

FPGA'en bruges som lup

I takt med, at priserne på FPGA'er er faldende, bliver anvendelsesområderne stadig bredere. Her beskrives en applikation, hvor det måske ikke umiddelbart virker nærliggende at anvende en FPGA, men hvor den programmerbare komponent alligevel viser sig at være oplagt at bruge.

Af Anders Enggaard, Axcon - The FPGA Power House (www.axcon.dk)

I en årrække har anvendelse af FPGA været naturligt i større og komplekse systemer, hvor prisen på FPGA-komponenten ikke har været afgørende for systemets samlede pris. Mindre systemer med højere følsomhed for komponentpriser har derimod traditionelt været afskåret fra at lade komplekse FPGA komponenter indgå i designet. Den seneste udvikling i pris/ydelsesforholdet i FPGA-verdenen har åbnet for nye anvendelser, og det er nu inden for rækkevidde at integrere ret komplekse System-on-Chip (SoC) løsninger i selv billige FPGA-komponenter.

Axcon har leveret SoC løsninger implementeret i FPGA'er i adskillige kundeprojekter. I det følgende fokuseres på en anvendelse, hvor det måske ikke umiddelbart virker nærliggende at bruge en FPGA, men hvor denne teknologi alligevel viser sig at byde på en række fordele i sammenligning med alternative løsninger.

Systemfunktionen

Den overordnede opgave har ligget i at udvikle en *Digital Video Processor* løsning, som



FPGA'en kan anvendes i en lang række sammenhænge - også i applikationer, hvor det måske umiddelbart ikke virker oplagt at bruge en sådan komponent, forklarer Anders Enggaard fra udviklingshuset Axcon, der har specialiseret sig inden for design af systemer omkring bl.a. FPGA'er.

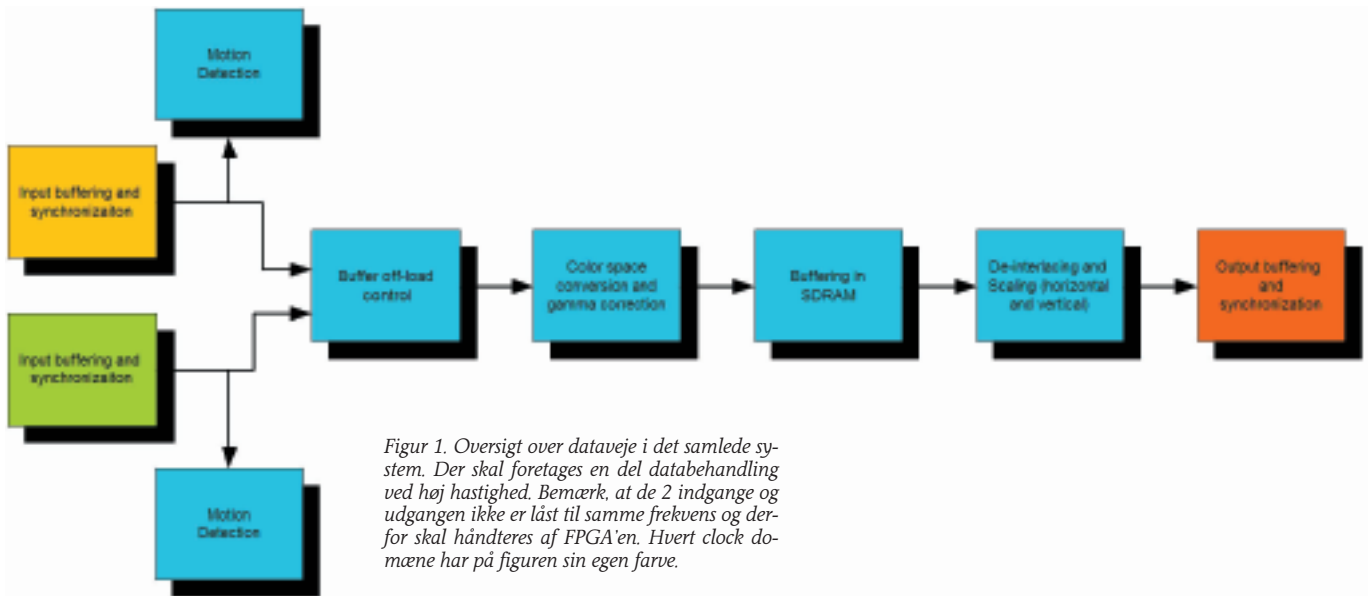
gør det muligt at vise videosignaler fra to PAL kameraer på en pc-monitor sammen med det oprindelige pc-billede.

