Försvarets Historiska Telesamlingar
Flygvapnet

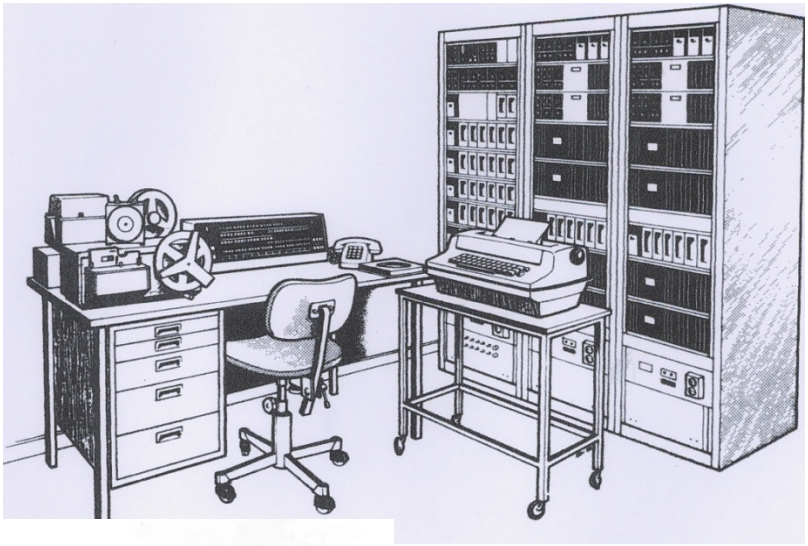
2013-03-17

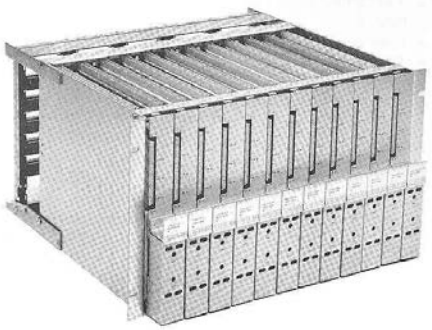
Beskrivning av Censor 932

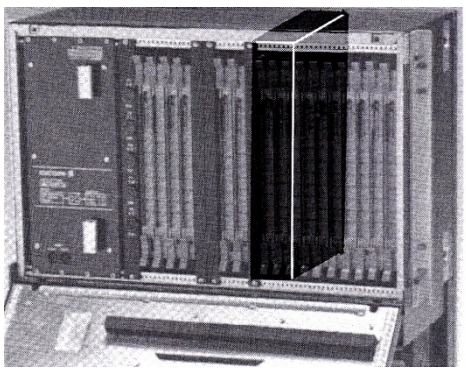
Bilaga 1
till
RRGC/F En viktig komponent i Stril m/60 Del 2

Hans Borgström och Bengt Olofsson

F02/13







Innehåll

1 Sammanfattning

Med Censor 932 fick rgc en för den tiden modern och generellt användbar dator som samtidigt var försedd med funktioner som gjorde den särskilt användbar i realtidssystem av den typ som rgc representerade. I datorsystemet C932 ingick dessutom ett avancerat databussystem som öppnade vägen dels för flexibel integration av datorerna Censor 220 och Facit DS 9000 och dels för kommande utökningar av datorsystemet i rgc. Detta var ett mycket viktigt steg i den tekniska utvecklingen av rgc.

Med Censor 932 och dess bussystem kunde nu de tidigare besvärande minnesbegränsningarna elimineras genom att bussystemet medgav flexibel inkoppling av dels primärminne (kärnminne och senare halvledarminne) och dels olika typer av sekundärminnen (skivminne, bandstation etc.).

Kapaciteten i rgc datorsystem kunde nu med hjälp av bussystemet utökas genom tillförande av nya datorer. Snabb dator-dator kommunikation realiserades på ett enkelt sätt med hjälp av databussystemet. Under hela tiden fram till avvecklingen av rgc utnyttjades bussystemets utbyggnadsmöjligheter. Den slutliga datorsystemkonfigurationen blev ytterst avancerad och med ett innehåll av en mängd olika typer av datorer.

Med Censor 932 kom dessutom en basprogramvara som utgjordes av ett modernt s.k. multiprocessing operating system (OS). Programutveckling i Censor 932 bedrevs enligt moderna principer i ett terminalbaserat time sharing system (TSS). Till stöd för rationell programutveckling tillhandahölls ett antal hjälpmedel i form av bl.a. en symbolisk assembler, ett spårningsprogram för test och verifiering av utvecklad programkod och ett systemgeneringsprogram för sammanfogning av programmoduler till ett slutligt programsystem.

Under hela tiden fram till avvecklingen av rgc utvecklades Censor 932 i flera omgångar. Inledningsvis utgjorde Censor 932 i princip en centralenhet (CPU) med enbart primärminne som datorns minnesenhet och med ett remsbaserat ”sekundärminnessystem”. I takt med ökade behov ersattes ”remsorna” med ett sekundärminnessystem omfattande både magnetband och skivminne. Genom tillkomsten av nya typer av integrerade kretsar och ny uppbyggnadsteknik kunde Censor 932 göras fysiskt mindre. Basprogramvaran förbättrades kontinuerligt genom tillförande av nya funktioner och förbättringar.

Följande uppbyggnadstekniska varianter av C 932 har använts i rgc:

- C 932K (kassettvarianten)
- C 932V (virkortsvarianten)
- C 932E (europakortsvarianten)

De tre varianterna var helt kompatibla med varandra och samma program kunde köras i alla varianterna.

2 Censor 932K, Introduktion

Censor 932K, som konstruerades av SRT under åren 1967-68, kan sägas var en representant för 2:a generationens datorsystem uppbyggd av den tidens mest populära mikrokretsar (i dag benämnda som SSI-kretsar) och med en struktur som gjorde datorn allmänt användbar (general purpose computer).

Förebilden för Censor 932 systemarkitektur var det under slutet av 60-talet förhärskande datorsystemet IBM 360. Datorns instruktionsrepertoar är i dess struktur hämtad från detta system. Censor 932 var tänkt att utgöra basen för olika typer av datorbaserade ledningssystem och marknadsfördes på olika sätt med broschyrer, annonser etc.

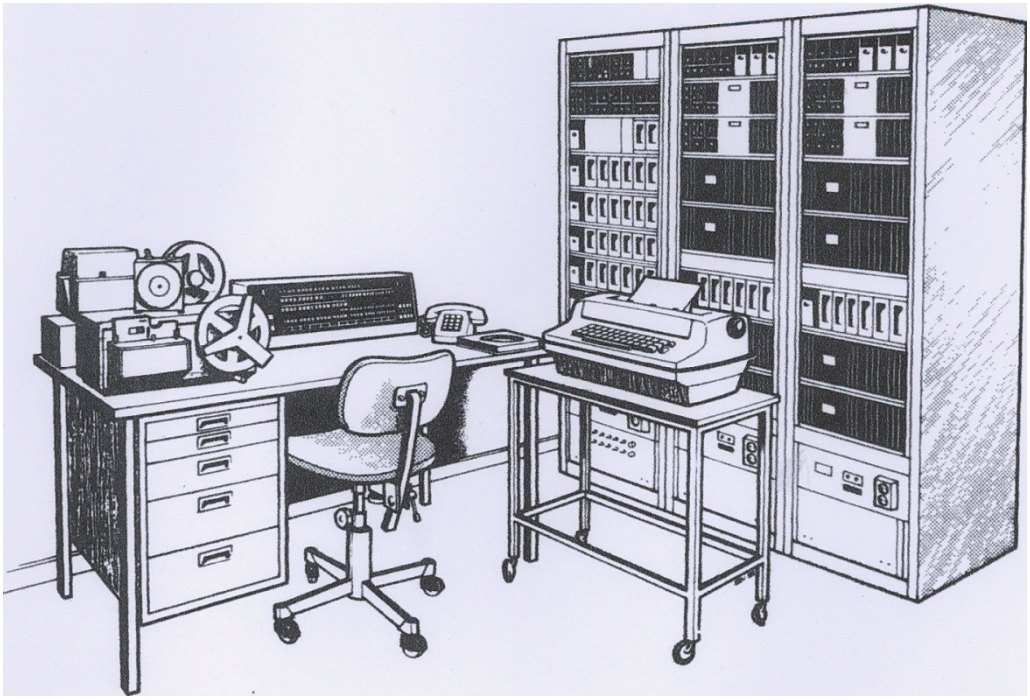
Andra styrande faktorer för Censor 932 systemarkitektur var de problem man haft med Censor 220 i form av otillräckligt med primärminne samt i praktiken avsaknaden av ett flexibelt system för inkoppling och hantering av perifer utrustning (skrivare, massminne etc.). I Censor 932 löstes dessa problem genom införande av ett minnes- och busshanteringssystem som medgav tillkoppling av i praktiken hur mycket minne som helst och anslutning av obegränsat antal typer av perifer utrustning till en Censor 932 genom dels anslutning av programkontrollerad utrustning som skrivare, modem etc. och dels anslutning av enheter som arbetade med direkt minnesacces (DMA). Det avancerade databussystemet gjorde det dessutom möjligt att skapa multidatorkonfigurationer.

