

MICROCONTROLERE AVR ATMEL PE 8 BIȚI

Caracteristici generale

Microcontrolerul AVR are la bază un procesor RISC cu o arhitectură Harvard (adică unitatea centrală de procesare are memorie de program și memorie de date separate). Pe baza acestui nucleu RISC firma Atmel a dezvoltat mai multe familii de microcontrolere, cu diferite structuri de memorie și de interfețe I/O, destinate diferitelor clase de aplicații.

Aceste microcontrolere sunt destinate aplicațiilor simple cum ar fi: controlul motoarelor, controlul fluxului de informație pe portul USB, controlul accesului de la distanță (Remote Access Control), ș.a..

Familia de microcontrolere AVR pe 8 biți prezintă următoarele caracteristici:

1. Are memorie Flash programabilă integrată, folosită ca memorie de program. Toate procesoarele AVR au memorie de program integrată. Asta înseamnă că nu ai nevoie de memorii EPROM sau ROM externe pentru a stoca codul programelor. Programarea se poate face în două moduri: serial și paralel.
2. Are 32 de registre de lucru de 8 biți fiecare (acest lucru este specific procesoarelor RISC). Numărul mare de registre de lucru permite stocarea variabilelor în interiorul procesorului, în loc să fie stocate în memorie, lucru care necesită un timp mai lung pentru accesare. În acest fel crește viteza de execuție a programului (deci va rula mai repede).
3. Are memorie de date integrată de tip EEPROM (**E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory) și memorie RAM integrată. Unitatea centrală de prelucrare are arhitectură Harvard, iar memoriile de tip RAM și EEPROM ale microcontrolerului sunt memorii de date, fiind folosite pentru a stoca variabile, respectiv constante.
4. Are frecvența de lucru care poate fi controlată prin software de la 0 la 16 MHz. Cele mai multe instrucțiuni sunt executate într-un singur ciclu de tact, ceea ce duce la îmbunătățirea performanței de 10 ori față de procesoarele convenționale care operează la aceeași frecvență (de exemplu: Intel 8051).
5. Realizează funcția de RESET fără decuplarea circuitului.
6. Are un timer programabil încorporat cu circuit de prescalare separat, care este folosit pentru sincronizarea aplicațiilor.
7. Surse interne și externe de întrerupere.
8. Are un timer watchdog cu oscilator independent, care este folosit pentru recuperarea informațiilor în caz de apariția unei erori software, dar mai poate fi folosit și în alte aplicații.
9. Are 6 moduri de operare pentru economisirea energiei. Poate lucra în modurile de operare SLEEP și POWER DOWN, moduri ce economisesc energie, atunci când procesorul nu lucrează.
10. Multe cipuri au oscilator integrat RC, care micșorează și mai mult numărul componentelor în cazul microcontrolerelor care-l conțin.
11. Sunt prevăzute cu o gamă largă de dispozitive de I/O și de periferice încorporate.
12. Beneficiază de existența unui set unitar de instrumente software pentru dezvoltarea aplicațiilor.
13. Are o compatibilitate mare, pornind de la procesoare mici cu 8 pini și ajungând la procesoare cu 68 de pini.

Microcontrolerele din familia AVR prezintă o organizare de tip RISC din punctul de vedere al execuției unei instrucțiuni pe ciclu mașină. Prezența unor blocuri interne precum: oscilator intern, timere, unitate UART (Universal asynchronous receiver/transmitter), interfață SPI, rezistoare pull-up, PWM (pulse width modulation), convertoare analog-digitale, comparatoare analogice determină utilizarea acestor microcontrolere într-o gamă foarte largă de aplicații.

Specific arhitecturii Harvard, microcontrolerele AVR folosesc spații de memorie și magistrale de acces separate pentru coduri și pentru date. Memoria de program care este de tip FLASH integrată, poate fi programată fără scoaterea din sistem, ceea ce permite efectuarea rapidă de upgrade-uri în programe.