Da systemet behandler livebilleder, er der store krav til opdateringshastigheden, og forsinkelserne i systemet skal være små. Det endelige system tilbyder ekstrem forstørrelse, og deraf følger store bevægelser i billederne. Bevægelserne skal detekteres af systemet og billederne skal gengives uden synlig kvalitetsforringelse.

Er det en oplagt FPGA-applikation?

Set ovenfra virker opgaven forholdsvis simpel, og umiddelbart virker det ikke som et system, hvori der i større omgang indgår programmerbar logik. Brydes systemet ned i funktionsblokke, så bliver det imidlertid klart, at der er et behov for en del funktionalitet, som kan løses effektivt ved hjælp af en FPGA.

...FORTSÆTTES SIDE 66



Figur 1. Oversigt over dataveje i det samlede system. Der skal foretages en del databehandling ved høj hastighed. Bemærk, at de 2 indgange og udgangen ikke er låst til samme frekvens og derfor skal håndteres af FPGA'en. Hvert clock domæne har på figuren sin egen farve.

FORTSAT FRA SIDE 64:

Som det fremgår af figur 1, er der i systemet et behov for forskellige typer af datahåndtering så som buffering, synkronisering, lagring og beregninger.

Buffering og synkronisering

Systemer vil ofte have flere grænseflader og i mange tilfælde også forskellige clock kilder. Håndtering af de forskellige clock domæner bør på et tidligt tidspunkt medtages i arkitekturdesignet på en måde, som gør designet så rent og simpelt som muligt. I dette design indgår fire forskellige clock domæner. Synkroniseringen er foretaget i forbindelse med buffering, både i indgange og udgangen på systemet. Herved bliver langt størstedelen af FPGA-funktionaliteten lagt i ét og samme clock domæne, hvilket simplificerer designet og timing analysen.

Buffering i indgangen tillader, at datavejen gennem color space conversion og gamma correction frem til SDRAM kan deles mellem de to kameraer. Dette er naturligvis betinget af, at båndbredden af datavejen frem til SDRAM er mere end dobbelt så høj som båndbredden fra kameraerne (36 MByte/s hver).

Lagring

Billedfrekvenserne for kameraerne og pc skærmen ikke er ens. Den simple FIFO buffering er derfor ikke tilstrækkelig, da pc

skærmen typisk vil have større billedfrekvens end kameraerne, og derfor vil bufferne tømmes hurtigere end billederne kommer ind. Som følge heraf bliver et lager nødvendigt. I lageret gemmes så et antal halvbilleder fra kameraerne.

Det endelige billede skal vises på flere typer af pc skærme med forskellige opdateringshastigheder og opløsninger (XGA, SVGA, VGA). Outputkravet til båndbredden bliver derved tæt på 200 MByte/s.

Båndbredekravet for SDRAM interfacet bliver derved 72 MByte/s i skriveretningen og 200 MByte/s i læseretningen. Frem for at anvende en dyr multiport DRAM, indgår der i systemet en SDRAM-controller, som er optimeret til behovet for henholdsvis skrive- og læse-båndbredde og som anvender single-port SDRAM.

I forbindelse med systemdesignet blev det overvejet at anvende en færdig SDRAM-controller IP blok. Det viste sig, at et eget SDRAM controller design ville give øget båndbredde, eftersom designet kunne tilpasses det øvrige system 100 procent. Den tekniske løsning indeholder bl.a. optimeret placering af pre-

charge, refresh og arbitrering. Resultatet bliver, at SDRAM banks holdes åbne i længere tid og dermed følger højere båndbredde.