Ett viktigt krav var att datorn skulle kunna möjliggöra avancerad multiprogammering (multitasking) i realtidsmiljö. Detta realiserades bl. a genom att datorn försågs med en kraftfull maskinvarustödd programväljarfunktion som kunde användas dels för hantering av utifrån kommande programväxlingssignaler och dels av programvaran skapade programväxlingssignaler. Genom maskinvarustödet kunde programavbrottsfrekevsen ökas drastiskt. En komplett växling av program bestod dels av en maskinvarubunden del om ca 10 mikrosekunder och dels en programvarubunden del om ca 20-30 mikrosekunder. Avbrottsfrekvensen kunde därför höjas till bättre än 2000 avbrott i sek. En för den tiden imponerade prestanda.

Som ytterligare stöd för datorns realtidsfunktion försågs datorn med en programmerbar digitalklocka. Baserat på erfarenheterna från Censor 220 försågs CPU 932 med ett s.k. maskinavbrottsystem som skulle medge programkontrollerad hantering av uppkomna felsituationer (programfel, maskinvarufel etc.) i avsikt att på ett ordnat sätt isolera fel och förhindra systemstopp och därmed öka systemtillgängligheten.

Bilden nedan visar Censor 932K som fristående konfigurationen. Datorn består av tre st. 19”-skåp. Det vänstra skåpet innehåller CPU 932K. Skåpet i mitten innehåller Censor 932K interna primärminne och skåpet till höger innehåller ett systemgemensamt primärminne (MAS-minnet). Kassettraderna i mitten av skåpen utgör elektroniken för Censor 932K databusssystem. I minnesskåpen anslöts kärnminnesmodulerna till bussystemet. Genom en arbitratorfunktion (AM) i stativet med det systemgemensamma minnet kunde övriga enheter och CPU:er få åtkomst till det gemensamma minnet. Genom en bussväxelfunktion skapades en möjlighet för CPU 932K att kunna adressera det systemgemensamma minnet.

På rullbordet och skrivbordet till vänster i bilden finns systemoperatörens utrustning i form av konsolskrivmaskinen IBM Selectric (känd för sitt typhuvud i form av en kula), Facit remsläsare för laddning av program och data, Facit remsstans för utmatning av data samt datorns manöverpanel OCP (Operators Control Panel).



Censor 932K, fristående konfiguration

3 Elektromekanisk uppbyggnad

3.1 Allmänt, Standardiserat mekaniskt uppbyggnadssystem

I likhet med strävandena att utforma Censor 932 att utgöra en generellt användbar dator så tillämpades även denna filosofi på valet av elektromekanisk uppbyggnadsteknik för densamma. Precis som fallet varit under datorteknikens barndom med allsköns specialkonstruerade ”monster” så hade även det mekaniska uppbyggnadssättet skapat en flora av icke passande systembitar. För att göra det möjligt att skapa en industri för massproduktion av mekanik för elektronik startade man under mitten av 60-talet olika grupper som skulle lösa problemet med att komma överens om en mekanisk byggstandard för elektronisk utrustning. Inom Standard Radios ägarbolag ITT beslutade man anta den s.k. ISEP-standarden (International Standard Equipment Practice) som standard för de skåp i vilka elektronik skulle monteras. ISEP-standarden var på alla väsentliga punkter lika med den av ECMA (European Computer Manufacturs Association) samtidigt publicerade Europastandarden för 19”-system. Beslutet innebar att det mekaniska byggsättet som tillämpats för Censor 220/120 övergavs till förmån för 19”-systemet ISEP. Nu kunde man bygga med billiga byggelement inköpta på den öppna marknaden. Mekanik och elektronik kunde måttsättas enligt en antagen standard.

3.2 Allmänt, elektroniken baseras på mikrokretstekniken

Inför konstruktionen av Censor 932 hade man inom SRT bestämt att den nya datorn måste vara baserad på den nya mikrokretstekniken. Ett stort undersökningsarbete genomfördes i avsikt att välja någon av den tidens (mitten av 60-talet) tillgängliga kretsfamiljer. Många olika krav skulle uppfyllas. De flesta byggde på de erfarenheter man dragit från uppbyggnaden av Censor 120/220. Genom att tillämpa mikrokretsar kunde man dels få till stånd platsbesparing och dels eliminera de problem som alltid finns med spridningen hos egenskapsvärdena (exempelvis kapacitanser) för diskreta komponenter. Med införandet av mikrokretsar kunde man få bättre kontroll på egenskaperna hos de producerade elektronikmodulerna och samtidigt förenkla systemkonstruktionen genom att man i princip enbart kunde ägna sig åt att dimensionera elektroniken efter kretsarnas belastningsfaktorer (fan in/fan out) samt angivna fördröjningstider. Övriga problem var så att säga isolerade och osynliga i varje kretskapsel. Bland de kretsfamiljer som undersöktes kan nämnas RTL (Resistor Transistor Logic), DTL (Diod Transistor Logic), TTL (Transistor Transistor Logic) och slutligen olika varianter av ECL (Emitter Coupled Logic). Resultatet blev att man beslutade använda TTL-kretsar för uppbyggnad av den nya datorn.

Elektriskt uppfyllde TTL-familjen de olika kraven bäst av alla de undersökta kretsfamiljerna och kommersiellt bedömdes att denna kretsfamilj kunde tänkas överleva för lång tid. Det var, som vi idag kan konstatera, ett bra beslut. Samma komponentfamilj kan man ännu i dag köpa hos Elfa 45 år (!) efter beslutet.

3.3 Vild kabeldragning.

Övergång till ISEP-standard innebar att elektronikskåpen för Censor 932K konfigurerades med fasta monteringsramar för kretskort och att tekniken med löstagbara kortramar och lödda förbindelser som tillämpats för Censor 220/120 övergavs. Olika tekniker för förbindningar mellan signalstift undersöktes. Man fann att den vid mitten av 60-talet lanserade prototyptekniken Wire Wrap skulle vara en ekonomiskt försvarbar tillverkningsteknik även vid produkt-

ion av mindre serier (som ju var fallet för Censor 932). Elektriskt innebar virtekniken ingen försämring relativt kretskorttekniken, snarare tvärtom. Med virteknik skapas en förbindelse mellan två punkter genom att mekaniskt vira en tråd kring anslutningspunkternas signalstift och att tråden drogs närmaste väg dvs. den drogs inte i särskilda kabelrännor. I fallet Censor 932 tillämpades virtekniken på så sätt att allt signalkablage i stativet utfördes med tvinnad tråd och att anslutningskabeln mellan signalstiften gjordes med kortast möjliga kabellängd. Resultat blev utifrån betraktat ett skatbo men elektriskt ett närmast välorganiserat system.

3.4 Kortlådan

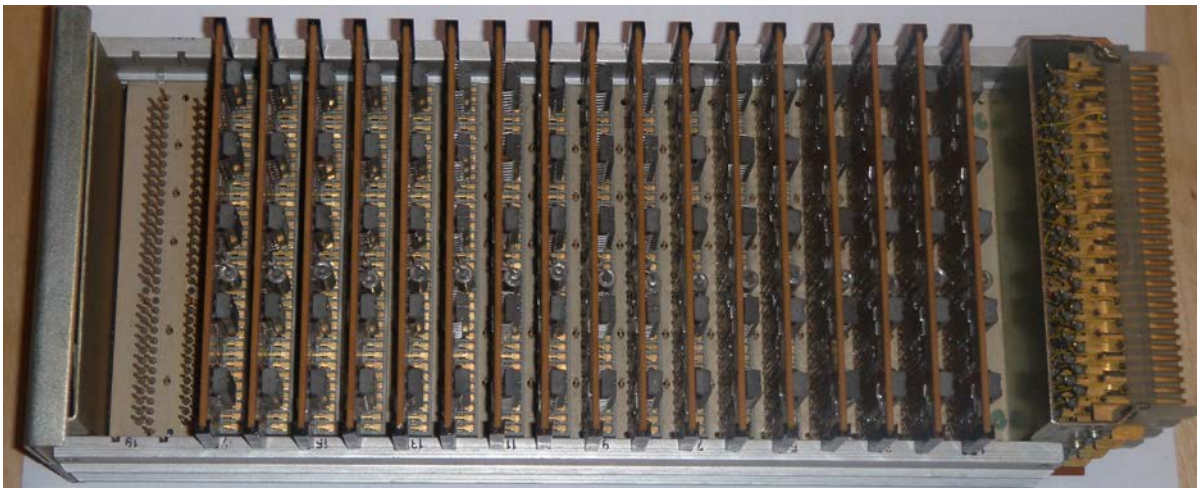
Vid samma tidpunkt som beslut togs om övergång till ISEP-standard och wire-wrap-teknik togs även beslut att Censor 932K inte skulle baseras på kretskort enl. den modell som tillämpats för Censor 220/120. Skälet var att det för detta system över tid visat sig bli opraktiskt, dyrt och tidskrävande att införa de med tiden oundvikliga ändringarna på befintliga kort. Uppbyggnaden av elektroniken för Censor 932K skulle istället vara så utförd att införande av modifieringar, ombyggnader etc. skulle kunna göras på kortast möjliga tid. I princip skulle man efter ändring av logikschemat direkt med virverktyget och lite virtråd kunna införa ändringen på plats.

