I/O grupate în patru porturi de 8 biți. Porturile sunt denumite cu literele A, B, C și D. Fiecare pin al oricărui port se poate seta individual ca intrare sau ieșire fără să afecteze ceilalți pini. În plus, anumiți pini se pot utiliza pentru funcții speciale ale microcontrolerului. În Figura 3 este prezentată structura generală a unui pin.

Toți cei 32 de pini au fiecare câte o rezistență „pull-up” care poate fi activată sau dezactivată.

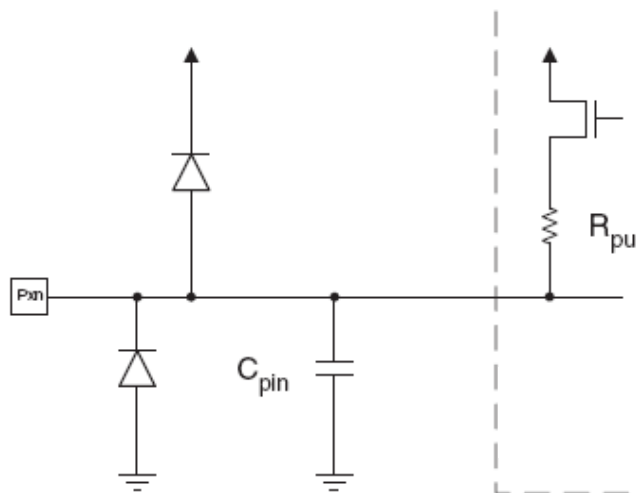


Figura 3. Structura generală a unui pin

Operațiile cu porturile se fac prin intermediul unui set de 3 registre alocate fiecărui port: PORTx, PINx și DDRx; x poate fi A, B, C sau D. Acești regiștri fac parte din categoria regiștri de intrare/ieșire de aceea instrucțiunile care pot lucra direct cu ei sunt CBI, SBI, IN și OUT.

Regiștrii DDRx stabilesc dacă un pin este intrare sau ieșire. Astfel, un bit cu valoarea „1” în registrul DDRx face ca pinul corespunzător să fie considerat ieșire; altfel pinul va fi intrare.

Regiștrii PORTx sunt utilizați pentru a **scrie** o valoare în portul corespunzător iar regiștrii PINx se folosesc pentru a **citi** valoarea prezentă pe pinii unui port.

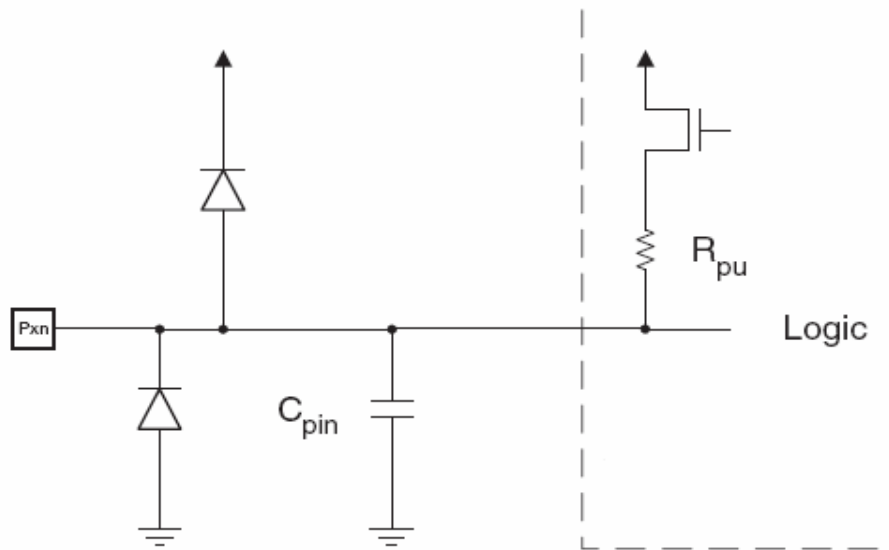
Activarea sau dezactivarea rezistențelor „pull-up” este determinată de bitul PUD din registrul SFIOR și de valorile regiștrilor PORTx și DDRx așa cum se poate observa în tabelul următor.

Bitul din DDRx	Bitul din PORTx	Bitul PUD	Tipul pinului	Rezistență pull-up
0	0	nu contează	intrare	inactivă
0	1	0	intrare	activă
0	1	1	intrare	inactivă
1	0	nu contează	ieșire	inactivă
1	1	nu contează	ieșire	inactivă

Toate porturile AVR au funcția Citeste – Modifica – Scrie atunci când sunt folosite ca porturi I/O digitale generale. Asta înseamnă că direcția unui pin port poate fi schimbată fără a schimba în mod intenționat direcția oricărui alt pin cu comenzile SBI și CBI. Aceeași regulă se

aplica cand se schimba valoarea drive-ului(daca este configurat ca output) sau a activarii/dezactivarii rezistorilor (daca sunt configurati ca input). Fiecare buffer de input are caracteristici similare de drive atat cu capacitatea sursei cat si cu sincronizarea. Driver-ul pinului este indeajuns de puternic ca sa activeze displayul LED direct. Toti pinii port au rezistori care pot fi selectati individual cu o sursa suplimentara de tensiune cu rezistenta invariabila. Toti pinii I/O au diode de protectie atat la Vcc cat si la impamantare asa cum este indicat in figura de mai jos :

I/O Pin Equivalent Schematic



Toti regisrii si referintele de bit din aceasta sectiune sunt scrisi in forma generala. Un "x" mic reprezinta litera care denumeste portul si un "n" mic reprezinta numarul bitului. Cand se utilizeaza registrul sau bitii intr-un program , trebuie sa fie folosita forma exacta. Adica, **PORTB3** pentru bitul numarul 3 in Portul B, prezentat general ca **PORTxn**.

Trei locatii de adrese de memorie I/O sunt alocate pentru fiecare port;

- una pentru fiecare registru de date - **PORTx**,
- Registrul directiei datelor – **DDRx**,
- port de intrare – **PINx**. Locatia pinilor port intrare este read only, in timp ce Registrul de Date si DDR sunt read si write. In plus, bitul PUD (dezactivare pull-up) din SFIOR dezactiveaza functia pull-up pentru toti pinii din toate porturile cand este setat.

Majoritatea pinilor au functii alternative pentru caracteristicile periferice ale dispozitivului.

Este de remarcat faptul ca activarea functiilor alternante ale unor pini port nu afecteaza utilizarea altor pini din port ca I/O general digital.

