

Citas funkcijas

Watchdog taimeris (sargtaimeris)

Watchdog taimeris reizēm tiek saukts arī par COP (computer operates properly – dators darbojas kārtīgi), tiek izmantots lai vērotu programmatūras izpildi. Galvenā ideja šim taimerim ir tā, ka tad kad tas ir palaists tas sāk atpakaļ skaitīšanu. Kad tiek sasniegta 0 tiek palaista restartēšana, kas pārlādē kontrolieri un restartē programmu. Lai izvairītos no kontroliera atiestatīšanas programmatūrai vajag pārinicializēt watchdog taimeri pirms tas ir sasniedzis 0.

Mērķa aplikācijas šim taimerim ir: tas tiek izmantots lai apstiprinātu noteiktas pozīcijas programmas kodā sasniegšanu noteiktā laikā. Tātad vienalga kad programma novirzās no sava normālās izpildes ceļa un nenotiek taimera pāriestatīšana pareizajā momentā, pāriestatīšana tiks palaista, kas iespējams atrisinās problēmu. Tas noved pie liela skaita situācijām kur šis taimeris nav derīgs: vienalga kad programma uzvedas citādi kā parasti bet tomēr tai izdodas noresetot watchdog taimeri īstajā laikā un visās situācijās kur iemesls kurš bija vainīgs pie pāriestatīšanās nepazūd pēc watchdog taimera nulles sasniegšanas.

Tā kā watchdog taimeris tiek izmantots lai sekotu pareizai programmas izpildei, kas nozīmē ka abas pārbaudes vai kontrolieris izpilda pareizās instrukcijas un vai programma paspēj izpildīt pāriestatīšanas norādes laikā, tiek paliktas nostāk no citiem kontrolieru moduļiem lai ļautu autonomām operācijām darboties. Kā sekas var būt tas ka watchdog taimerim piemīt savi iekšējie oscilatori kas netiek iespaidoti pie iemigšanas režīmiem kas atslēdz sistēmas pulksteni. Watchdog taimeris prezentē pats savu iedarbināšana bitu un piedāvā dažus bitus kuri kontrolē savus taimauta periodus. Lai izvairītos no šī taimera netīšas izslēgšanas, noteikta procedūra ir jāveic lai to izslēgtu vai lai mainītu tā uzstādījumus. Piemēram HCS12 pieprasa lai programma no sākuma uzraktsta 0x55 un tad 0xAA lai taimeris pāriestatītu reģistru. ATmega16 pieprasa programmai uzstādīt divus bitus reģistrā uz 1 un tad pāriestatīt watchdog taimera ieslēgšanas bitu četru ciklu laikā.

Enerģijas taupīšana un Miega režīms

Mikrokontrolieri bieži vien tiek izmantoti mobilās sistēmās kas ir darbināmas ar baterijām. Tā ka zems enerģijas patēriņš ir ļoti nopietns rādītājs mikrokontrolierim. Lai samazinātu enerģijas patēriņu E, ir iespējams izmantot dažas tehnikas.

Takts frekferences samazināšana

Šī tehnika ļauj kontrolierim darboties kā ierasts bet pie zemākas takts frekferences. Enerģijas patēriņš ir proporcionāls frekfencei. Tā kā kontrolieriem ir statisks dizains tad frekfence var tikt samazināta patvaļīgi.

Lai samazinātu šo īpašību izstrādātāji protams var uzstādīt kontrolieriem minimālo frekfenci kas ir nepieciešamas lai izpildītu aplikācijas uzdevumu. Bet ar piemērotu shēmu ir iespējams arī dinamiski samazināt frekfenci brīžos kad tas ir iespējams. Tātad neskatoties uz to ka frekfencei iespējams ir jābūt augstai dažu uzdevumu veikšanai tā tomēr var tikt samazināta dīkstāves brīžos.

Voltāžas samazināšana.

Šī metode izmanto faktu ka enerģijas patēriņš ir proporcionāls izmantojamās voltāžas kvadrātam. Tātad izmantojamās voltāžas samazināšana ļoti iespaido patērētās enerģijas daudzumu. Diemžēl šo lielumu nav iespējams mainīt patvaļīgi. Kontrolieris ir uzstādīts noteiktai voltāžas amplitūdai. Ja voltāža nokrītas zem norādītā līmeņa tad kontrolieris var sākt uzvesties patvaļīgi, minimālā voltāža pie kuras kontrolieris ir spējīgs darboties ir atkarīga no apkārtējās vides.