Elektronikkomponenterna, som nu utgjordes enbart av integrerade kretsar i s.k. DIL-förpackning löddes på enkla kretskort som fungerade som bärare av ett antal integrerade kretsar av samma typ utan inbördes förbindning. DIL-kapslarnas ben förbands via kortfoliet med kretskortets kontaktdon. Kretskortens storlek var 98 * 42 mm och hade ett 70-poligt kontaktdon.

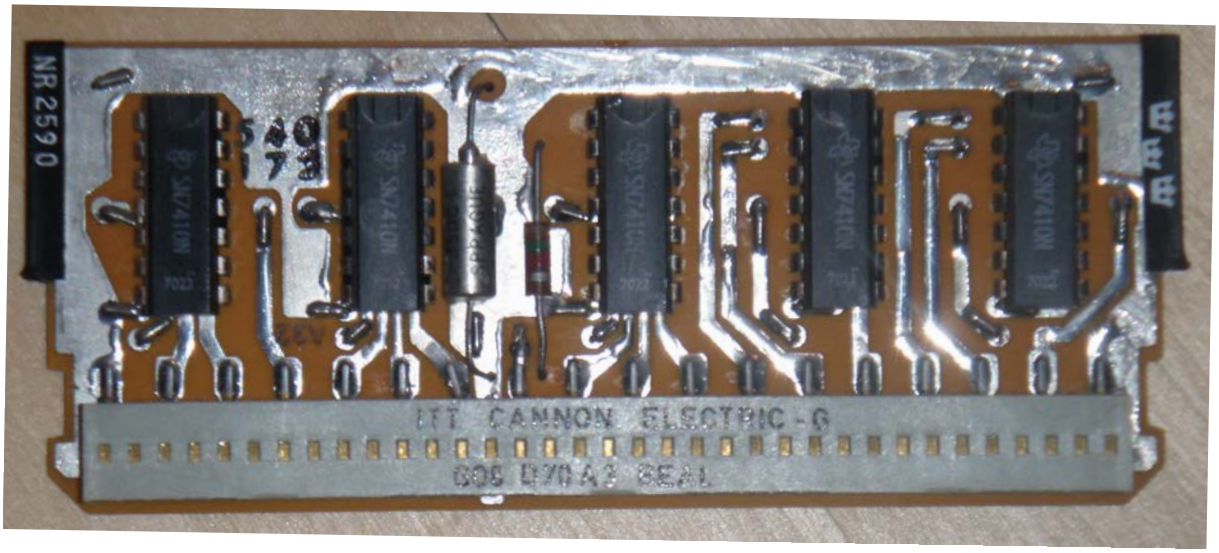
Kretskorten placerades i ett kortmagasin med plats för 19 kretskort med upp till 5 integrerade kretsar/kort eller totalt 95 kretsar per kassett. Kortmagasinet var utformat som en 11 cm bred, 6 cm hög och 25 cm lång låda med en 3 mm bottenplatta av aluminium. Lådans ena ände hade ett infällbart handtag. I den andra ändan placerades 4 stycken 33 pols SEL kortkontakter. Mellan var och en av de 4 kontaktdonen placerades ett förgyllt fosforbronsbleck (den s.k. jordkniven) med anslutning för 33 stycken virtrådar. Genom arrangemanget blev det nu möjligt att ansluta 128 tvinnade signalkablar. Signalkablarnas signalledning anslöts till SEL-donet och returledningen till jordkniven.

Magasinet bestyckades med de korttyper som behövdes för att realisera den funktion som kassetten var konstruerad att utföra (jämför kassetten med folierat kretskort). Funktionen erhölls sedan genom att förbinda olika kretsar (grindar och vippor) enligt logikschemat för kassetten. Förbindningen internt i kassetten utfördes genom virning med enkel tråd. Kortens jordplan förbands med speciella jordstift i bottenplattan.

Kortmagasinet placerades i de stativfasta kortramarna som kunde bestyckas med 6 magasin. I kortramen fanns på motsvarande sätt 4 stycken SEL-don för varje magasin och mellan donen monterades ett gaffellikt förgyllt fosforbronsbleck. Vid anslutning av ett kortmagasin till kortramen slöt gaffeln om båda sidor på magasinets jordkniv och en perfekt returförbindelse hade skapats. På magasinets bottenplatta monterades rader av inpluggade virstift som svarade mot kretskortens kontakter.



Kortlådan



Kort

3.5 Kraftmatningssystem

I toppen av varje skåp monterades nödvändiga kraftaggregat (+ 5V råkraft, +12 V – 12V) samt en kontrollenhet för spänningsövervakning. Likspänningarna matades ut till kortramarna och deras kortmagasin via 4 kopparskenor monterade på stativets vänstersida. Råkraft + 5V matades till spänningsstab monterad i på varje kortrams vänstra plats. Från kopparskenorna kopplades sedan med tvinnad kabel till de önskade spänningarna +12Voch -12 V till resp. magasin. +5V matades från spänningsstabben i ramen vid matningsskena till resp. kassett.

Beroende på det aktuella systemets konfiguration kunde kraftmatningssystemet anpassas efter angivna krav på spänningsövervakning och centralt krafttillslag.

I fallet Censor 932K fungerade kontrollenheten i stativet för CPU 932K som övervakningsenhet för samtliga skåp ingående i Censor 932K d.v.s. CPU-skåpet, internminnesskåp samt MAS-skåpet. Med kontrollenheten realiserades centralt krafttillslag och central kraftövervakning av dessa skåp. Vid krafttillslag övervakade kontrollenheten att samtliga matningsspänningar i samtliga skåp uppnått sina specificerade nivåer. Under tiden innan matningssystemet

blivit redo genererade kontrollenheten en ”system reset” som förhindrade att CPU-verksamheten kunde starta. När ”redo-läge” inträffat hävdes system reset och systemets olika delar kunde tas i bruk.

För CPU 932K innebar detta att funktionen Power on utförde initiering av CPU’s olika register.

3.6 Bussystem.

3.6.1 Inledning.

Driftserfarenheterna från datorsystemet i rgc pekade på att ett databussystem i ett nytt datorsystemkoncept måste var betydligt immunare mot störningar än det som använts i Censor 120 och Censor 220. Man konstaterade samtidigt att ett datorsystems databussystem är den komponent i ett systembygge, som om det är väl utfört, möjliggör byggande av prestandamässigt välbalanserade system och utgör grunden för integration och sammankoppling av olika typer av datorbaserade delsystem. I rgc gällde det inledningsvis att sammankoppla Censor 932 K med Facit DS 9000, Censor 220 och Censor 120. I fortsättningen kopplades sedan varje tillkommande dator i rgc till bussystemet som tillslut blev oerhört komplext.

