En energisnål design kräver finlir i flera steg

Rätt kisel, kod och verktyg ger optimal energiförbrukning

EXPERTARTIKEL

Av Keith Odland, Silicon Laboratories

Keith Odland har marknadsansvar för styrkretsar på Silicon Labs. Tidigare har han ansvarat för tekniska lösningar på Future Electronics, där han bland annat utvecklat marknads- och affärsutvecklingsprogram för 8- och 16-bitars styrkretsar. Han har även varit med och grundat företaget Technology Kitchen, som utvecklat specialinstrument för flygindustrin, samt arbetat på Dell Computer och Eaton.

Konstruktion av energisnåla system omfattar allt från kiselprocessen till den mjukvara som körs på den inbyggda plattformen. Tre viktiga parametrar som avgör energieffektiviteten hos en styrkrets är: effektförbrukning i aktivt läge, effektförbrukningen i standby samt pulslängden (duty cycle).

Ett energisnålt standby-läge kan göra att en MCU tycks vara extremt energieffektiv, men verklig prestanda visar sig först när hänsyn tagits till samtliga faktorer som styr den aktiva effektförbrukningen. Avvägningar mellan processteknik, kretsarkitektur och programvarukonstruktion är några av de många beslut med subtila och ibland oväntade resultat. Det sätt på vilket funktionella block kombineras i en MCU får stor inverkan på energieffektiviteten. Även till synes obetydliga förändringar av hårdvarans implementering kan leda till stora förändringar för den sammanlagda effektförbrukningen över ett systems livslängd.

MÄT- OCH LARMSYSTEM drivs exempelvis ofta av ett batteri under cirka tio år. En liten ökning av strömförbrukningen för en sensoravläsning kan leda till att livslängden kortas med flera år. En enkel rökvarnare som detekterar rökpartiklar i luften en gång per sekund kommer att utföra 315 miljoner avläsningar under sin livstid.

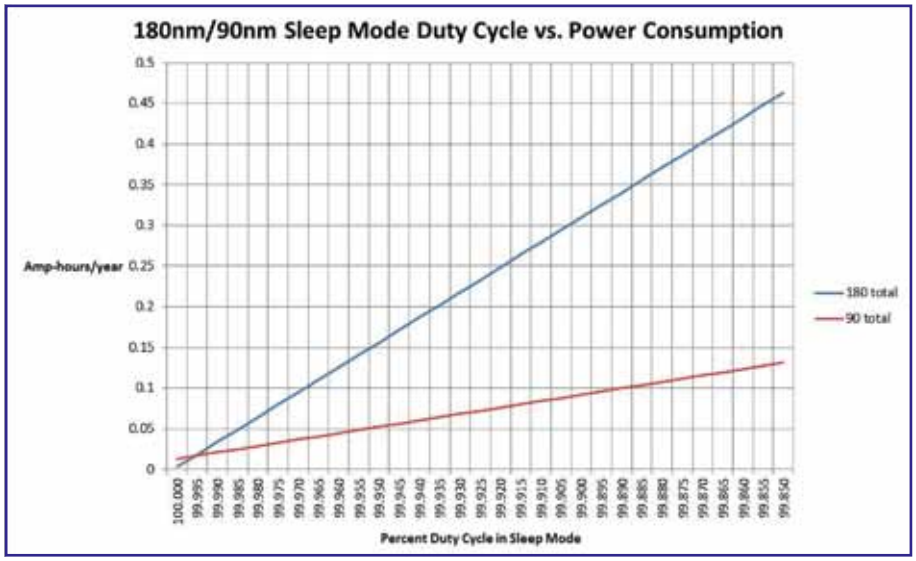
Aktivitetskvoten eller pulslängden hos en enkel rökvarnare är relativt låg. Varje sensoravläsning tar kanske inte mer än några hundra mikrosekunder. En stor del av den tiden ägnas åt kalibrering och inställningar när MCU:n väcker AD-omvandlaren och andra analoga delar och låter dessa stabilisera sig. I detta fall kommer pulslängden troligen att leda till en konstruktion som är inaktiv cirka 99,98 procent av tiden.

En rökvarnare är relativt enkel. Ta istället en mer komplicerad RF-konstruktion där resultat skickas över ett givarnät till en värdapplikation. Givaren måste vara uppmärksam på aktivitet från en masternod så att den antingen kan signalera att den fortfarande finns i nätverket eller tillhandahålla insamlad information till routern. Denna ökade aktivitet behöver inte påverka den sammanlagda pulslängden. Fler funktioner kan istället utföras under varje aktiveringsperiod om man använder en krets med högre prestanda. Nyckeln ligger i en förståelse för sambandet mellan processteknik, MCU-arkitektur och programvaruimplementering.

