Spotlight

EDI022 - Digitala Projekt SK

Fredrik Svensson
Hans Månsson

Handledare: Bertil Lindvall

LUNDS UNIVERSITET
Lunds Tekniska Högskola

Abstract

A CCD camera deliver raw composite signal. We extract information from this signal that can be used by the system to move the camera around. The task could be to create a panorama photo, chase movements in a picture or just follow the brightest point. In this project we have chosen to do the last thing, follow a light spot.

To be able to process the composite video signal it needs to be stored so the processor can process it. Due to the low frequency of the CPU a extern image capture part is needed. This part is created with a lattice, LM1881N, AD converter and some address-counters. Where the lattice controls when an image should be taken and when the processor should process the picture stored in the SRAM. The processor used in this project is the Motorola 68008.

Innehållsföteckning

Specifikationer
Systemet skall klara av att följa en ljuspunkt med en uppdateringsfrekvens på minst 1Hz. Upplösningen för bildbehandlingen skall vara tillräcklig för att indikera om ljusaste punkten ligger till höger eller vänster i bilden.

Lösningsförslag
Lösningsförslaget som arbetats efter bygger på att dela upp bildinläsning och bildbehandling i två separata processer. Bildinläsningen körs externt med adressräknare och Lattice styrd klockning då processorn inte hinner med. Bildbehandlingen ligger som en avbrottsrutin i processorn, det är även här servot ställs in för att centrera den ljusaste punkten.

Komponenter
Komponenter valdes för att tillgodogöra de krav som ställs på hastighet, lagringskapacitet, tri-state möjligheter o.v.s. Under projektets gång har komponenter både tillkommigt och frånfalligt.

Digital del
Alla komponenter i denna del kan gå i tri-state för att undvika kollisioner på data- och adressbuss. Tri-state innebär att in- och utgångar på komponenten blir högimpediva.

Processor
Motorola 68008 valdes som processor då den i stora drag arbetar likt en modern processor. God dokumentation och genuin kunskap hos vår handledare lät lovande. 68008 är en asynkron processor med 8 bitars arkitektur d.v.s. 8 bitars databuss. Processorn kan adressera 1 MB vilket betyder att den har 20 bitars adressbuss. 68008 har stöd för både DMA och avbrott. DMA betyder att processorn kan lämna över data- och adressbuss till en annan komponent samtidigt som processorn går i tri-state.

Systemklocka
Processorn behöver en klocka, till detta ändamål valdes den färdiga kristalloscillatorn EXO-3 med grundfrekvensen 20 MHz. Denna frekvens delades ner till 10 MHz, max för processorn.

Minne
Ett EPROM behövs för att lagra processor programmet. Eftersom EPROM inte går att skriva till behöves ett minne som både kan läsas och skrivas till ett så kallat SRAM.

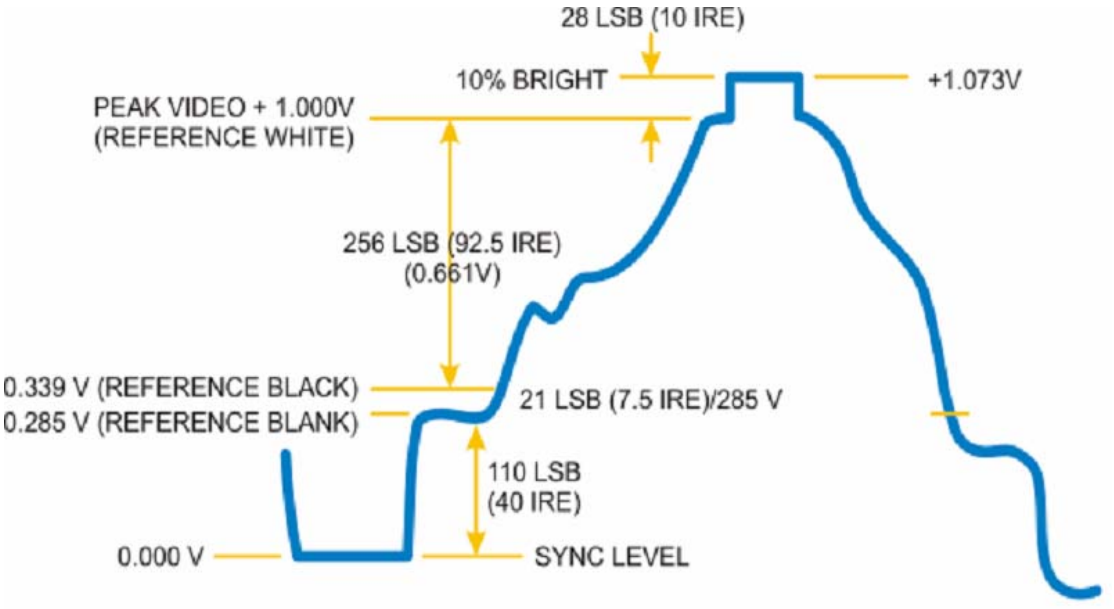
- NM27C010 1,048,576-Bit (128K x 8) High Performance CMOS EPROM
- IDT7MP4045 512 x 8 Parallell CMOS SRAM