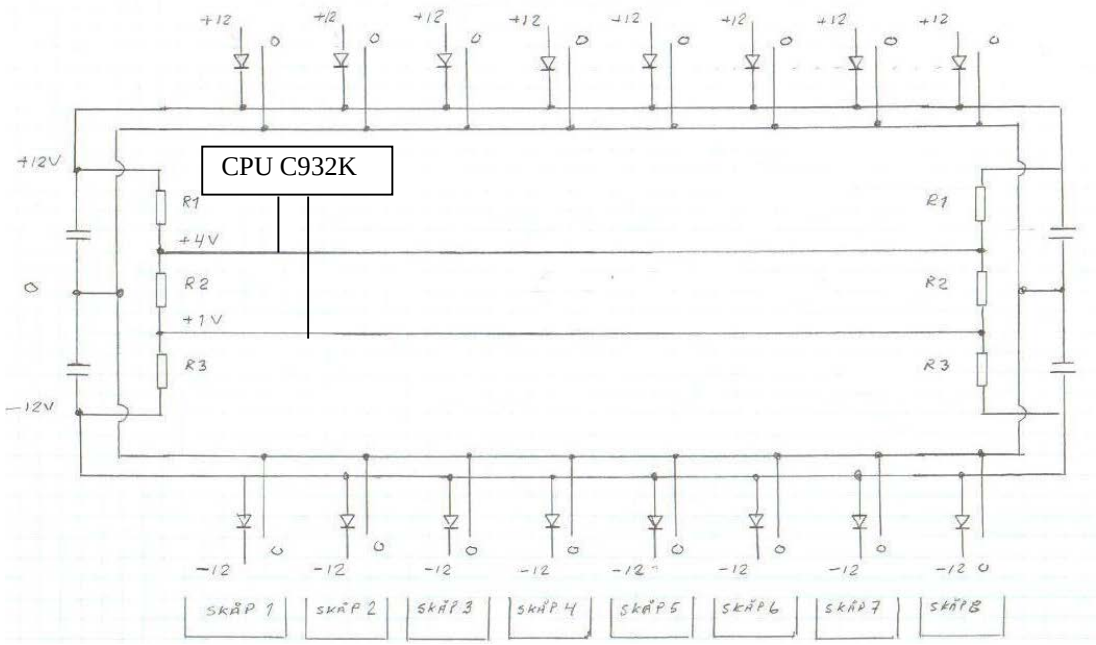
Med de vunna driftserfarenheterna och de nya insikterna i ryggen genomfördes inom SRT en rad studier och experiment med målet att finna ett koncept för ett framtidssäkert databusssystem. Man fann att bästa störimmunitet erhölls om varje enskild bussledning (bussen omfattade total mer än 60 signalledningar) utgjordes av en partvinnad ledning som var elektriskt balanserad. Problemet med överhörning och s.k. common modestörningar kunde då kontrolleras. Genom att bussens ändpunkter försågs med anpassningsdon (bussledningen gavs en karakteristisk impedans) minskade dessutom problemen med de reflektionsstörningar som normalt uppstår i en transmissionsledning.

Resultatet av det hela blev till slut den s.k. kniptångsbussen. Systemets grundkomponenter blev enheterna SI (storage interface) och BB (bus buffer) för anslutning av minnesmoduler till bussen, en enhet benämnd AM (auto access multiplexor) för styrning och tilldelning av access till bussen samt viktigast av allt bussens elektriska och funktionella specifikation. Nu kunde varje konstruktör förse sina konstruktioner med korrekta bussgränsytor. I konceptet för den nya bussen ingick även teknik för att råda bot på de elektriska problem som alltid uppstår vid inkoppling och urkoppling av enskilda enheter i ett bussystem under drift.

3.6.2 Kniptångsbussen, teknisk beskrivning

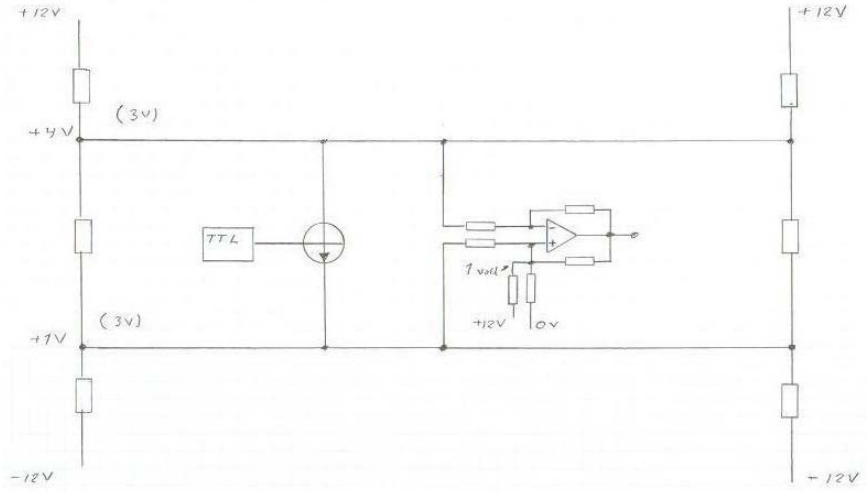
Elektrisk uppbyggnad.

Kniptångsbussens uppbyggnad visas i bilden nedan. Varje bussledning utgörs av ett trådpar som avslutas i bussens ändpunkter med anpassningsmotstånden R1, R2 och R3. Varje bussledning ”matas” med +12V och -12V från spänningsaggregat i varje stativ som den passerar. Busslängden kunde vara upp till 10 m och på sin väg kunde den passera ett stort antal skåp. De två ledningarna mellan motstånden R1 och R3 i figuren är ett balanserat ledningspar som funktionellt utgör en dubbelriktad kommunikationslänk. I bilden nedan visas hur ett ledningspar i bussavnämaren CPU 932 K ansluts till bussen.



Kniptångsbussens uppbyggnad

I varje bussavnämare (exempelvis CPU 932 K) ingick ett elektriskt interface i enlighet med bilden nedan. Varje databit avnämaren önskade koppla till bussen gjorde detta via detta elektriska interface. Transistorn i figuren utgör den anslutna enhetens ”bitsändare”. När en ”etta” skulle läggas ut på bussen sattes transistorn i sitt ON-läge som elektriskt sett ”knep ihop” bussledningen. När transistorn var i sitt OFF-läge (”nolla”) öppnades ”tången”. Operationsförstärkaren i bilden nedan registrerade bussledningens ”knipläge”-och ”tångläge” som etta resp. nolla.



Elektriskt interface

Mekanisk uppbyggnad
Bussen omfattade följande delar:

- 32 + 4 ledningspar för data resp. paritetsbitar
- 22 ledningspar för minnesadressdata
- ca 10 busstyrsignaler

- två unika ledningspar per ansluten enhet (CALL/RESPONSE) som används för tilldelning av bussaccess

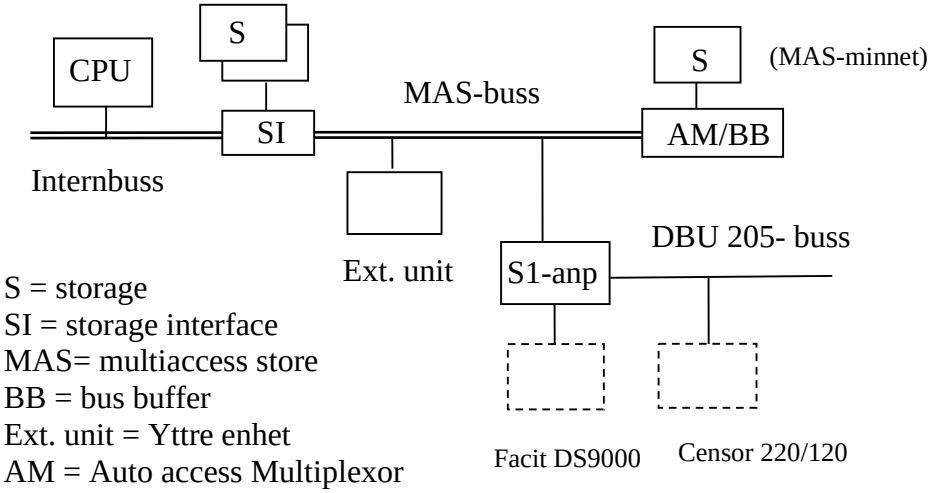
Bussledningarna kopplades till respektive avnämare i ett stativ och avslutades sedan i det s.k. ELCO-donet för vidarekoppling till angränsande stativ.

3.6.3 Censor 932K, inkoppling i rgc

Tidigt konstaterades att främst minneskapaciteten hos Censor 220 inte räckte till för den funktionalitet man önskade sig. Censor 220 behövde ersättas med en kraftfull dator anpassad för realtidsapplikationer. Beslutet blev att i ett första steg installera Censor 932K och ett ”systemgemensamt minne” (MAS) och överflytta (konvertera) huvuddelen av det operativa programsystemet till Censor 932K samt att använda Censor 220 enbart för radarsimuleringsfunktionen. Censor 120 och DS 9000 skulle behållas oförändrade. Detta medförde att Censor 932K måste kunna samverka med de ”gamla maskinerna”. I bilden nedan visas hur Censor 932K med hjälp av Censor 932 bussystem kopplades samman med Censor 220, Censor 120 och Facit DS 9000. Av prestandaskäl delades bussystemet upp i en internbuss och en externbuss. På internbussen hade CPU 932K via bussanpassarenheten SI access till internminnet S utan att störas av s.k. minnescykelstölder från andra enheter. CPU’n kunde således arbeta med full kapacitet mot sitt internminne.

