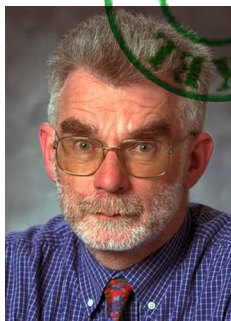


# ELEKTRONIK TIDNINGEN



**Peter Bishop**  
Informationschef  
Atmel Corporation

## Den perfekta kompromissen för volymer under asic

Styrkretsar, signalprocessorer, fpga:er och asicar har sina för- och nackdelar.

En kustomiserbar styrkrets – som Atmels CAP-familj – är en perfekt kompromiss vad gäller kostnad, prestanda och energi.

**Redaktör**  
Jan Tångring  
jan@etn.se  
0734-17 13 09

**EMBEDDED**  
EXPERT

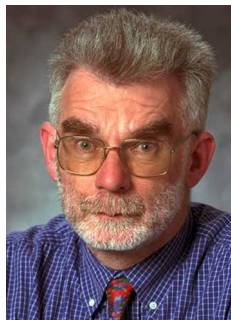
© Elektroniktidningen och Atmel, 16 oktober 2009

[www.etn.se/expert](http://www.etn.se/expert) – kostnadsfria tekniska rapporter

# Den perfekta kompromissen

En kustomiserbar styrkrets ger de extra prestanda du behöver när dsp-mcu-fpga-lösningen inte räcker, men volymen inte motiverar en asic.

Av Peter Bishop, Atmel Corporation



**Peter Bishop** är informationschef på Atmel (stationerad i franska Rousset) och som sådan ansvarig för mediakontakter, informationsmaterial, mässverksamhet, datablad, manualer, videomaterial, med mera, med särskild inriktning på asic, strömstyrning och ljudsyntes. Peter Bishop har varit lärare på universitetsnivå och är författare till ett antal läroböcker om elektronik och informationsteknik.

När du konstruerar inbyggda system av integrerade kretsar, är det åtminstone fyra faktorer som du måste minimera.

För det första antalet transistorer. De avgör krets- och kapselstorlek, pris per enhet och effektförbrukning. Process-tekniken krymper visserligen hela tiden transistorerna, men både statisk och dynamisk effektförbrukning beror av hur många de är.

För det andra antalet klockcykler. Färre ger lägre effektförbrukning.

För det tredje utvecklingstiden. Den avgör tillämpningens marknadsacceptans. En produkt som missar sitt marknadsfönster är bara ett slöseri med utvecklingsresurser. Ofta är det utvecklingen av programvara som tar längre tid och är dyrare än utvecklingen av maskinvara.

Den fjärde och sista utmaningen är NRE-kostnader (non-recurring engineering) som masktillverkning och utveckling av maskin- och programvara. NRE-kostnaderna har idag blivit så höga att många inte kan utnyttja de ledande processteknikerna.

Fyra olika tekniklösningar existerar för att möta de fyra utmaningarna.

Den första är **styrkretsar** (micro controller unit, mcu). De är en sorts generella verktyg för informationsbehandling och styrning som skräddarsys för sin specifika uppgift via programvara. Endast programutveckling och validering krävs för att skapa en tillämpning, och NRE-kostnaderna kan amorteras ut på alla

som använder samma styrkrets. Klockcykeloptimering i styrkretssammanhang är detsamma som kodoptimering, och kodens storlek och effektivitet påverkar bland annat hur många transistorer som måste offras på minnesceller. Kompakt kod och effektivt arkitekturutnyttjande är ett måste. Styrkretsar är i allmänhet duktiga på att använda sina transistorer och klockcykler effektivt, men de är inte helt optimala.

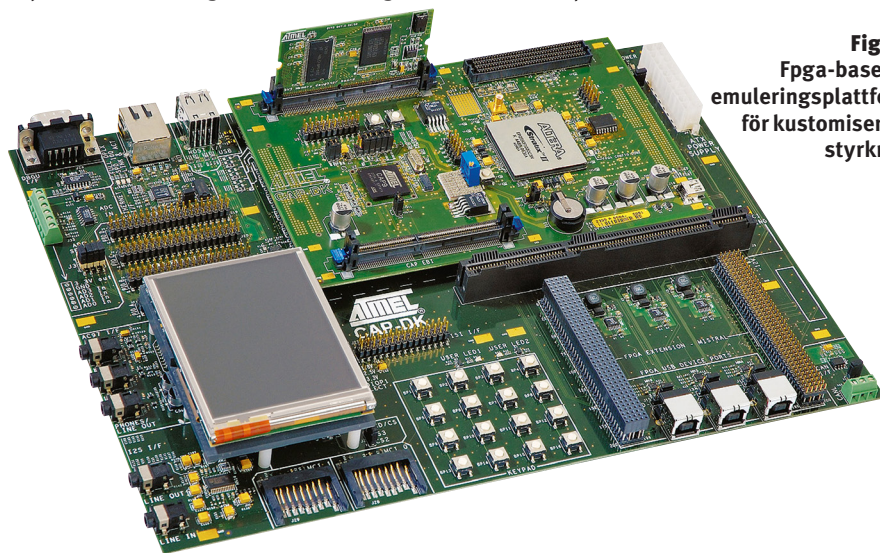
Tekniklösning nummer två är **signalprocessorer** (digital signal processor, dsp). De har hårda, snabba implementationer av signalalgoritmernas grundoperationer, vilket minimerar antalet transistorer och klockcykler, på viss bekostnad av flexibilitet. Koden är enklare än styrkretskod. En signalkrets är troligen