Parallelport
För att processorn ska kunna styra servot som reglerar kamerans position används en buffert. Denna har en Byte allokerat i SRAM, beroende på värdet som är sparat ska servot rotera medurs, moturs eller vara stilla.

- SN74HC373 Octal transparent D-type latches with 3-state outputs

Analog del

Kamera
Kameran som valdes var en TVCCD-30MP. CCD kameras utsignal är kompositvideosignal enligt 625 linjers CCIR standarden. Upplösningen är 512(H)×582(V) med en uppdateringsfrekvens på 50 Hz. I figur 1 syns hur en typisk komposit signal kan se ut.



Figur 1: *Detta är en typisk komposit signal.*

Video Synk Separator
LM1881 är en video synk separator som kan plocka ut både H-Synk och V-Synk. Dessa signaler används sedan för att synka bildinläsningsprocessen.

Samplingsklocka
Med en H-synk frekvens på 15.625kHz valdes en samplingsfrekvens på 8MHz. Själva samplingen styrs via Latticen som drivs med en extern klocka, en EXO-3 med grundfrekvens 16 MHz.

A/D omvandlare
Vid val av A/D omvandlare var dess hastighet en av de viktigaste parametrarna eftersom den var tvungen att sampla med 8 MHz. TDA8703 (8-Bit High Speed A/D Converter) passar in perfekt eftersom den både är snabb och är 8-Bitars.

Räknare
Varje sampel A/D omvandlaren skapar ska sparas i minnet (SRAM). Detta går fortare än vad processorn klarar av så en extern adressräknare är ett måste. Som adressräknare valdes tre stycken SN74HC590A (8-bit binary counters with 3-state output registers) som sedan kaskadkopplades. Klockningen av den minst signifikanta räknarn sker, liksom AD-omvandlaren, via Latticen.

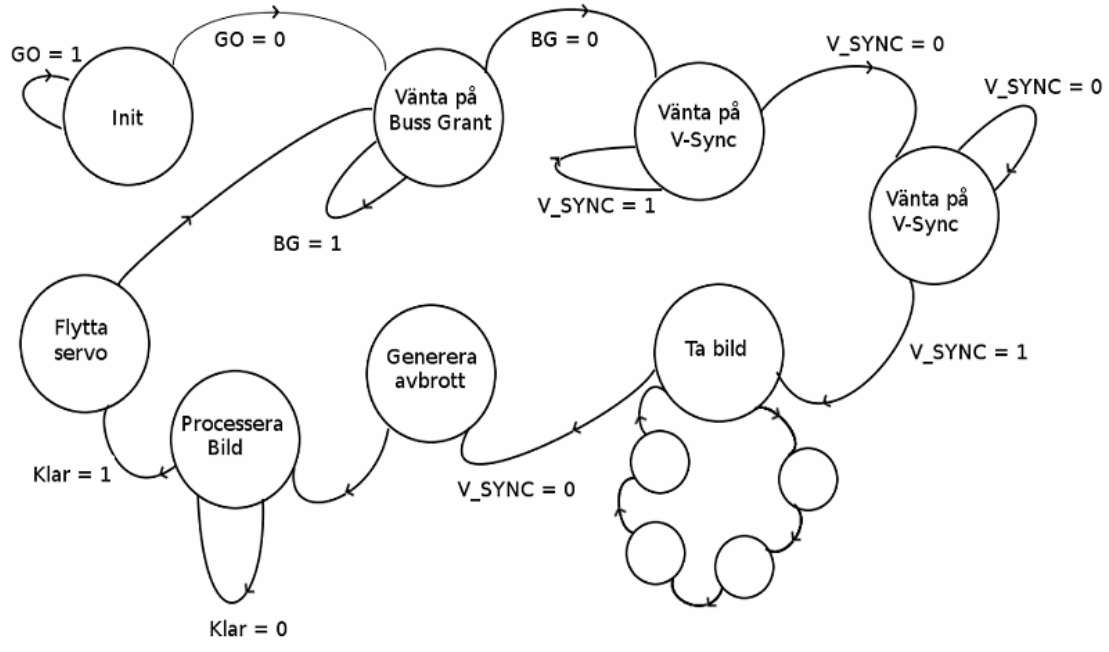
Logik
All logik sköts av en programmerbar logik krets, en ispLSI 1032E. Denna används för att generera chip-select, läs/skriv, avbrott och DMA signaler för periferienheter.