Instrucțiunile acestei familii de microcontrolere au fost proiectate pentru a reduce dimensiunea unui program scris în limbaj C sau în limbaj de asamblare. Posibilitatea programării memoriei FLASH și a memoriei EEPROM, determină ca aceste microcontrolere să aibă o largă utilizare datorată costului mic de dezvoltare a unei aplicații (au timpul de proiectare scurt).

Aceste microcontrolere au consum redus de energie. Domeniul tensiunilor de alimentare este cuprins între 1,8V și 5V. Prezintă 6 moduri diferite de operare cu consum redus de energie, ceea ce ne asigură că aceste microcontrolere nu vor consuma energie decât atunci când este nevoie.

Controlul software al frecvenței de tact garantează o viteză maximă de execuție atunci când este nevoie, iar în restul timpului microcontrolerul poate trece în modul STANDBY, unde consumul de energie este minim. Utilizarea acestor microcontrolere poate reduce semnificativ timpul de dezvoltare a unei aplicații datorită prezenței unui bloc de depanare în timp real, circuitul aflându-se chiar pe placă.

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

Microcontrolerele din familia AVR pot fi atât microcontrolere de uz general, cât și microcontrolere cu funcții specializate, împărțindu-se în următoarele categorii:

1. **Familia TINY AVR** – cuprinde microcontrolere destinate unor aplicații simple, cu 1÷8 KB memorie de program.
2. **Familia MEGA AVR** – cu un set dezvoltat de instrucțiuni și cu 4÷256 KB memorie de program.
3. **Familia XMEGA AVR** – destinată unor aplicații complexe. Acestea au unități DMA (Direct Access Memory), module pentru criptare, ș.a. și folosesc un spațiu de memorie de program de 16 ÷ 384 KB.
4. **Microcontrolere AVR destinate unor aplicații specifice**, care se împart în mai multe categorii:
 - 4.1. **LIGHTING AVR**: microcontrolere ce conțin un procesor de semnal, fiind special conceput pentru controlul motoarelor de curent continuu.
 - 4.2. **LCD AVR**: microcontrolere ce conțin un controler pentru adresarea dispozitivelor LCD
 - 4.3. **CAN AVR**: microcontrolere ce conțin o interfață CAN implementată hardware.

Microcontrolerele din familia AVR de 8 biți au un set de 131 instrucțiuni și 32 de registre de uz general. Cele 32 de registre sunt direct adresabile de unitatea aritmetică și logică (UAL), permițând accesarea a două registre independente într-o singură instrucțiune. Astfel se obține o eficiență sporită în execuție, fiind de până la zece ori mai rapide decât microcontrolerele convenționale de tip CISC.

Microcontrolerul ATMEGA 16

Arhitectura AVR (firma ATMEL)

ATMEGA 16 este un microcontroler realizat de firma Atmel, având o tehnologie CMOS de 8 biți, de mică putere și o arhitectură RISC AVR îmbunătățită.

Caracteristicile principale ale acestui microcontroler sunt:

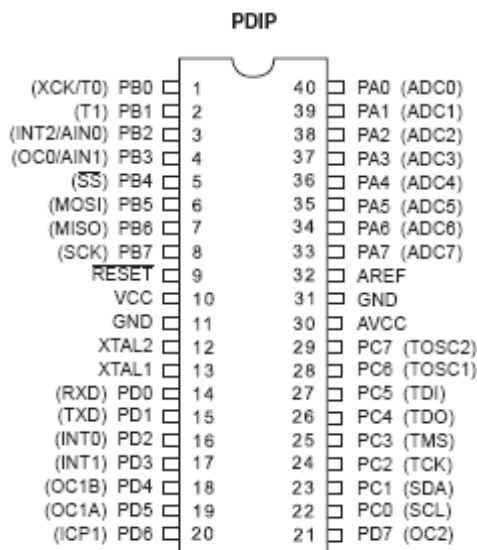
- 16 KB de memorie FLASH reinscriptibilă pentru stocarea programelor;
- 1 KB de memorie RAM;
- 512 B de memorie EEPROM;
- două numărătoare/temporizatoare de 8 biți;
- un numărător/temporizator de 16 biți;
- un convertor analog digital de 10 biți, cu 8 intrări (intrări multiple);
- un comparator analogic;
- 4 canale PWM;
- conține 3 interfețe pentru comunicație: un modul USART pentru comunicație serială (port serial), interfață serială TWI și interfață serială SPI;