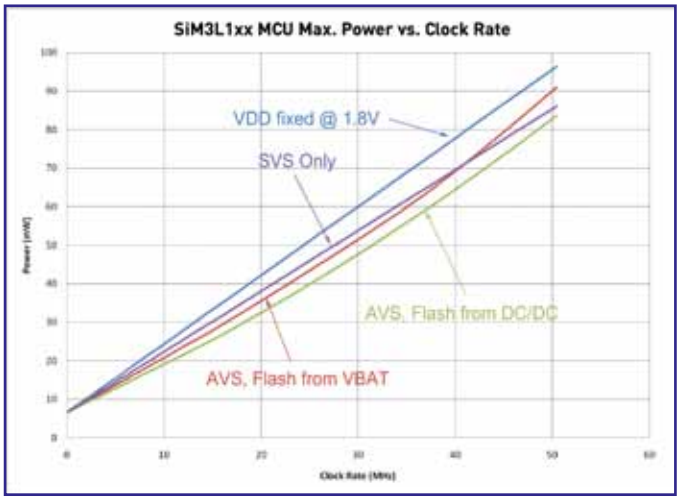
Effektförbrukningen för en logikkrets ges av formeln CV^2f, där C är kapacitansen, V är matningsspänningen och f är arbetsfrekvensen. Spänningen och kapacitansen bestäms av processtekniken. Arbetsspänningen för CMOS-logik har minskat i takt med att transistorernas storlek minskat. Idag ligger den under 2 V, vilket har stor betydelse för effektförbrukningen.

ÄVEN KAPACITANSEN MINSKAR när transistorerna blir mindre. För en viss given logisk funktion kommer en modernare process att ha lägre kapacitans än sina föregångare. Dessutom gör avancerade konstruktionstekniker att man kan minska den sammanlagda switchfrekvensen genom att bara använda de transistorer som har något att göra – tekniken kallas klockgrindning.

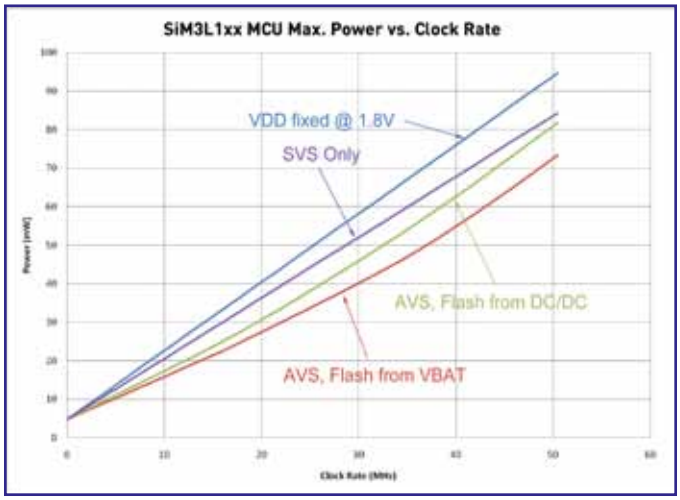
Läckströmmarna ökar däremot med Moores lag, vilket måste tas med i beräkningen. I likhet med den aktiva effektförbrukningen har kretskonstruktionen mycket stor inverkan på läckaget. Likt klockgrindning kan kraftgrindning väsentligt förbättra effekterna hos läckaget och göra så att det går att välja mer avancerade processnoder vid kort pulslängd – även om



Figur 1. Det är viktigt att förstå pulslängdens (duty-cycle) inverkan på energiförbrukningen vid val av processgeometri.



Figur 2. Resultatet om ingen (VDD), statisk (SVS) och aktiv (AVS) spänningsskalning används då batterispänningen är 3,6 V.



Figur 3. Resultatet om ingen (VDD), statisk (SVS) och aktiv (AVS) spänningsskalning används då batterispänningen är 2,4 V.

en äldre processteknik kan ge lägre teoretiskt läckage.

För varje funktionsuppsättning finns det en lämplig processteknik. Men bara för att kretsen kommer att ägna mycket tid i sovläge kan man inte enbart förlita sig på en processteknik med lägsta teoretiska läckage. I sovläge går det att stänga av matningen till stora delar av MCU:n, vilket utesluter läckagekomponenten ur ekvationen. Läckage är ett större problem när kretsar är aktiva, men det kan uppvägas av transistorer som switchar mer effektivt.