Chip-Select
För att kunna läsa och skriva till flera enheter, tex EPROM och SRAM, måste dessa placeras ut i minnesrymden. För placering används de högsta adressbenen. I tabell 1 visas hur allokeringen definieras. När en adress anropas stänger Latticen av ickeberörda enheter och ser till att aktivera den enheten adressen tillhör.

A19	A18	A17	Komponent
0	0	X	EPROM
0	1	0	Parallellport
0	1	1	Avbrott
1	X	X	SRAM

Tabell 1: *Adressrymdsallokering, här indikerar X ”don’t care”.*

Tillståndsmaskin
Då en bild skall läsas in från kameran måste CPUn stängas av då denna är för långsam. Detta kallas att den går i DMA mode och dess data och adressben blir högimpediva. Nästa steg är att vänta in en V-synk från synk separatorn och sedan klocka både AD omvandlare och de externa adressräknarna med 8MHz. När både samplet och adress är stabil skrivs data in i SRAMet. Detta fortsätter tills nästa V-synk signal kommer. Lite enkelt visat i figur 2.



Figur 2. *En schematisk tillståndsgraf*

Första tillståndet initierar hela systemet. Med det menas att ekvationer för adressrymden och DTACK signaler sätts upp. Här väntar systemet tills en starknapp trycks ned. Tillstånd 2 ber processorn gå i DMA mode, detta betyder att processorns adress- och databuss går i tristate så att andra enheter kan kommunicera över

bussarna. När detta skett lyssnar Latticen på V-synk signalen från LM1881 kretsen, när denna signal går låg och i tillståndet efter går hög påbörjas inläsningen högst upp från vänster på en ny bild. Själva inläsningen sker i fyra olika tillstånd som loopar tills att en ny V-synk signal kommer. Vad som händer i dessa *Ta bild* tillstånd är att adressräknarna klockas fram ett steg samt att AD omvandlaren ombedes att sampla och hålla detta värde. Andra tillståndet är till för att vänta så att data på bussarna är stabila. Nästa tillstånd ber SRAM att skriva in data på databussen på platsen där adressräknarna pekar. Det fjärde tillståndet väntar så att detta säkert skett innan nästa cykel påbörjas.
När nu nästa V-synk signal kommer lämnar Latticen tillbaka bussarna till processorn och genererar ett avbrott. Avbrottsrutinen i processorn tar hand om bildprocesseringen för att sedan kunna avgöra om kameran måste roteras för att centrera ljusaste punkten.

Avbrott

Processorn har tre avbrottsnivåer 2, 3 och 7 varvid 7 har högst prioritet. Då ett avbrott genereras används två insignaler till processorn, IPL0/2 och IPL1. Dessa indikerar vilken avbrottsnivå som gäller. Vidare finns det två olika typer av avbrott, vektoriserat och autovektoravbrott. Vektoriserat avbrott innebär att periferienheter kan leverera en adress till avbrottsrutinen. Är inte detta möjligt används autovektoravbrott där mjukvara fördefinierat avbrottsadressen. För att indikera vilken av dessa typer av avbrott som skall användas sätts VPA hög respektive låg.
Benen FC0, FC1 och FC2 indikerar vilket tillstånd processorn befinner sig i, är samtliga höga är processorn redo att ta mot ett avbrott.
Då bildinläsningen är klar genereras ett avbrott, i detta fall ett autovektoravbrott på nivå 2. Valet av nivå spelar i detta projektet ingen roll då det endast finns ett tillfälle då avbrott kan genereras.