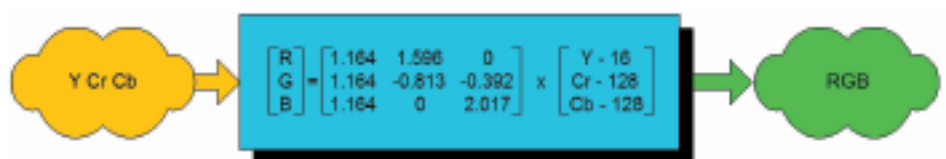
Beregninger i FPGA

Fra kameraerne kommer billedinformationen i 24 bit YCrCb format (YUV), som skal omsættes til RGB før billederne kan præsenteres på en pc skærm – dette kaldes color space conversion (figur 2).

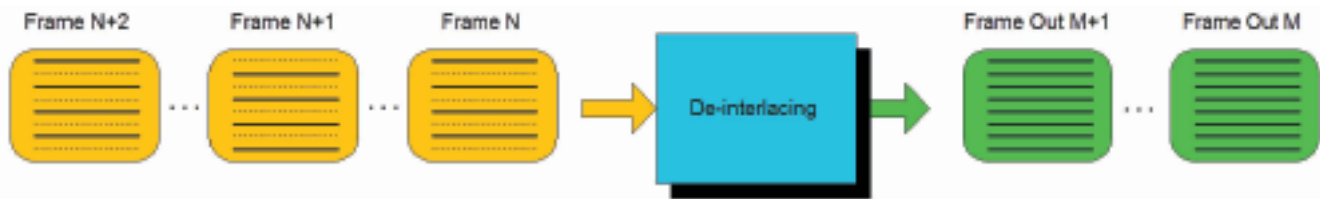
Denne beregning udføres af et dedikeret kredsløb, hvori beregningerne paralleliseres for at sikre tilstrækkelig båndbredde. De aritmetiske operationer er implementeret direkte i FPGA'ens logiske blokke. Eftersom beregningsformlerne er faste, kan disse udføres simpelt og langt hurtigere i hardware end i instruktioner afviklet på en processor.

Gamma korrektion

Baggrunden for, at der skal udføres en gamma korrektion, vil ikke blive beskrevet her, men overordnet er der tale om en overførings-



Figur 2. Color space conversion er paralleliseret, således at de 7 multiplikationer, 7 additioner og 3 subtraktioner udføres for hver clockperiode på 10ns.



Figur 3. Fra kameraerne kommer skiftevis lige og ulige halvbilleder, som sammensættes til helbilleder, før disse kan vises på pc skærmen i non-interlaced format. Tidsforskydning mellem halvbillederne håndteres af algoritme i FPGA.

funktion, som påtrykkes de indkomne videosignaler for at opnå mere nøjagtig billedgengivelse. Korrektionen kan simpelt udføres af FPGA'en.

De-interlacing og intelligent real time billedkorrektion

Billederne kommer fra kameraerne i interlaced form og skal de-interlaces, inden billederne præsenteres. Den simple metode til de-interlacing er blot at lægge de forrige billede sammen med det sidst ankomne (figur 3). Eftersom kameraet leverer skiftevis lige

og ulige halvbilleder, vil hver anden linie være tidlig gammel.

Sammensmeltningen af de to halvbilleder, som har forskellige optagelsestidspunkter, vil resultere i et efterslæb i hver anden linie, hvis der ikke foretages en mere intelligent behandling af halvbillederne, der kommer ind. Denne problematik er håndteret af en speciel algoritme, som skifter karakter afhængig af den indkommende billedinformation (figur 4 på næste side).