- un cronometru cu oscilator intern;
- oferă 32 de linii I/O organizate în patru porturi (PA, PB, PC, PD) cu 8 biți fiecare;
- are o magistrală generală de date la care sunt conectate mai multe module și anume: unitatea aritmetică și logică (UAL), registrele generale, memoria RAM și memoria EEPROM, liniile de intrare (adică porturile) 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, de decodare și de 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 microcontrolerului să execute instrucțiunile foarte rapid.

În modul POWER DOWN se salvează conținutul registrelor, dar se blochează oscilatorul, dezactivându-se toate celelalte funcții ale chip-ului până la următoarea întrerupere externă sau la următorul RESET hardware. În modul POWER SAVE, timer-ul asincron continuă să meargă în timp ce restul dispozitivelor sunt oprite. 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, iar în modul STANDBY EXTINS (Extended Standby Mode), atât oscilatorul principal cât și timer-ul asincron continuă să funcționeze.

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

Descrierea pinilor la ATMEGA 16



Vcc – Sursa de curent

GND – Masa

Port A (PA7 .. PA0)

Portul A servește drept port de intrări analogice pentru convertorul A/D. Mai poate fi folosit, de asemenea, și ca un port bidirecțional I/O de 8 biți, în cazul în care convertorul A/D nu este folosit. Pini de port pot fi conectați opțional la sursa de curent Vcc prin rezistori interni (selectați pentru fiecare bit). Buffer-ele de ieșire ale portului A au și caracteristici de amplificare .

Port B (PB7.. PB0)

Portul B este un port I/O de 8 biți bidirecțional cu rezistori interni (opționali), dar poate îndeplini și funcții speciale ale microcontrolerului ATMEGA 16. Buffer-ele de ieșire ale portului B au și ele caracteristici de amplificare.

Port C (PC7...PC0)

Portul C este un port I/O de 8 biți bidirecțional cu rezistori interni (opționali), poate îndeplini și funcții ale interfeței JTAG (de depanare), cât și alte funcții speciale ale microcontrolerului ATMEGA 16. Buffer-ele de ieșire ale portului C au caracteristici de amplificare. Dacă interfața JTAG (de depanare) este activată, rezistorii pinilor PC5(TDI), PC3(TMS) și PC2(TCK) vor fi activați, chiar dacă are loc o resetare.

Port D (PD7...PD0)

Portul D este un port I/O de 8 biți bidirecțional cu rezistori interni conectați opțional la sursa de curent Vcc (selectați pentru fiecare bit), dar poate îndeplini și funcții speciale ale microcontrolerului ATMEGA 16. Buffer-ele de ieșire ale portului D au caracteristici de amplificare.

RESET - Un nivel scăzut la acest pin mai mare ca durată decât o valoare prestabilită, va genera o inițializare.

XTAL 1 - reprezintă intrare pentru amplificatorul inversor al oscilatorului.

XTAL 2 - reprezintă ieșire pentru amplificatorul inversor al oscilatorului.

AVCC - este pinul de alimentare pentru portul A și convertorul A/D. Trebuie conectat extern la Vcc, chiar dacă convertorul analog-digital nu este folosit. Dacă convertorul analog-digital este folosit, ar trebui conectat la sursa de curent Vcc printr-un filtru trece-jos.

AREF - este pinul de referință analogică pentru convertorul A/D.

Unitatea centrală de prelucrare pentru AVR

Funcția principală a nucleului unității centrale de prelucrare (UCP) este aceea de a asigura execuția corectă a programului. Din acest motiv, nucleul UCP este capabil să acceseze memoriile, să execute calcule, să controleze perifericele și întreruperile.