Metod

Bakgrund

Eftersom kameran som valts levererade kompositvideo signal var första steget att ta reda på hur denna signal skulle tolkas. Tydligen var videodatan överlagrad på synksignaler. Att separera ut dessa synksignaler själv verkade för avancerat så för detta ändamål införskaffades en LM1881. För att skapa en digital representation av den analoga videosignalen samplas denna. Vid närmare beräkningar visade det sig att processorn var för långsam för att handha inläsningen av bilden. Lösningen som användes var att använda snabba räknare för att peka på vilken adress det av AD omvandlaren samplade värdet skulle sparas.

Kretsschema och virning

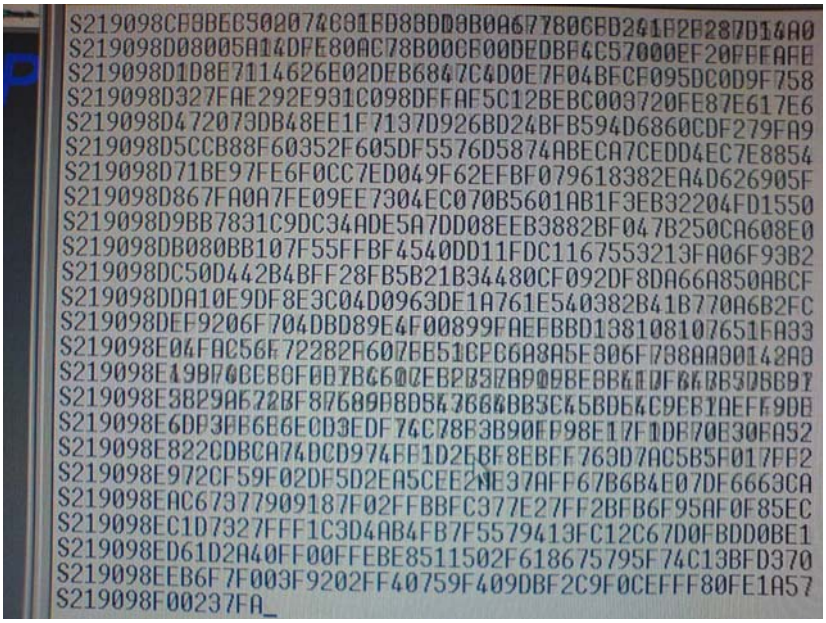
Ett schema över hur processor, lattice, minnen, räknare samt AD omvandlaren skulle placeras gjordes i Powerlogic. Alla komponenter virades på plats och efter det lödades strömförsörjningen, denna ordning är dock inte att rekommendera! Virningsmetoden är att föredra under utvecklingsfasen då förändringar i kopplingen är tämligen små ingrepp. För att minska oönskade strömmar i jordplanet avkopplades samtliga komponenter.

Testning

Först ut var LM1881 som skulle plocka ut synksignaler från videosignalen. Att kraftiga störsignaler var närvarande vid koppling på kopplingsbord var känt, men att de skulle ha förödande konsekvenser för synkseparatorn var ovisst och tidsödande. Korta avstånd god avkoppling samt sammankopplad digital och analog jord var ett måste. När kretsen efter lång tid började fungera sattes den på plats och mätningar visade att signalförutsättningarna var kraftigt förbättrade.
Testningen av AD omvandlaren var bristfällig med den fungerade fint med stabila signaler från ett nätaggregat. Så efter detta enkla test monterades den på plats.
Nästa steg var att programmera logikdelen, detta gjordes i abel. Detta program till Latticen initierade en bildinläsning, detta steg tog några veckor. Först använde vi den externa 8MHz klockan för att klocka både AD omvandlare och de externa räknarna. Detta visade sig sedan inte gå att synka med WE (Write Enable) signalen till SRAMen. Resultatet var att bara skräp skrevs in i minnet. Först när Latticen helt styrde bildinläsningen fungerade inläsningen i minnet bra. Dock blev det idel nollor, videosignalen låg utanför AD omvandlarens intervall. Först nu insågs varför gedigen testning av olika komponenter är att föredra. Videosignalen låg 1.5V under AD omvandlarens intervall. Alla försök med att sänka upptagningsintervallet på ADn misslyckades. Det gick inte häller att med OP förstärkare förstärka signalen eller som summator addera 1.5V, frekvensen var för hög. Lösningen verkade mycket avlägsen men visade sig snart vara både nära och enkel. Två våningar ner fanns två mycket hjälpsamma och kompetenta personer som förklarade hur man skulle göra, tre resistorer var allt som behövdes. Perfekt, allt på plats och bara mjukvara kvar att fila på. Under de senare testerna användes it68 som emulerade en CPU. Nu gick det att, via hyperterminalen, ansluta till utvecklingssystemet för att söka igenom minnet eller göra minnesdumpar till en dator.