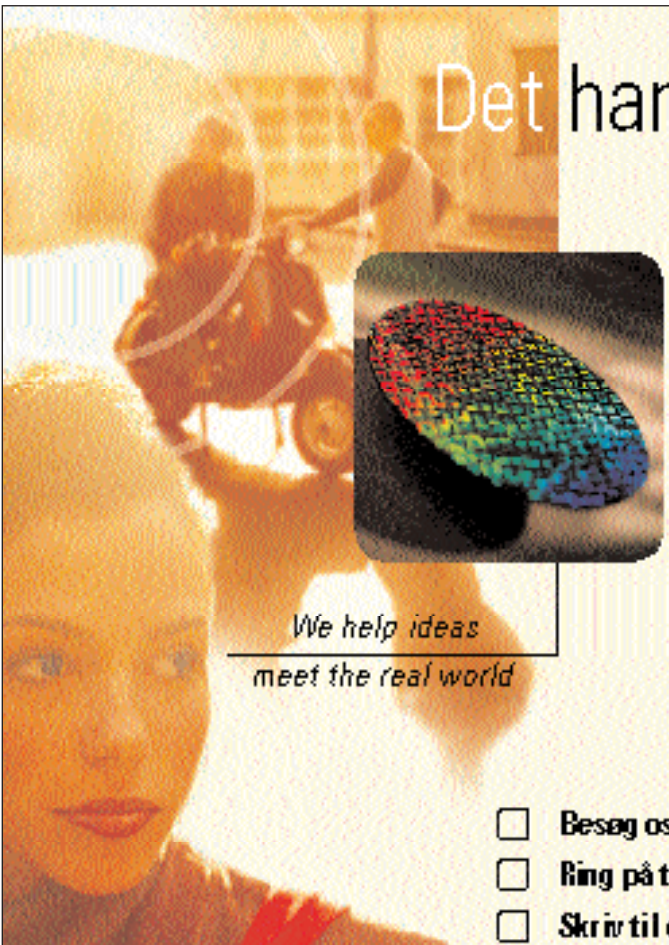
Med andre ord vil bevægelser i billedet betyde ændringer i de-interlacing algoritmen. Bevægelse registreres realtime ved at sam-

menligne tidligere modtagne billeder med det aktuelle halvbillede.

FPGA'en foretager altså både detektering af bevægelse og kompensering herfor. Som det fremgår kræver mekanismen, som håndterer de-interlacing og kompensation af bevægelser, at flere halvbilleder fra hvert af de to kameraer lagres. Igen viser der sig ved nærmere undersøgelse af denne funktion, at der følger en større kompleksitet, end hvad der umiddelbart forventes.

...Fortsættes næste side

Det handler om ASIC




- Kort udviklingstid
- Low power
- Høj pålidelighed
- Lav kost

Vi kan også skræddersy din næste chip, så du kan sige farvel til dit lager af standardkomponenter og godkald til din helt egen unikke chip, der er perfektioneret til at møde dine personlige krav.

Mød os på **Electronica 2004** den 9.-12. november i München, hal A5, stand A5.279.

We help ideas meet the real world

- Besøg os på www.delta.dk
- Ring på tlf 72 19 40 00
- Skriv til delta@delta.dk



FORTSAT FRA SIDE 67:

Skalering

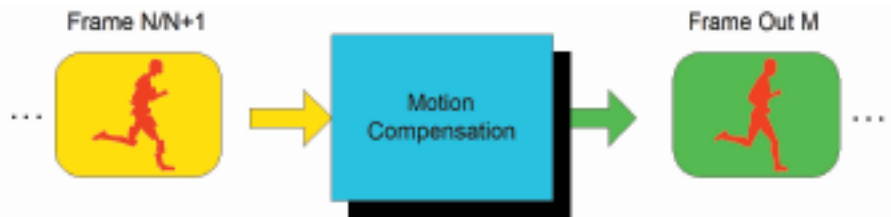
Hvert af de to kameraer vil ofte have en anden opløsning end den skærm, som resultatet skal vises på, og systemet skal derfor kunne foretage skalering. Hver enkelt pixel i det resulterende billede bliver udregnet af FPGA'en med en frekvens i omegnen af 100MHz (output pixelfrekvens), alt afhængig af opløsning.