Via bussanpassarenheten SI hade CPU dessutom accessmöjlighet till det externa minnet S som var kopplat till den externa bussen (benämnd MAS-bussen) via minnesanpassaren BB. CPU måste nu konkurrera om minnesaccesserna till det externa minnet S med övriga enheter anslutna till MAS-bussen. I detta fall Facit DS9000 och Censor 220/120 via bussanpassarenheten, S1-anpassaren. Tilldelning av minnesaccesser på MAS-bussen styrdes av enheten AM (Auto access Multiplexor).

Ur systemsynpunkt innebar denna lösning att externa enheter kunde nå sitt minne utan att störa CPU medan CPU i de fall den adresserade MAS-minnet ”straffades” med viss väntetid. Funktionellt realiserades samverkan mellan datorerna genom datautbyte via MAS-minnet.



4 Teknisk beskrivning

4.1 Introduktion Dator Censor 932 K

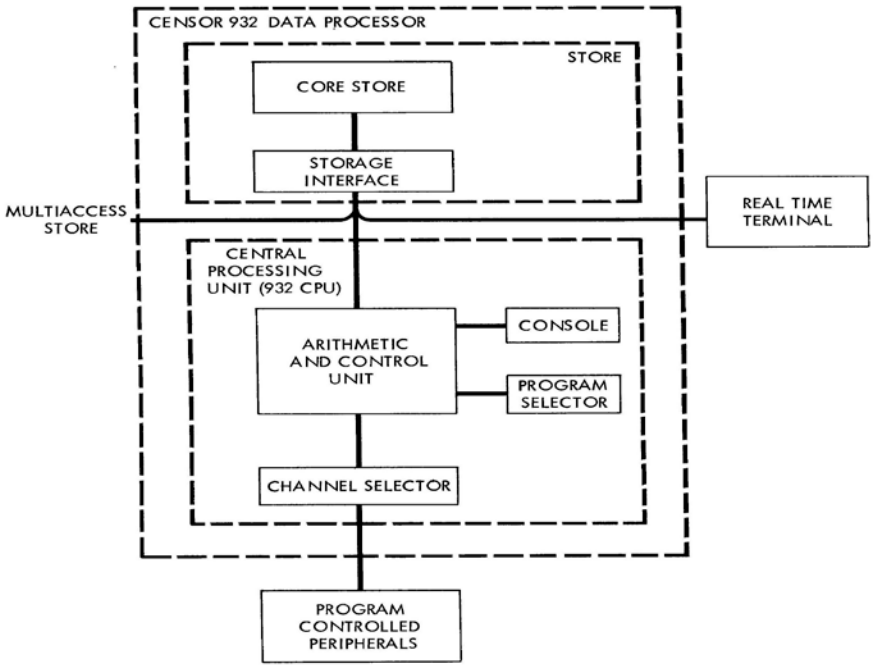
Censor 932K, som konstruerades av SRT under åren 1967-68, kan sägas vara en representant för 2:a generationens datorsystem uppbyggd av den tidens mest populära mikrokretsar (i dag benämnda som SSI-kretsar) och med en struktur som gjorde datorn allmänt användbar (general purpose computer). I detta fall har Texas Instruments TTL-serie 74 använts genomgående. Förebilden för Censor 932 systemarkitektur var det under slutet av 60-talet förhärskande datorsystemet IBM 360. Datorns instruktionsrepertoar är i dess struktur hämtad från detta system. Andra styrande faktorer på Censor 932 systemarkitektur var de problem man haft med Censor 220 i form av otillräckligt med primärminne samt i praktiken avsaknaden av ett flexibelt system för inkoppling och hantering av perifer utrustning (skrivare, massminne etc.). I Censor 932 löstes dessa problem genom införande av ett flexibelt minnes- och busshanteringssystem som medgav tillkoppling av i praktiken hur mycket minne som helst och anslutning av obegränsat antal typer av perifer utrustning till en Censor 932 genom dels anslutning av programkontrollerad utrustning som skrivare, modem etc. och dels anslutning av enheter som arbetade med direkt minnesaccess (DMA). Det avancerade databussystemet gjorde det dessutom möjligt att enkelt skapa s.k. multidatorkonfigurationer.

Andra styrande krav var att datorn skulle kunna möjliggöra avancerad multiprogammering (multitasking) i realtidsmiljö. Detta realiserades bl.a. genom att datorn försågs med en kraftfull programväljarfunktion som kunde användas dels för hantering av utifrån kommande programväxlingssignaler och dels av programvaran skapade programväxlingssignaler. Som ytterligare stöd för datorns realtidsfunktion försågs datorn dessutom med en med en programmerbar digitalklocka.

4.2 Systemöversikt Censor 932K

I korthet beskrivs Censor 932K som en parallellt arbetande 32-bits dator. Datorns primärminne konfigureras med kärnminnesmoduler om 4 kord om 32 bitar + 4 paritetskontrollbitar. Datorns adresseringskapacitet medger adressering av upp till 128 kord om 32 bitar (= 512 kB). I rgc konfigurerades Censor 932 med 4 minnesmoduler om 4 kord vardera resulterande i ett totalt internt primärminne om 16 kord.

Censor 932 K översiktliga systemblockschema framgår av bilden nedan.



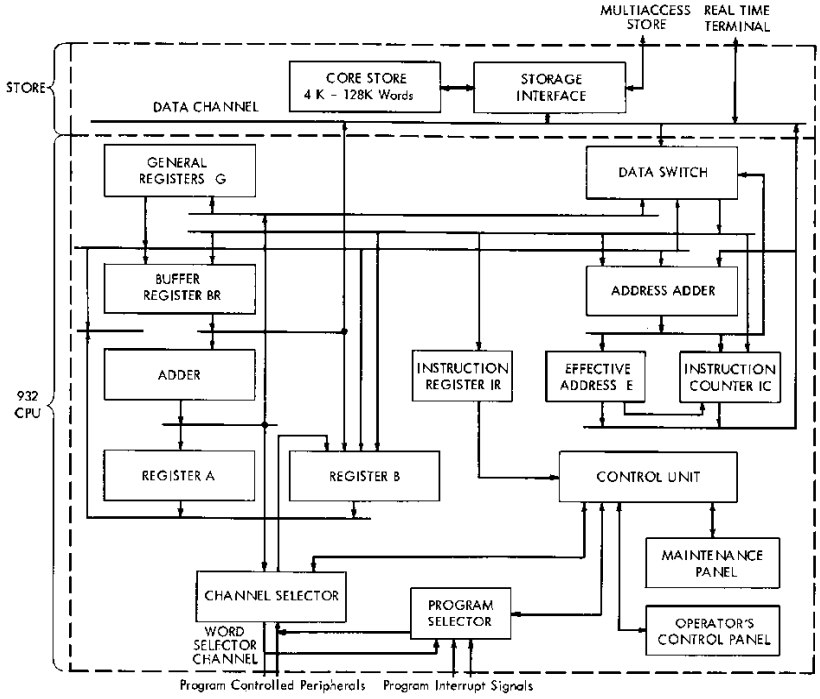
Censor 932K består av följande huvuddelar:

- Centralenhet (Central Processing Unit) innehållande aritmetisk logisk enhet, programväljare, kanalväljare för anslutning av programstyrd kringutrustning som skrivare, remsläsare, remsstans etc.
- Censor 932 interna primärminne (STORE i bilden ovan). Med internt primärminne menas att CPU ensam hade accessrättighet till detta minne. Minnesaccesserna från CPU störtes inte av konkurrerande minnesaccesser från andra systemenheter. CPU kunde på så sätt arbeta optimalt vid programexekveringen.
- Övriga systemenheter med behov av minnesaccess anslöts till en separat minnesenhet som medgav konkurrerande minnesaccesser (Multiaccess store)
- Censor 900 bussystem

4.3 Censor 932 systemuppbyggnad

4.3.1 Systemblockschema

Censor 932 CPU interna uppbyggnad visas i bilden nedan. CPU'n utgörs av följande huvudfunktionsblock:



Systemblockschema

4.3.2 Styrenhet (Control Unit, CU)

Styrenheten CU utgör CPU'ns ”hjärta” som styr och kontrollerar all aktivitet i CPU.