Läckströmmen är ungefär fem gånger högre för en 90 nm än en energisnål 180 nm-process. Samtidigt är effektförbrukningen i aktivt läge en faktor fyra lägre hos en finare geometrin – och då handlar det om ett mycket högre värden. Ta exempelvis en MCU i 180 nm. Den har en aktiv strömförbrukning på 40 mA och drar 60 nA i djupt sovläge. Jämför detta med en 90 nm-krets vars aktiva strömförbrukning ligger på 10 mA, medan den drar 300 nA i sovläge.

I exemplet måste MCU:n bara vara aktiv 0,0008 procent av tiden för att 90 nm-kretsen totalt sett ska bli energieffektivast. Så om systemet är aktivt en sekund per dag blir 90 nm-versionen cirka 1,5 gånger mer energieffektiv än 180 nm-versionen. Det är alltså viktigt att förstå pulslängden vid val av processgeometri. (Se figur 1.)

NÄR LÄMPLIG PROCESSTEKNIK valts kan kretskonstruktören optimera energiprestanda ytterligare genom klockgrindning. Tekniken ökar komplexiteten eftersom konstruktören måste känna till vilka logikvägar som kräver en klocksignal vid en given tidpunkt.

Flertalet MCU:er utnyttjar en hierarkisk struktur vid fördelningen av klocksignaler och spänningsnivåer till kretsens olika delar. Funktionella enheter, som block för instruktionsbehandling och periferier, är indelade i grupper. Dessa grupper kommer att matas av ett separat klockträd och kraftnät. Klocksignalen för varje grupp härrör från samma klockkälla med hjälp av en frekvensdelare eller multiplikator. På liknande sätt kommer spänningen till varje grupp av periferienheter att styras av en uppsättning krafttransistorer och spänningsregulatorer.

MCU:er har traditionellt utnyttjat en relativt enkel metod för klockgrindning, där hela klockträd stängs av om inga funktionella enheter i en grupp är aktiva. Det innebär dock att logik som inte utför något nyttigt arbete kan klockas i grupper som är aktiva. Exempelvis kan en adderare i en CPU-kärna motta en klocka även om instruktionen i fråga är en sidogren.

Förbättrade konstruktionsverktyg och -tekniker har gjort det möjligt att förfina klockgrindningen så att inga periferienheter eller funktionell enhet mottar en klocksignal om de inte är aktiva under en viss cykel.

Spänningsskalning minskar också energiförbrukningen genom att vissa enheter kan matas med lägre spänning. Med hjälp av integrerade spänningsregulatorer eller DC/DC-omvandlare samt övervakningsenheter kan man mata en grupp av funktionella enheter eller periferienheter med lämplig spänning.

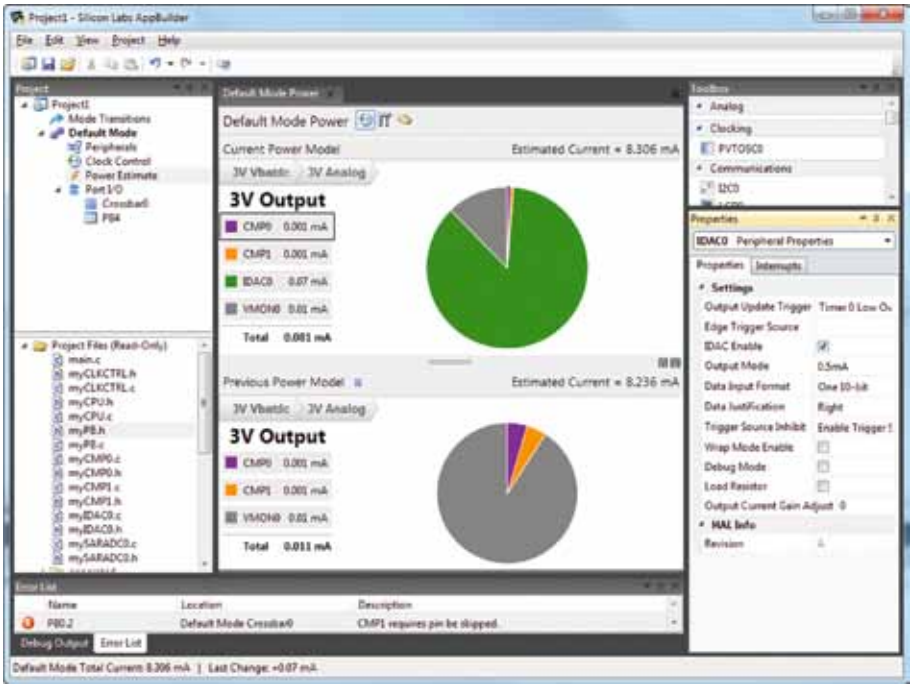
Inbyggda spänningsregulatorer ger mer flexibilitet och möjliggör uttag av mer laddning från ett batteri. En switchad buck-omvandlare – som den som integrerats i Silicon Labs MCU-familj SiM3L1xx – kan omvandla 3,6 V från ett industribatteri till 1,2 V med en verkningsgrad på över 80 procent. Många MCU:er har inte denna funktion utan använder mindre effektiva linjära komponenter för att sänka spänningen. I avancerade implementeringar kan buck-regulatorn stängas av när batteriet har nått en laddnivå då det inte längre är vettigt med omvandlingen.