Tāpat kā takts frekvences samazināšana tad arī voltāžas samazināšana var tikt veikta gan statiski gan dinamiski, tā var tikt apvienota ar miega režīmu, tā kā tas ir 8051.

Neizmantoto moduļu izslēgšana.

Šī metode izmanto faktu ka kontrolieris sastāv no vairākiem moduļiem kas var netikt izmantoti vienlaicīgi. Tā kā katrs aktīvais modelis patērē enerģiju tā acīmredzami ir laba ideja, izslēgt neizmantotos moduļus. Tātad ja kontrolieris veic tikai iekšējās darbības tad tā I/O komponentes var tikt atslēgtas. No otras puses skatoties ja kontrolieris gaida uz kādu ārējo notikumu, tā CPU un citas daļas var tikt atslēgtas līdz brīdim kad tas notiek. Jāievēro ka tad kad tiek atslēgts I/O modulis tas var novest pie tā, ka visi pini tiek uzstādīti uz ievades režīmu, tātad nav iespējams vadīt izejas pinu un to izslēgt vienlaicīgi.

Šī metode galvenokārt tiek izmantota kontroliera miega režīmā. Kontrolieris cenšas apkalpot dažādus miega režīmus dažādām komponentēm. Daži režīmi pat iet tik tālu ka atslēdz visus iekšējos modeļus ieskaitot ārējo oscilatoru. Tikai daži ārējie notikumi var pamodināt kontrolieri kas ir iegājis šādā režīmā, visbiežāk tā ir pāriestatīšanās vai kāds ārējais pārtraukums. Tajā pat laikā kontrolieri cenšas uzmantot iekšējos oscilatorus, lai ņemtu paraugus no ievades līnijas.

Pamošanās no miega režīma prasa dažus ciklus, bet var arī ieilgt līdz dažām milisekundēm. Tas ir tādēļ, ka oscilatoram ir nepieciešams laiks lai stabilizētos pēc aktivizācijas. Pēc pamošanās arī jāpārliciecinās par modeļu stāvokli. Daži modeļi var kļūdaini izsaukt pārtraukumus ja pārtraukums bija ieslēgts pirms moduļa izslēgšanās. Tātad vajag veikt papildus pārbaudes pirms miega režīma aktivizēšanas. Dažos gadījumos ir pat nepieciešams manuāli neaktivizēt neizmantotos moduļus pirms ieiešanas miega režīmā, tā lai tie neizmantotu enerģiju lieki.

Optimizēts dizains

Visbeidzot protams ir iespējams optimizēt kontroliera enerģijas patēriņu jau izveidojot tā uzbūvi. Labs piemērs tam ir MSP430 ģimene, kas ir optimizēta ņemot vērā enerģijas patēriņu un patērē mazāk kā 400 μ A pie normālas darbības. Salīdzinājumam citi kontrolieri patērē 1.1 mA un 350 μ A miega režīmā.

Atiestatīšana (Reset)

Atiestatīšana ir nozīmīga funkcija mikrokontrolieros, kas bieži tiek izvietoti vides iespaidā kas var novest pie programmatūras vai datordetaļu bojājumiem. Pie šādiem nosacījumiem, sistēmas atiestatīšana vienkārši nozīmē atgriezt to pie tās sākuma iestatījumiem. Un nodrošināt bezklūdu operācijas. Tātad mikrokontrolieris var reaģēt uz daudzveidīgiem atiestatīšanas nosacījumiem un izsaukt atiestatīšanu.

Tiklīdz atiestatīšanas nosacījumi ir aktivizēti, mikrokontrolieris inicializē visus savus reģistrus uz sākuma vērtībām kas parasti nozīmē ka visi pini tiek pārvērsti par ievades piniem. Kontrolieris

saglabā šādu stāvokli tik ilgi kamēr atiestatīšanās stāvoklis ir aktīvs, kā arī nogaida kamēr oscilators un enerģija stabilizējas. Pēc tam kontrolieris izpilda pirmo programmas instrukciju kas atrodas programmas adresē 0x0000. Aplikācijas programmētājs tur parasti ievieto atiestatīšanas standarta funkcijas, tādas kā, sākuma kodu, steka rādītāja inicializāciju un citas lietas. Pēdējā atiestatīšanas instrukcija ir pāriešana uz galveno programmu, līdz ar to normālas programmas izpildi.

Ieslēgšanas Atiestatīšana

Ieslēgšanas atiestatīšana izraisa atiestatīšanu jebkurā brīdī kad voltāžā pārsniedz noteiktu robežu. Tas nodrošina ka sistēma tiek atiestatīta pēc ieslēgšanas.

Brown-out atiestatīšana (BOR)