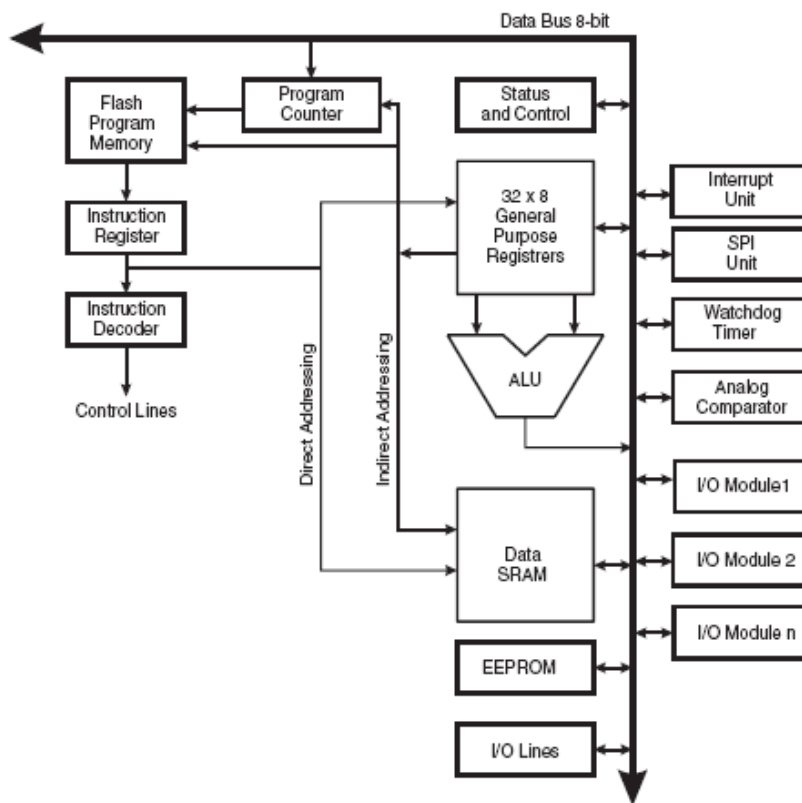


Figura 1 – Schema bloc a nucleului UCP la microcontrolere AVR

Pentru a maximiza performanța, AVR folosește o arhitectură Harvard cu memorii și magistrale separate pentru program și date. Instrucțiunile din memoria programului sunt executate într-un singur nivel în timp ce o instrucțiune este executată și următoarea este adusă de la memoria de program. Acest concept permite executarea instrucțiunilor la fiecare ciclu de ceas. Memoria de program este o memorie de tip FLASH reprogramabilă.

Registreele la AVR sunt în număr de 32×8 biți fiecare, scopul acestor microcontrolere fiind acela de a accesa registrele într-un singur ciclu de ceas. Acest singur timp de acces se datorează unității aritmetice și logice (UAL=Unitatea Aritmetică și Logică). Într-o unitate aritmetică și logică operația are loc astfel: operanzii sunt scoși din registru, se efectuează operația și rezultatul este introdus în registri, toate acestea fiind executate într-un singur ciclu de ceas. Șase din cele 32 de registre pot fi folosite ca trei registre de 16 biți cu acces indirect la informații, permițând astfel calcularea eficientă a adresei. Una dintre aceste adrese poate fi folosită pentru a căuta tabele în memoria de program.

Unitatea aritmetică și logică efectuează (suportă) operații aritmetice și logice între registre sau între o constantă și un registru. După efectuarea unei operații aritmetice registru afișează rezultatul operației.

Programul furnizează salturi condiționate, necondiționate și apelări de instrucțiuni capabile să acceseze tot spațiul de adresă. Majoritatea instrucțiunilor AVR sunt formate dintr-un cuvânt de 16 biți.

Spațiul memoriei de program (FLASH) este împărțit în două secțiuni, secțiunea BOOT și secțiunea de aplicare a programelor. Secțiunea BOOT are biți speciali pentru protecția la scriere și citire/scriere. De exemplu instrucțiunea SPM cu ajutorul căreia se scrie în memoria FLASH aplicații trebuie să fie în secțiunea BOOT.