Mjukvara

För att kontrollera vad som lagrats i minnet gjordes en dump av SRAMen via Hyperterminalen, en långsam process men förhoppningsvis mycket givande. Filen som speglade minnet var i S2 format, ett slags rådata format som krävde lite stränghantering innan den var till nytta. Figur 3 illustrerar hur en dumpning av SRAMen kan se ut.



Figur 3: *Skärmbild från det tidsödande momentet: minnesdump till datorn.*

Av erfarenhetsbrist i snabbare programmeringsspråk skrevs denna procedur i PHP. En titt i den redigerade rådatan visade att h-synkarna uppenbarade sig som segment av idel nollor. Med en enkel faltning kunde således bilden delas upp till en matris. Resultatet, figur 4 och 5, var klart upplyftande!

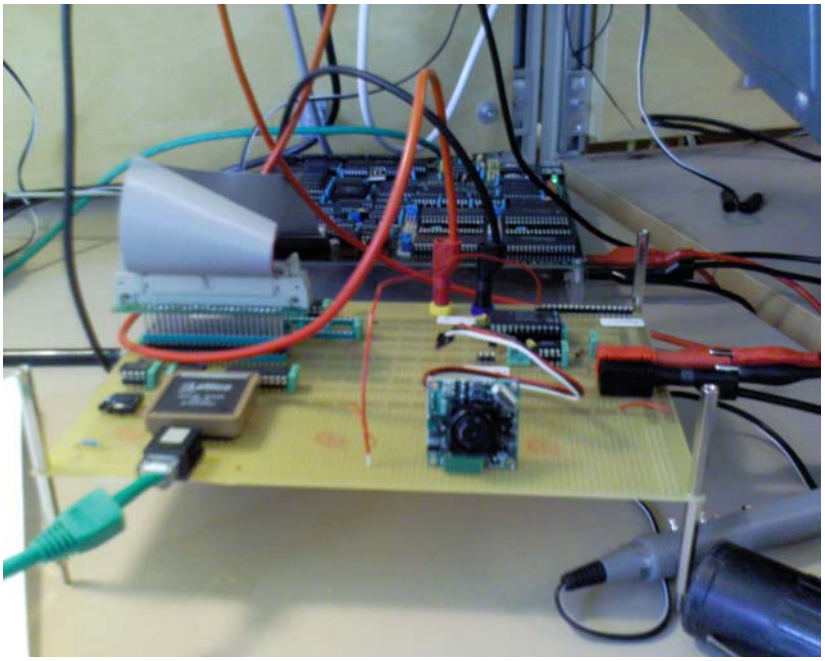
Figur 4 och 5: *Fredrik och Hans*

Givetvis finns det mer att önska av kvalitén, men för vår applikation är detta fullt tillräckligt. I skrivande stund är ingen C-kod klar men implementationen torde vara enklare än den i PHP då endast ett x-värde för ljusstarkaste punken skall beräknas.