Styrenhetens viktigaste delar är:

- klockpulsgenerator och mikroprogramstyrenhet (Fasregister)
- avkodningsmatris för operationskod i aktuell maskininstruktion
- styrlogik för styrning av Initiell Programladdning (IPL) via remsläsare
- styrlogik för styrning av CPU vid krafttillslag och kraftfrånslag (Power on resp. Power off)
- styrlogik för presentation av CPU-data på manöverpanelen (OCP, operators console) samt för hantering av inmatning från OCP
- styrlogik för hantering av programväxling vid mottagen programanropsignal i program selector (PS)
- styrlogik för hantering av CPU vid inträffade maskinavbrott (Machine interrupt) dvs. när vissa speciella felfall inträffat i CPU. Hanteringens syfte är att man genom att frysa innehållet i CPU'ns samtliga registerfunktioner lagra dessa samt på ett ordnat sätt bryta pågående programexekvering och starta exekvering av programvara för hantering av den uppkomna felsituationen.

4.3.3 16 generella register (G).

Registerna var uppbyggda med den tidens enda MSI-kapsel innehållande en matris med 16 flip-floppar där varje flip-flop adresserades med en unik adress (scratch pad-memory) Dessa kapslar byggdes sedan samman till ett minnesblock om 16 stycken 32-bitars register. Registren var direkt adresserbara i datorns maskininstruktioner. Som en speciell finess kunde registren även adresseras som vanlig primärminnescell.

4.3.4 32-bits adder.

Addern utgjordes av grindnätverk som kunde styras att utföra add/sub, logiska operationer samt shift 1 eller 4 bitar höger/vänster. För att uppnå bästa prestanda var addern dessutom försedd med en snabb logik för hantering av minnessiffra (Carry Look Ahead).

4.3.5 Arbetsregister.

Tre stycken 32-bitars arbetsregister A, B och BR (inte adresserbara i instruktionslisan). Registrens funktion var att hålla operander och mellanresultat vid utförande av maskininstruktionerna.

4.3.6 Adressadder

Adressaddern omfattade dels en 18 bitars adder (E) och dels en 4-bitars adder (JY) för beräkning av adress till nästa instruktion eller adress till minnesoperander. Adressberäkningen styrdes av villkor i aktuell maskininstruktion och kunde utföras i flera nivåer innan slutlig adress (Effective Addres, E) var klar att användas. 4-bits addern kontrollerade detta förlopp.

4.3.7 Instruktionsräknare (IC)

18-bitars register för lagring av adressen till nästa maskininstruktion.

4.3.8 Adressregister (E)

18-bitars register för lagring av adressen (Effective Addres) till aktuell minnesoperand.

4.3.9 Instruktionsregister (IR)

16-bitars register för lagring av den aktuella maskininstruktionens 8-bitars operationskod (OP-code) samt 2 stycken 4-bits register för lagring av maskininstruktionens direkta adresser till registerblocket G alternativt förlagring av maskininstruktionens adressmodifieringsdel IX.

4.3.10 Programselector (PS)

64-bitars register för registrering av upp till 64 olika programavbrottsignaler samt ett 64-bitars register för programstyrd blockering av anropssignaler. Såväl externa som av programvaran internt genererade signaler kan registreras. Signalernas prioritet bestäms av den position de har i registret. Pos 0 (MSB) håller anrop med högsta prioritet och signal med position 63 (LSB) representerar lägst prioritet.

4.3.11 Maskinavbrottsregister (MIR)

24-bitars register för lagring av olika typer av 16 olika felsignaler vid inträffade maskin– och programvarufel under programexekvering. De resterande 8 bitarna i registret lagrade aktuellt maskinstatus vid inträffat fel. Felaktig operationskod, felaktig minnesadress, division med 0, paritetsfel vid läsning av data är några av de felsignaler som registreras i MIR.

Vid inträffat fel, d.v.s. när MIR registrerat en felsignal under programexekvering, bröts den pågående programexekveringen automatiskt. Alla registervärden ”frystes” och deras innehåll överfördes av kontrollenheten till en för maskinavbrottsfunktionen reserverad area i primär-

minnet. Därefter växlade programexekveringen i datorn till ett program för hantering av det uppkomna läget. Genom växling till detta ”felhanteringsprogram” gavs möjlighet för systemets operativsystem att dels analysera och rapportera felet och dels om nödvändigt ”isolera” det felaktiga programmet från fortsatt exekvering för att undvika ett större systemstopp. Efter analys och eventuell isolering kunde operativsystemet sedan fortsätta den ordinarie programexekveringen med reducerad funktion men utan risk för systemhaveri.

4.3.12 Minnesnyckelregister (SK)

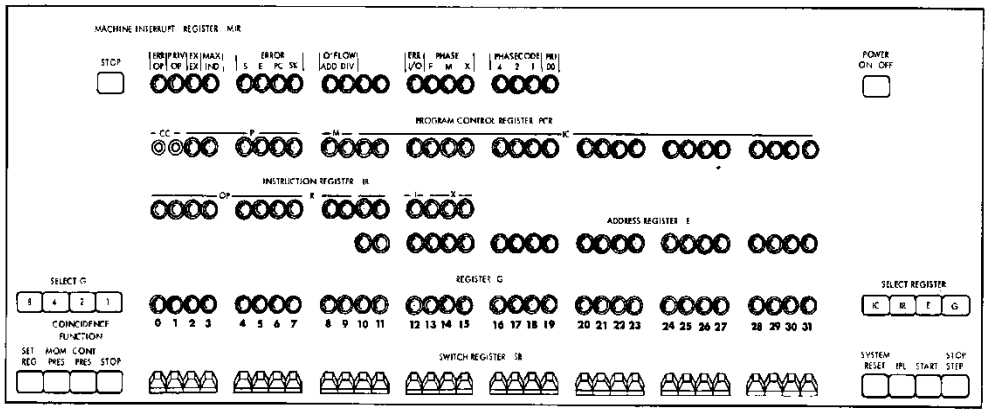
32-bitars register för fast indelning av primärminnesarean i upp till 32 lika delar. Varje bit i registret motsvarade en del av primärminnet. Minnesnyckelfunktionen utnyttjades inte i rgc.

4.3.13 Channel selector (CS)

Standardiserad generell datakanal för anslutning av programkontrollerad kringutrustning till Censor 932K. Med CS skapades ett generellt funktionsgränssnitt för alla typer av kringutrustning som skulle anslutas till C 932K. Datakanalen ombesörjde på ett för alla typer av kringutrustning likartat sätt att hantera adressering av enheter, överföring av data samt hantering av förekommande programanropssignaler. Varje enskild kringutrustning kopplades via en speciell anpassningsenhet som var utformad dels efter den aktuella enhetens specifikation och dels efter funktionskraven hos det generella funktionsgränssnittet. Genom detta förfarande uppnåddes en hög grad av standardisering vid anslutning av kringutrustning till C932K.
I rgc kopplades anpassningsenheter för Facit remsläsare (PTR), Facit remsstans (PTP) samt skrivmaskinen IBM Selectric (TTY) till CS.

4.3.14 Operators console panel (OCP)

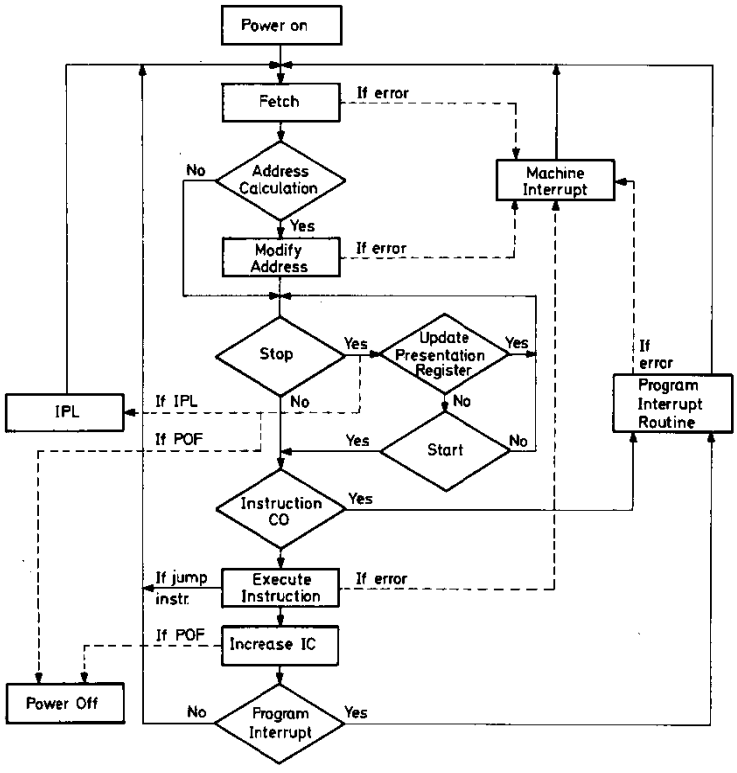
OCP (se bilden nedan) utgörs av en bordsplacerad enhet som användes av systemoperatören för övervakning och styrning av CPU. OCP bestod av ett antal lamprader där varje rad representerar en typ av CPU-register och varje bit i det aktuella registret representeras av en lampindikator. För inmatning av data från OCP användes 32 omkopplare monterade längst ner på panelens front. På OCP presenterades kontinuerligt innehållet i de register som representerar aktuellt systemtillstånd (innehållet i interna arbetsregister presenteras inte). Operatören kunde i stoppläge (mode) manuellt ladda eller ändra register- och minnesdata. Med panelens ”stegafunktion” kunde operatören/programmeraren manuellt stega sig igenom ett program vid felsökning eller programkontroll. Med panelens ”koincidensfunktion” kunde operatören skapa en ”programbrytpunkt” för presentation av visst registervärde alt för att stoppa CPU vid den aktuella brytpunkten.