În timpul întreruperilor sau a apelării subrutinelor, adresa de întoarcere este conținută în stivă. Stiva este efectiv alocată în informațiile generale SRAM și în consecință mărimea ei este limitată doar de mărimea totală a SRAM și de uzura ei. Toți utilizatorii de program trebuie să inițializeze

indicatorul de stivă SP (Stack Pointer) înainte ca subrutina sau întreruperea să fie executată. SP se poate citi/scrie în spațiul de I/O. Informațiile din SRAM pot fi accesate cu ușurință prin cele cinci moduri diferite de adresare suportate de arhitectura AVR.

Spațiile de memorie în arhitectura AVR sunt liniare și normale.

Modulele întreruperilor au registrele de control în spațiul I/O și în Registrul STATUS se află bitul de întrerupere globală. Toate întreruperile au prioritate în funcție de locul ocupat în tabelul de vectori de întrerupere. Cu cât este mai jos situat în tabel vectorul de întrerupere cu atât acea întrerupere are prioritate mai mare. Prioritatea mai mare o are întreruperea cu vectorul cel mai slab plasat în tabel.

Spațiul de memorie I/O conține 64 de adrese pentru funcțiuni periferice ale UCP precum: controlul registrelor, indicatorul SPI sau alte funcțiuni de I/O. Memoria I/O poate fi accesată direct sau indirect.

Execuția în timp a instrucțiunilor

Această parte descrie în general timpul necesar executării instrucțiunilor. UCP a MC AVR este controlată de ceasul UCP generat direct de la sursă. Nu se folosește nici un ceas interior.

Figura 6 ne prezintă pipeline-ul de execuție al unei instrucțiuni permise în arhitectura Harvard și accesul rapid la fișierul de registre. Acesta este conceptul de bază pentru a obține mai mult de 1 MIPS/MHz și cele mai bune rezultate din punct de vedere funcțiuni/cost, funcțiuni/timp și funcțiuni/unitate.

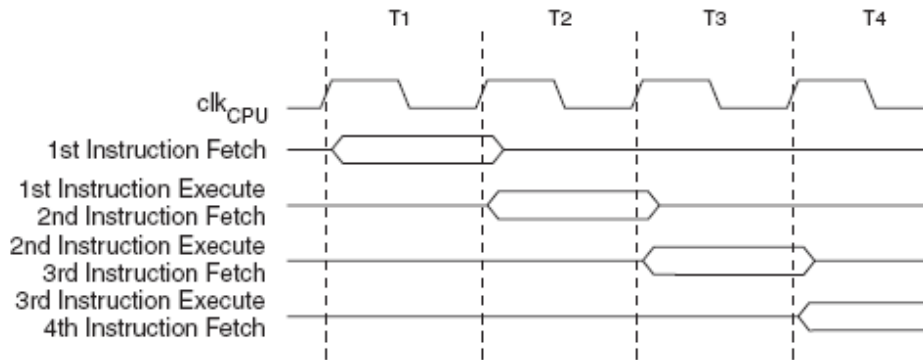


Figura 6 - Instrucțiuni paralele de execuție

Figura 7 ne arată timpul de lucru cu registrele. Într-un singur ciclu de ceas o operație a UAL folosește 2 registre pentru a executa calculul respectiv, iar rezultatul este stocat înapoi în registrul de destinație.