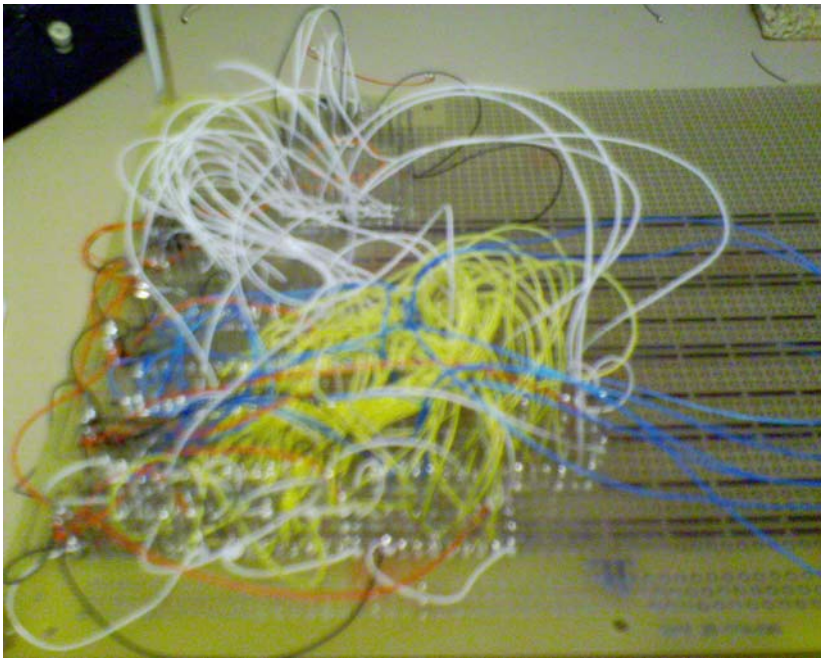
Sammanfattning

Att tro att den stora kursen innebär att man har gott om tid är helt fel. Det finns hur många ställen som hälst att göra fel på men bara 16 veckor till felsökning. Att få upp rutin för felsökning är nyttigt men framförallt erfarenheten om att genomtänkt testning är viktigt i ett projekt som detta. Vidare vill vi båda framhålla att detta är en ypperlig kurs för att få ’hands-on’ erfarenhet på många hittills endast teoretiska plan. Vad gäller detta projekt blev vi inte klara med C programmeringen i utsatt tid. Men vårt största mål blev uppfyllt, vi lyckades ta bilder med vår kamera. Förbättringar finns givetvis att önska. Största förlustfaktorn i bildinläsningen är att videosignalen inte är helt matchad med AD omvandlaren. Detta skulle kunna bli möjligt om man antingen lyckades förstärka signalen eller anpassa input området hos AD omvandlaren. Detta låter vi dock vara en notering för framtida utmaningar, bildkvalitén är för detta projekt tillräcklig.

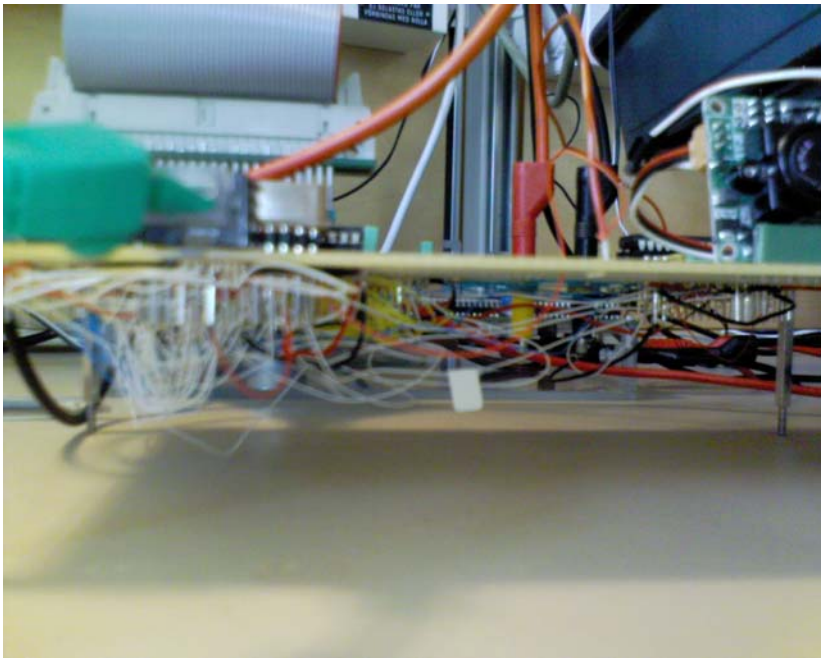
Bilder



Bakom vårt bygge syns utvecklingssystemet it-68.



Redan efter några timmars virande var skatboet ett faktum.



Ytterligare en bild

Kopplingsschema