Inbyggda applikationer är beroende av programvara som utnyttjar hårdvaran på bästa sätt. Vad som är lämpligt beror inte bara på tillämpningen, utan även på hårdvarans implementering. Likaså, ju mer flexibel CPU, klocka, spänning och minne, desto större energibesparingar kan man åstadkomma. Maskinvarumedvetna programvaruverktyg ger inbyggnadsingenjörer större insikt i vilka besparingar som kan göras.

ETT ALTERNATIV ÄR DYNAMISK spänningsskalning (se figur 2 och 3). Inbyggda DC/DC-omvandlare och prestandaövervakande kretsar användas för att minska matningsspänningen när tillämpningen inte behöver exekvera instruktioner vid högsta hastighet. Följande figurer visar de relativa skillnaderna mellan då ingen spänningsskalning används (VDD fixed), statisk spänningsskalning (SVS) och aktiv spänningsskalning (AVS).

AVS-strategin kan ändras beroende av inspänningen till systemet. I detta exempel, när inspänningen är 3,6 V, är det effektivast att mata den interna logiken och flashminnet från en intern DC/DC. I takt med att inspänningen sjunker är det mer effektivt att mata minnet direkt från inspänningen, eftersom den interna logiken kan arbeta vid lägre spänning än minnet. Silicon Labs nya SiM3L1xx är exempel på en MCU-familj med flexibel kraftarkitektur med sex separata och varierbara kraftdomäner som möjliggör dynamisk optimering.

CMOS-logik arbetar normalt långsammare vid lägre spänning. Om tillämpningen kan tolerera lägre prestanda (vilket ofta är fallet med kommunikationsprotokoll som kräver att data inte levereras snabbare än en viss standardiserad frekvens) kan en lägre spänning ge stor energibesparing. Läckaget sätter dock en lägsta gräns för spänningsskalningen. Om varje operation



Figur 4. Verktygsstöd gör att utvecklare kan optimera för låg effekt.

tar för lång tid kommer läckaget att dominera energiekvationen. Av den anledningen är det vettigt att exekvera en funktion så snabbt som möjligt och sedan sätta processorn i sovläge.

TA EN TRÅDLÖS GIVARTILLÄMPNING som måste utföra en väsentlig mängd signalbehandling, exempelvis en glaskrossdetektor. Här utnyttjas Fast Fourier Transform (FFT) för att analysera vibrationer. Det görs genom att en audiosensor plockar upp de frekvenser som karakteristiskt alstras när glas krossas. En FFT-analys är relativt komplicerad – att exekvera den vid låg frekvens kommer troligen att öka läckaget väsentligt även i äldre processer. Därför är det i detta fall bästa att exekvera vid nära maximal frekvens för att sedan återgå till sovläge tills resultatet ska rapporters till en värdnod.

Trådlösa protokollkoder ställer andra krav. Radioprotokoll har fast timing för händelser och protokollen kan hanteras helt i hårdvara. Det är bättre att minska processorkärnans spänning och köra den kod som behövs för paketering och överföring vid en hastighet som passar det trådlösa protokollet.

Hårdvarublock som intelligent direkt minnesaccess (DMA) kan ändra energiavvägningen ytterligare. Många DMA-block, som den som erbjuds hos den ursprungliga ARM Cortex-M3-processorn, kräver dock stöd från en processor. Mer intelligenta DMA-block tillåter processorn att beräkna packethuvuden, kryptera data och sätta samman paket för att sedan lämna över arbetet med att överföra paketen med lämpliga mellanrum till de minnesbuffertar som utnyttjas av radio-front-end. Under större delen av tiden som radiolänken är aktiv kan processorn sova, vilket sparar mycket energi.