Derfor er FPGA det rigtige valg

Som det tydeligt fremgår af ovenstående, er kompleksiteten af det på overfladen simple system tilstrækkelig stort til, at simple PLD-løsninger ikke er effektive. Havde løsningen ikke omfattet en FPGA, ville en stor del af databehandlingen være blevet pålagt en processor. En processorløsning, som kan levere samme datahastighed som FPGA-løs-



Systemets basale funktionalitet gør det muligt at vise billeder fra 2 PAL kameraer på skærmen samtidigt med et pc billede.



Figur 4. Bevægelser i billederne registreres og en special algoritme korrigerer billedet afhængig af de registrerede bevægelser. Uden kompensation for bevægelse vil billedet have udtrækninger i hver anden linie og virke uskarpt. Kompensation forbedrer billedkvaliteten markant. Såvel detektering som kompensation foretages real time.

ningen i denne applikation, vil ikke være konkurrencedygtig på hverken pris, størrelse eller effektforbrug. For år tilbage havde en FPGA-løsning været bekostelig, men nu er pris/ydelsesforholdet blevet gunstigt i systemer af denne karakter.

Valget af en FPGA-løsning åbner endvidere mulighed for at løse designopgaver som den omtalte med en høj grad af fleksibilitet. I applikationen er en del af de anvendte algoritmer blevet udviklet i samarbejde med testbrugere, som med løbende bedømmelse af resultaterne har haft stor indflydelse på algoritmeudviklingen.

Algoritmerne er blevet opdateret løbende gennem projektet. Valget af FPGA muliggør, at vi i Axcon har mulighed for at arbejde mere dynamisk med kunden. I løbet af projektet kan det således lade sig gøre at præsentere flere løsninger til en konkret problemstilling, som herefter er blevet underkastet en sammenlignende A/B-test.

Fleksibiliteten tæller

Den oplagte fordel ved anvendelse af FPGA er naturligvis den grundlæggende fleksibilitet – og det gælder ikke alene denne applikation, men også mange andre.

Desuden vil det ofte være muligt at flytte funktioner fra printet og ind i FPGA'en, hvor funktionerne på simple vis kan ændres og tilpasses.

Printudvikling og afprøvning er oftest en parallel aktivitet til FPGA-udviklingen, og FPGA-designet vil typisk indeholde funktionalitet til test af print og af komponenter monteret på printet. Senere i produktcyklus bliver testfunktionerne i FPGA'en opdateret til at indeholde logik, der muliggør og simplificerer produktionstest. På et tidligt tidspunkt kan f.eks. dataveje implementeres i forsimplet udgave for at muliggøre test af

øvrige funktioner på printet og åbne for systemtest, alt imens FPGA-udviklingen færdiggøres.

Sidst men ikke mindst øger fleksibiliteten sammen med modulopbygning mulighederne for genanvendelse af byggeblokke i umodificeret eller modificeret form til næste generation af produktet.

Achilleshælen

Naturligvis giver FPGA-verdenens fortræffeligheder også et nyt sæt af udfordringer. Med fokus på funktionalitet og de mange muligheder, som åbner sig ved anvendelse af FPGA, bliver emner, som er essentielle for slutproduktets kvalitet, ofte skubbet i baggrunden.

Stigende udbredelse af komplekse FPGA'er i systemer, hvor mere simpel programmerbar logik tidligere har løst opgaven, åbner for en glidende overgang til mere kompleks anvendelse, da mulighederne bliver mere åbne. Discipliner, som stammer fra ASIC-verdenen, bliver hurtigt mere aktuelle for moderne FPGA-design.

Respekt, forståelse og håndtering af discipliner som statisk timing analyse, floor planning, clock domæner, clock træer og verifikation er essentiel for designets endelige kvalitet. I Axcon ser vi desværre ofte designs, som fra start udelukkende har haft fokus på front-end designet. Resultatet kan blive, at der på et sent tidspunkt i projektføreløbet bliver kæmpet med at finde fejl, som burde have været undgået. Her skal det bemærkes, at fejl kan være funktionelle, såvel som at størrelses- og timingkravene ikke er mødt.

Pointen er, at godt forarbejde kan redde en fra mange ubehagelige overraskelser på et sent og som regel også kritisk tidspunkt.