Operatörspanelen

4.4 CPU funktion

Det av styrenheten CU skapade arbetsflödet i CPU visas i flödesschema i bilden nedan.



Styrenhetens flödesschema

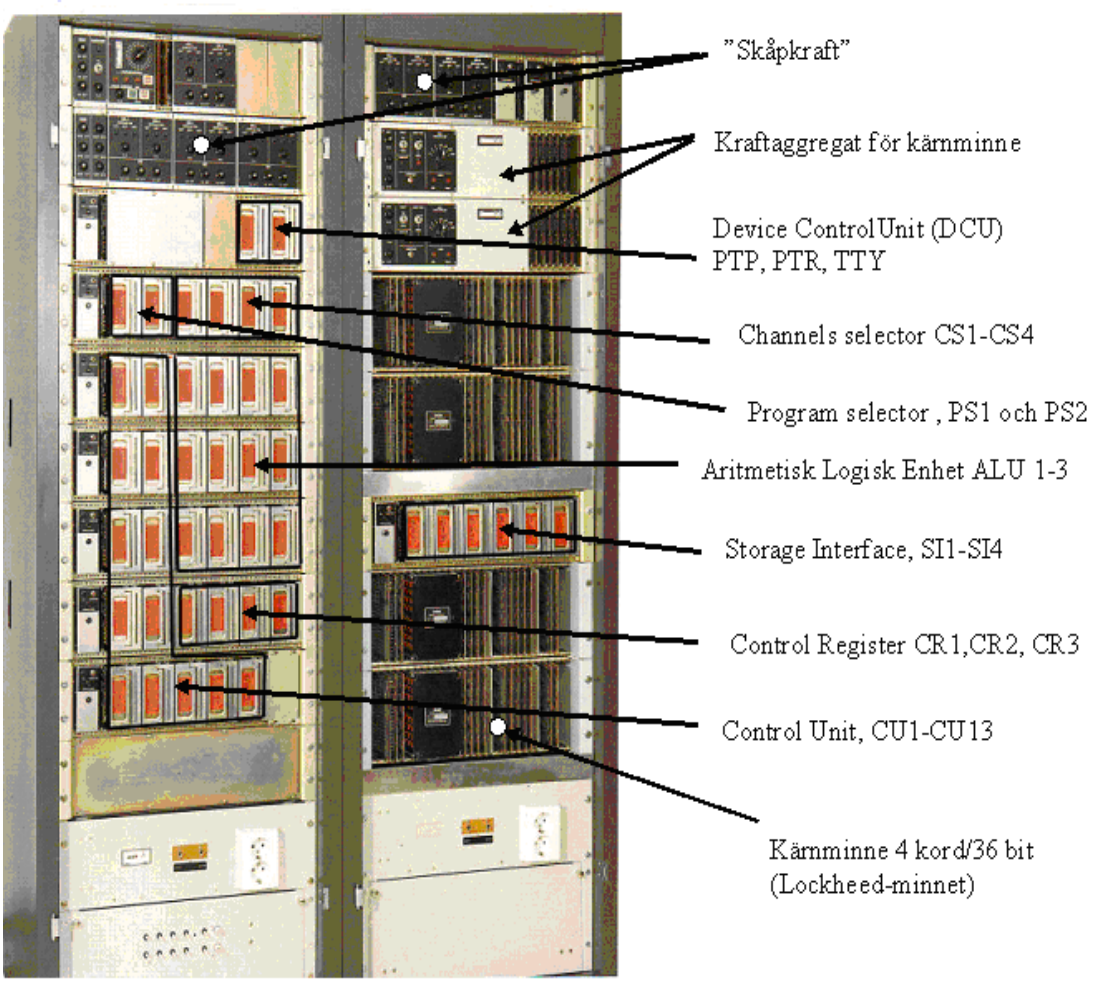
Vid krafttillslag startar CPU i arbetsfasen Power on med att ladda datorns register med de data som lagrades i primärminnet under föregående arbetsfas Power off. I och med att arbetsfas Power On är avslutad kan CPU starta programexekvering från det systemläge som gällde vid senaste kraftfrånslag. Programexekveringen påbörjas genom att CPU aktiverar arbetsfas Fetch som innebär att ny maskininstruktion hämtas från primärminnet och avkodas. Om adress till instruktionens minnesoperand skall beräknas aktiverar CPU arbetsfasen Modify Address. När adressen till minnesoperanden är beräknad aktiverar CPU arbetsfasen Execute i vilken den aktuella maskininstruktionen utförs, varpå CPU därefter aktiverar arbetsfas Fetch

för hämning av ny maskininstruktion. Och så här ”snurrar” det på instruktion för instruktion, förhoppningsvis utan avbrott.

Systemvillkor som bryter det normala flödet Fetch > Modify > Execute kan vara ett inträffat fel som resulterar i att CPU aktiverar arbetsfas Machine Interrupt eller att datorn registrerat en programanropsignal med sådan prioritet att en programväxling måste genomföras. I detta fall aktiverar CPU arbetsfasen Program Interrupt. Det normala arbetsflödet tillåts dessutom att brytas om operatören kommenderat stopp via Operators Console (OCP) eller om initiell programladdning skall genomföras med funktionen IPL. I båda fallen aktiverar CPU arbetsfas Stop.

4.5 Stativkonfiguration

Bilden nedan visar konfigurationen hos de två stativ som tillsammans utgjorde Censor 932 K i rgc. Stativet till vänster visar de kortmagasin som ingick i CPU 932. Det högra stativet är Censor 932 internminnesskåp, som i rgc bestyckades med en SI2 (adress) och fyra SI3 (data).

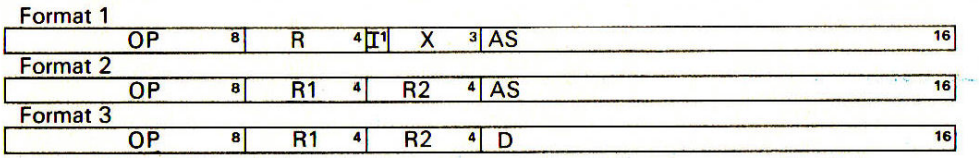


5 Censor 932 maskininstruktionslista

5.1 Introduktion.

Censor 932 maskininstruktionslista omfattade totalt 86 instruktioner. I tabell 1 och 2 nedan visas en sammanställning av dessa instruktioner och deras egenskaper. Beroende på typ av instruktion kunde halvords-, helords- eller dubbelordsoperander (16, 32 respektive 64 bitar) hanteras vid utförande av en instruktion.

Instruktionslängden var fast och omfattade 32 bitar i tre olika format (se bilden nedan).



Censor 932 instruktionsformat

Längden hos instruktionens operationskod (OP) var fast och omfattade 8 bitar.

Instruktioner enligt Format 1 använde fälten I, X och AS för beräkning av effektivadressen (E) till instruktionens minnesoperand (S). Fält R angav adressen till instruktionens register-operand (g). Fält I angav om adressen E skulle utgöra en indirekt adress (pointer) eller om adressen E skulle beräknas relativt aktuellt instruktionsräknarvärde (IC). Fält X angav om värdet i ett av Censor 932 7 indexregister skulle ingå i beräkningen av effektivadressen E.

Instruktioner enligt Format 2 använder fälten R1 och R2 för att ange instruktionens operander. R1 och R2 anger något av Censor 932 16 allmänna register G. Fält AS används för beräkning av effektivadressen E, som används för lagring av operationens resultat i S. Beräkning av E utförs genom addition av innehållet i fält AS med innehållet i aktuellt instruktionsräknarvärde IC (relativadressering).