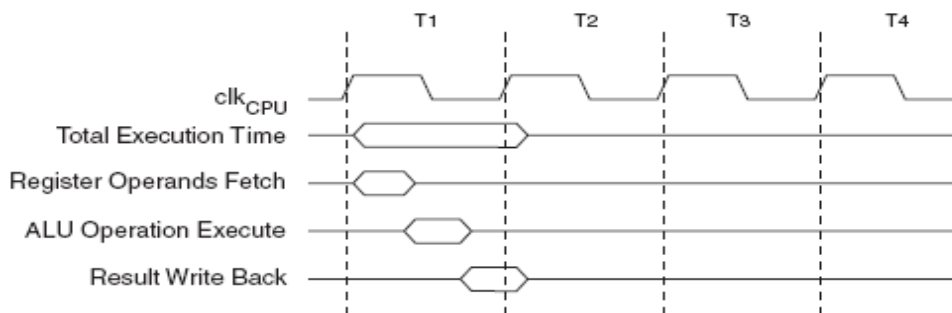


Figura 7 - Operații UAL într-un singur ciclu de ceas

Întreruperile

MC AVR furnizează mai multe tipuri de întreruperi. Aceste întreruperi și vectorul de resetare au câte un vector de program fiecare aflat în spațiul memoriei de program. Tuturor întreruperilor le sunt alocate individual biți care trebuie scriși logic 0 dată cu bitul GIE (Global Interrupt Enable) în registrul de stare cu scopul de a permite întreruperea. În funcție de starea în care se află PC (Program Counter), întreruperile pot fi invalidate, atunci când Boot Loader-ul este programat. Această tehnică îmbunătățește securitatea.

Cele mai joase adrese din memoria de program sunt definite ca vectori de resetare și întrerupere. În funcție de listă se determină și nivelurile de prioritate ale diferitelor întreruperi. Cu cât întreruperea are nivelul mai jos, cu atât prioritatea este mai mare. RESET are cea mai mare prioritate, iar după aceasta este INTO - cererea de întrerupere externă 0. Vectorii de întrerupere pot fi mutați la începutul secțiunii Boot Flash prin setarea bitului IV SEL din registrul global de control al întreruperilor (GICR). Vectorul de RESET poate fi, de asemenea, mutat la începutul aceleiași secțiuni prin programarea secțiunii BOOTRST.

Când apare o întrerupere, bitul GIE (bitul I – de întrerupere) este șters și toate întreruperile sunt invalidate. Utilizatorul de software poate scrie 1 logic în bitul I pentru a permite executarea întreruperilor. Toate întreruperile permise pot, la rândul lor, întrerupe întreruperile de rutină. Bitul I este automat corectat când instrucțiunea RETI este executată.

Practic sunt două tipuri de întreruperi:

1. Primul tip este declanșat de evenimentele care setează indicatorul de întrerupere. Pentru aceste întreruperi PC este trimis la vectorii de întrerupere în scopul executării întreruperilor de rutină și hardware-ul șterge indicatorul de întrerupere corespunzător. Indicatorul de întrerupere poate fi șters și prin scrierea cu 1 logic. Dacă se întâmplă o întrerupere în timpul în care bitul care permite întreruperea este șters, atunci indicatorul de întrerupere va fi setat să rețină întreruperea până când aceasta va putea fi permisă sau indicatorul este șters de software. În caz similar se procedează atunci când este vorba de ștergerea indicatorului GIE.

2. Cel de-al doilea tip de întreruperi este dat atâta timp cât condiția de întrerupere este prezentă. Aceste întreruperi nu au neapărat indicator de întrerupere. Dacă condiția de întrerupere dispare înainte ca întreruperea să fie permisă, întreruperea nu va mai fi executată. Când AVR iese dintr-o întrerupere se întoarce la programul principal și mai execută o dată instrucțiunile înainte de a interveni altă întrerupere. Starea registrului nu este automat stocată când apare o întrerupere de rutină, nici când revine din întreruperea de rutină. Acesta trebuie susținut de software.

De exemplu:

1. Când se folosește instrucțiunea CLI (Clear Interrupt) pentru invalidarea întreruperilor, întreruperea va fi invalidată imediat. Nici o întrerupere nu va mai fi executată după acționarea instrucțiunii CLI, chiar dacă se întâmplă simultan cu instrucțiunea CLI.
2. Când se folosește instrucțiunea SEI (Set Interrupt) pentru a permite întreruperi, instrucțiunea SEI este rulată înaintea oricărei instrucțiuni aflate în așteptare.