det optimala för vissa av tillämpningens funktioner, men oftast inte för alla.

Det finns även styrkretsar med dsp-instruktioner som gör att du slipper kalla in en dedicerad dsp för vissa enklare signalbehandlingsuppgifter.

Den tredje teknologin är **fältprogrammerbara grundmatriser** (Field-programmable gate array, fpga). Här begränsas utvecklingsinsatsen till den kodning som krävs för att konfigurera dem, och du delar NRE-kostnaderna med en stor mängd användare, till priset av ett stort transistoröverskott – och därmed höga kostnader per enhet – och begränsad klockcykeloptimering. Effektförbrukningen blir långt ifrån optimal.

Nämnda tre teknologier levereras som standardprodukter. De finns ofta i mäng-



**Figur 1**  
Fpga-baserad emuleringsplattform för kustomiserbar styrkrets

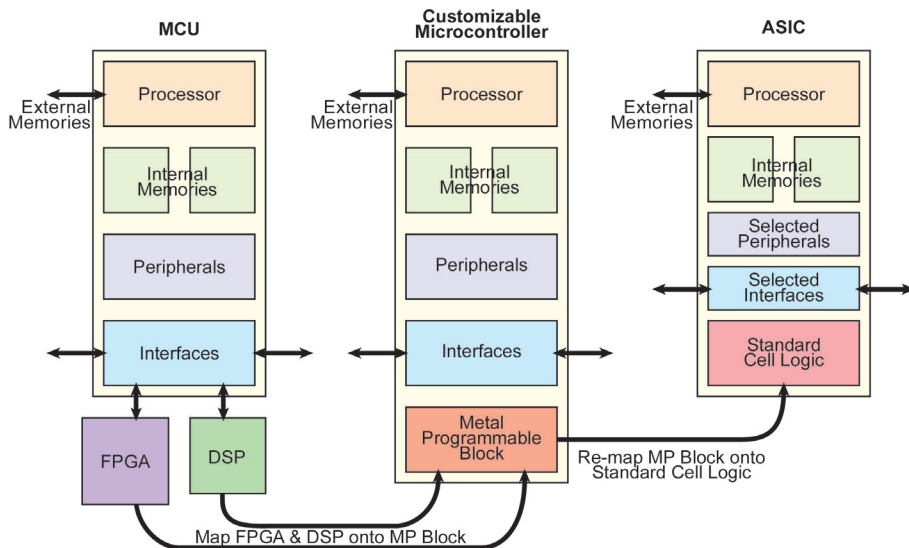
der av olika utföranden, men du får alltid några transistorer och in- och utgångar över, hur mycket du än lusläser databladet för att hitta den bäst lämpade. Tillämpningsspecifika funktioner, särskilt analoga beräkningar, måste oftast implementeras utanför kretsen.

Kretsstorlek, kapselstorlek, pinout och effektförbrukning är långt ifrån optimal jämfört med vad som kan åstadkommas med den fjärde teknologin, som är **tillämpningsspecifika integrerade kretsar** (application specific integrated circuit, asic). En asic är specialkonstruerad för en viss tillämpning, med potentiellt en eller flera styr- eller signalkärnor och så mycket som möjligt av systemfunktionerna integrerade. Asicar ger dig det minsta antalet transistorer och klockcykler, en minimal kostnad per enhet och en minimal effektförbrukning. Men på bekostnad av utvecklingstid och NRE-kostnader — typiskt en magnitud högre än mcu:er, dsp:er och fpga:er.