Moderna 32-bitars MCU:er ger stor frihet när det gäller hur minnesblock utnyttjas. Normalt har de en blandning av flashminne och SRAM. Vanligtvis tar accessen av flashminnet mer effekt än accessen av SRAM:et. I normalfallet kommer flashminnet att läsas tre gånger så ofta som SRAM:et. Att skriva till ett flashminne drar ännu mer energi – men det är ovanligt att skriva till flashminnet så den genomsnittliga effektförbrukningen påverkas inte.

Flashminnets effektförbrukning beror också av hur accesserna från processorn distribueras. Varje flashminnesblock har flera sidor, om vardera upp till 4 kB, som kräver spänningsmatning. Sidor som inte används kan hållas i ett energisnålt tillstånd. Därför att det viktigt att utnyttja minnet effektivt. Det är ofta effektivt att lägga funktioner som används frekvent i SRAM:et.

ENERGIOPTIMERING KULLKASTAR även traditionella idéer om kodeffektivitet. Under flera årtionden har inbyggnadsingenjörer varit inriktade på att optimera kod for minnesstorlek (om inte prestandan är kritisk). Energioptimering ger en helt annan uppsättning mätvärden. En viktig avvägning hur det inbyggda cacheminnet utnyttjas.

Optimering av kodstorlek gör att mer kan behållas i cacheminnet, vilket förbättrar både hastighet och energiförbrukning. Funktionsanrop och grenar som används för att minska storleken på applikationen genom återanvändning av gemensam kod kan leda till oavsiktliga konflikter i cacheminnet. Kod som ska köras ofta bör vara tillräckligt kompakt för att få plats i cacheminnet, det gäller dock inte gren- eller anropsfunktioner.

Ta t ex ett brandlarm; även om det larmar en gång i veckan är det bara 520 händelser av 315 miljoner under larmets tioåriga längd. Av alla sensoravläsningar kommer färre än 0,0002 procent att resultera i exekveringen av larmgenererande kod. Återstående 99,9998 procent av kodexekveringen kommer att vara av den grundläggande sensoravläsande slingan. Att tillse att denna kod körs rakt ut från cacheminnet kan vara nyckeln till minimerad energianvändning, medan den återstående koden optimeras med hjälp av traditionella tekniker.

VERKTYGSSTÖD ÄR NÖDVÄNDIGT för att maximera energieffektiviteten hos en MCU-plattform. Förmågan att allokera funktioner till diskreta sidor av flashminnet kräver en länkare som förstår minnesstrukturen hos MCU:n. Länkaren kan ta utvecklarens information och generera en binärkod som är optimerad för den mest energieffektiva användningen av icke-flyktig lagring. I princip används också denna kod för att tillse att funktioner och data är placerade på sådant sätt att de som exekveras inte kolliderar. Denna detaljnivå kan lättast åstadkommas när verktygen tillhandahålls av MCU-tillverkaren, som ju känner till minnets layout och kraftbehoven för varje målplattform.

MCU-tillverkarna har också detaljerad förståelse för periferienheter och on-chip-bussar. Den kunskapen kan tillämpas i verktygen för att vägleda ingenjören att göra val som inte slösar energi. Den grafiska AppBuilder-miljön, utvecklad av Silicon Labs, är ett sådant exempel (se figur 4).

AppBuilder kan titta på periferienheternas inställning och avgöra om energibesparande förändringar är möjliga. Om exempelvis en användare har fört in en UART i applikationen och satt dess hastighet till 9 600 baud kommer verktyget att bestämma lämplig inställning. ARM:s periferibuss (APB), som exempelvis används för UART:er och AD-omvandlare, kan arbeta vid upp till 50 MHz. I detta fall är hastigheten mycket högre än nödvändigt, därför frågar verktyget om användaren vill minska datahastighet till en mer passande nivå.

AppBuilder-programvara ger även annan applikationsspecifik information rörande effektförbrukningen. Med hjälp av en simulering av MCU:n kan verktyget tillhandahålla ett interaktivt histogram för beräknad ström för hela applikationen liksom för processorn och alla periferienheter.

Utvecklingsverktyg kommer att bli mer "energimedvetna." Traditionellt har avlusningsfunktioner som brytpunkter satts för händelser, tex läsning och skrivning av ett minne. I framtiden är det möjligt att brytpunktsstöd kommer att utvecklas till att hantera energirelaterade frågor. ■