**De fyra teknologierna** kompromissar var och en på sitt sätt med de fyra optimeringskraven. Ditt val blir därmed alltid en kompromiss. Det som faller avgörande är en oftast komplex kombination av faktorer — ingen enda av teknikerna är alltid ideal, och olika blandningar av teknologier är lämpliga vid olika stadier av slutprodukten. För prototypframställning och produktionsupprampning kan en fpga, eventuellt kombinerad med en mcu eller dsp, vara det som bäst minskar utvecklingstid och -kostnad. När produkten börjar göras i stora volymer kan funktionaliteten mappas om till en asic som integrerar mcu- eller dsp-kärnan och absorberar logiken från fpga:n. Då optimerar du kretsstorlek, kostnad per enhet, klockcykler och effektförbrukning utan att behöva skriva om programvara. De höga NRE-kostnaderna för asicutveckling amorteras över den stora tillverkningsvolymen.

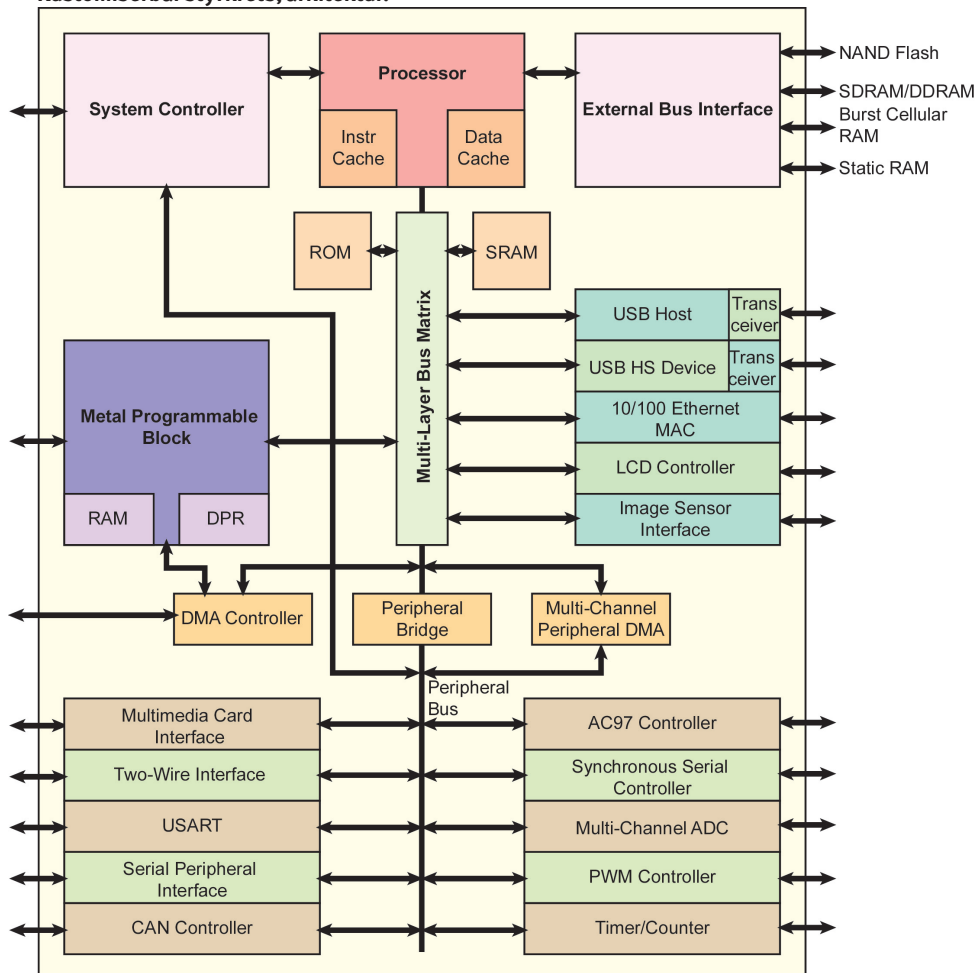
Det finns en alternativ teknik som utnyttjar styrkan hos mcu, dsp och fpga men dessutom tar ett steg i riktning mot en asic, nämligen en **kustomiserbar styrkrets** (customizable microcontroller). En sådan har en fast del bestående av en mcu — inklusive processor, minnen, kringutrustning och gränssnitt — och ett metallprogrammerbart block (MP-block) som kan implementera en dsp eller en extra mcu med tillämpningsspecifik logik.

Låt oss som exempel ta en typisk tillämpning där en mcu styr övergripande systemfunktioner, nätverk, datahantering och användargränssnitt medan en dsp sköter signalbehandling och en fpga sköter tillämpningsspecifik logik. I första utvecklingsstadiet kan du använda en trekretslösning där programmeringen av



**Figur 2**  
En kustomiserbar styrkrets är ett mellanting mellan en mcu/fpga/dsp-lösning och en asic.

**Figur 3**  
Kustomiserbar styrkrets, arkitektur.



mcu, dsp och fpga står för den största utvecklingsinsatsen. Du får inga NRE-kostnader för maskinvara, men enhetskostnaderna blir ganska höga, i synnerhet för fpga:n. Systemprestandan blir suboptimal på grund av dataöverföringen mellan mcu, dsp och fpga — den sistnämnda ligger oftast under de tidigare i klock-

frekvens. Effektförbrukningen är också ganska hög, med fpga:n som den största syndaren. Och kretsar och förbindningar tar mycket plats på kortet. Implementeringen är ideal för prototyper och vid upprampning till volymtillverkning, men suboptimal för volymtillverkning.

Denna trekretskonfiguration kan om-

vandlas till en kustomiserbar styrkrets (Figur 3) med minimal omskrivning av mcu- och dsp-programvara. I och med att den kustomiserbara styrkretsen använder en standardkärna är den troligen kodkompatibel med standard-mcu:n. Därmed kan programvaran skrivas om med en begränsad insats. För att mappa dsp:n på den kustomiserbara styrkretsens MP-block använder du dsp-arkitekturens HDL-beskrivning. Fpga-logiken mappas på MP-blocket med samma verktyg som användes för att utveckla det. Bortsett från den ökade klockhastigheten är logiken i fpga:n och MP-blocket identisk.

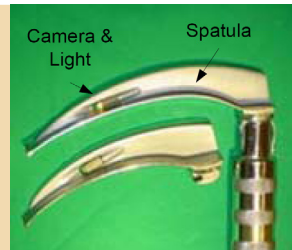
Dsp- och fpga-implementeringarna i MP-blocket kan optimeras med hjälp av ett antal RAM- och Dual Port RAM (DPRAM)-block som ligger utspridda i MP-blocket. Dessutom kan man gå vid sidan av processorn och använda dels DMA-styrkretsen och dels parallella portar som länkar MP-blocket till en snabb flerskiktad bussmatris, för att överföra data mellan MP-block, interna och externa minnen, kringutrustning och gränssnitt. På det viset spar du klockcykler när flera tillämpningar samtidigt behöver dataöverföring och -behandling.

Utöver mindre kortyta och sänkta materialkostnader ger den kustomiserbara styrkretsen fördelar i alla de fyra faktorer som omnämndes inledningsvis. Du använder betydligt färre transistorer, främst för att fpga:n ersätts med MP-blocket. Du använder färre klockcykler

**E**tt mediskt instrument, ett *videolaryngoskop*, kan tjäna som illustration av hur en kustomiserbar styrkrets kan användas.

Konventionella laryngoskop exponerar och belyser struphuvudet in till stämbanden. Moderna versioner har inbyggd video. En styrkrets buffrar och överför okomprimerad video via bredbandig korthållsradio – som UWB (Ultra Wideband) – till en basstation med högupplöst skärm och sekundärminne.

Elektronik, antenn och batteri – minst en timme – måste rymmas i laryngoskopets



Laryngoskop med inbyggd video och UWB-länk.

handtag.

En kustomiserbar styrkrets visade sig ha vad som krävdes. Processorn programmerades för styrfunktionerna och de metallprogrammerbara blocken för tillämpningsspecifik logik och UWB-gränssnitt.

Klockfrekvens och parallella interna datavägar gav en mer än tillräcklig bandbredd. Utrustningen fick plats och effektförbrukningen uppfyllde batteritiden. Tillverkningsvolymen på 10 000 till 100 000 enheter per år var tillräcklig för att amortera ut utvecklingskostnaden och hålla kostnaden per enhet rimlig.

för en given funktion, särskilt om DMA-arkitekturen hos den kustomiserbara styrkretsen används fullt ut. Du håller implementationstiden till ett minimum genom att du återanvänder kod från mcu/fpga/dsp-lösningen i den kustomiserbara styrkretsen. Ännu effektivare blir det om du kan utnyttja de fpga-baserade emulatorer som ofta levereras med kustomiserbara styrkretsar (Figur 1).

NRE-kostnaderna begränsar sig till placering-och-ledningsdragning för MP-blocket och dess metallmasker.

Kustomiserbara styrkretsar är optimala för medelstora och stora tillverkningsvolymen. Vid ännu större volymer kan du göra ytterligare en optimering

– kostnadseffektivt – genom att mappa den kustomiserbara styrkretsen på en standardcells-asic och samtidigt passa på att eliminera outnyttjad kringutrustning, gränssnitt, minnesblock och logik i MP-blocket. Du använder den HDL-kod du redan utvecklat för den kustomiserbara styrkretsen, med vissa minimala justeringar. Efter uttömmande simuleringar – som verifierar ekvivalensen mellan de två versionerna – gör du placering-och-ledningsdragning och maskframställning för asicen. Maskkostnaderna är visserligen inte oväsentliga, men amorteras över den stora volymen. ■