Instruktioner enligt Format 3 använder fält R2 för att ange instruktionens registeroperand. Innehållet i fält D utgör operationens andra operand (direktoperand). Fält R1 anger plats för lagring av operationens resultat. Fält R1 och R2 anger något av Censor 932 16 allmänna register G.

Instruktionslistan enligt tabell 1 och 2 innehåller följande kolumner och beteckningar

Data Transfer Instructions:	Mnemonic	OP-code	Condition code	Format	Function	
INTERCHANGE HALFWORDS	IH	4E	–	1	s (E) ↔ g (R)	
INTERCHANGE WORDS	IW	5E	–	1	s (E) ↔ g (R)	1)
LOAD COMPLEMENT	LC	CD	1	1	−s (E) → G (R)	1)
LOAD DIRECT	LD	98	–	3	d → G (R1)	
LOAD DOUBLE WORD	LDW	68	–	1	s (E, E+2) → G (R, R+1)	1)
LOAD HALFWORD	LH	48	–	1	s (E) → G (R)	
LOAD NEGATIVE	LN	CB	1	1	−|s (E)| → G (R)	1)
LOAD I/O CONTROL	LOC	A9	–		s (E) → Status I/O (R)	1)

– Kolumn Mnemonic anger instruktionens benämning i form av ett antal alfabetiska tecken (lättare att komma ihåg än operationens hexadecimala op-kod)

– Kolumn OP-code anger instruktionens operationskod i hexadecimal notering.

– Kolumn Condition Code anger typ av resultatkod som operationen genererar. Följande typer av Condition Code förekommer i Censor 932:

Arithmetic setting	Logical setting	Comparison setting
0 result is zero	0 result is zero	0 operands equal
1 result is less than zero	1 result is other than zero	1 first operand is less than second operand
2 result is greater than zero		2 first operand is greater than second operand
3 overflow		

Arithmetic setting = Condition code 1
Logical setting = Condition Code 2
Comparison setting = Condition code 3

– Kolumn Format anger typ av instruktion.

– Kolumn Function beskriver operationens funktion.

Följande beteckningar användes för beskrivning av instruktionens funktion:

s(E) anger att innehållet i en minnesposition vars minnesadress är effektivadressen E
S(E) anger en minnesadress med adressen E
g(R) anger innehållet i ett generellt register med adressen R
G(R) anger ett allmänt register med adressen R
d anger innehållet i direktoperanden D
R anger i I/O-operationer adress till aktuell I/O-enhet. Standard I/O-enheter är Program Selector, Digital Clock, Paper Tape Reader och Punch samt Console Type Writer

Exempel: Beteckning i instruktionslista s(E) ↔ g(R). Innehållet i primärminnet på adressen E byter plats med innehållet det allmänna register G med adressen R.

5.2 Maskininstruktionslista.

INSTRUCTION FORMATS

Format 1

OP	R	I	X	AS
8	4	1	3	16

Format 2

OP	R1	R2	AS
8	4	4	16

Format 3

OP	R1	R2	D
8	4	4	16

Data Transfer Instructions:	Mnemonic	OP-code	Condition code	Format	Function
INTERCHANGE HALFWORDS	IH	4E	–	1	s (E) ↔ g (R)
INTERCHANGE WORDS	IW	5E	–	1	s (E) ↔ g (R) [1]
LOAD COMPLEMENT	LC	CD	1	1	−s (E) → G (R) [1]
LOAD DIRECT	LD	98	–	3	d → G (R1)
LOAD DOUBLE WORD	LDW	68	–	1	s (E, E+2) → G (R, R+1) [1]
LOAD HALFWORD	LH	48	–	1	s (E) → G (R)
LOAD NEGATIVE	LN	CB	1	1	−|s (E)| → G (R) [1]
LOAD I/O CONTROL	LOC	A9	–		s (E) → Status I/O (R) [1]
LOAD I/O UNIT	LOU	A8	–	1	s (E) → I/O (R) [1]
LOAD POSITIVE	LP	CA	1	1	|s (E)| → G (R) [1]
LOAD REGISTERS	LRS	B8	–	2	s (E, E+2Z) → G (R1−R2) [1] [4]
LOAD AND TEST	LT	CC	1	1	s (E) → G (R) [1]
LOAD WORD	LW	58	–	1	s (E) → G (R) [1]
RESET HALFWORD	RZH	4F	–	1	Zero → S (E)
RESET WORD	RZW	5F	–	1	Zero → S (E) [1]
STORE I/O CONTROL	SIC	A1	–		Status I/O (R) → S (E) [1]
STORE I/O UNIT	SIU	A0	–	1	I/O (R) → S (E) [1]
STORE REGISTERS	SRS	B0	–	2	g (R1−R2) → S (E,E+2Z) [1] [4]
STORE DOUBLE WORD	STDW	60	–	1	g (R, R+1) → S (E, E+2) [1]
STORE HALFWORD	STH	40	–	1	g (R) → S (E)
STORE WORD	STW	50	–	1	g (R) → S (E) [1]
Fixed-point Arithmetic Instructions:					
ADD DIRECT	AD	9A	1	3	g (R2)+d → G (R1)
ADD DOUBLE WORD	ADW	6A	1	1	g (R, R+1)+ s (E, E+2) → G (R, R+1) [1]
ADD HALFWORD	AH	4A	1	1	g (R)+s (E) → G (R)
ADD AND STORE	AS	1A	1	2	g (R1)+g (R2) → S (E) [1]
ADD TO STORE	ATS	2A	1	1	s (E)+g (R) → S (E) [1]
ADD WORD	AW	5A	1	1	g (R)+s (E) → G (R) [1]
COMPARE DIRECT	CD	99	2	3	g (R2)=d
COMPARE HALFWORD	CH	49	2	1	g (R)=s (E)
COMPARE WORD	CW	59	2	1	g (R)=s (E) [1]
DIVIDE DIRECT	DD	9D	–	3	g (R2)/d → G (R1) [2]
DIVIDE HALFWORD	DH	4D	–	1	g (R)/s (E) → G (R) [2]
DIVIDE AND STORE	DS	1D	–	2	g (R1, R1+1)/g (R2) → S (E, E+2) [1] [3]
DIVIDE WORD	DW	5D	–	1	g (R, R+1)/s (E) → G (R, R+1) [1] [3]
MULTIPLY DIRECT	MD	9C	–	3	g (R2)·d → G (R1)
MULTIPLY HALFWORD	MH	4C	–	1	g (R)·s (E) → G (R)
MULTIPLY AND STORE	MS	1C	–	2	g (R1)·g (R2) → S (E, E+2) [1]
MULTIPLY WORD	MW	5C	–	1	g (R)·s (E) → G (R, R+1) [1]
SUBTRACT DIRECT	SD	9B	1	3	g (R2)−d → G (R1)
SUBTRACT DOUBLE WORD	SDW	6B	1	1	g (R, R+1)−s (E, E+2) → G (R, R+1) [1]
SUBTRACT FROM STORE	SFS	2B	1	1	s (E)−g (R) → S (E) [1]
SUBTRACT HALFWORD	SH	4B	1	1	g (R)−s (E) → G (R)
SUBTRACT AND STORE	SS	1B	1	2	g (R1)−g (R2) → S (E) [1]
SUBTRACT WORD	SW	5B	1	1	g (R)−s (E) → G (R) [1]
(OPTION) Floating-point Arithmetic Instructions:					
ADD FLOATING	FA	7A	1	1	g (R, R+1)+s (E, E+2) → G (R, R+1) [1]
COMPARE FLOATING	FC	79	2	1	g (R, R+1)=s (E, E+2) [1]
DIVIDE FLOATING	FD	7D	–	1	g (R, R+1)/s (E, E+2) → G (R, R+1) [1]
MULTIPLY FLOATING	FM	7C	–	1	g (R, R+1)·s (E, E+2) → G (R, R+1) [1]
SUBTRACT FLOATING	FS	7B	1	1	g (R, R+1)−s (E, E+2) → G (R, R+1) [1]

[1] The effective address must be an even number
[2] The remainder is dropped
[3] The remainder is in S (E+2) and G (R+1) respectively
[4] Z=number of registers defined between R1 and R2

CONDITION CODE SETTING

Arithmetic setting
0 result is zero
1 result is less than zero
2 result is greater than zero
3 overflow

Logical setting
0 result is zero
1 result is other than zero

Comparison setting
0 operands equal
1 first operand is less than second operand
2 first operand is greater than second operand

Logic Instructions:	Mnemonic	OP-code	Condition code	Format	Function
COMPARE LOGICAL DIRECT	CLD	97	2	3	g (R2)=d
COMPARE LOGICAL HALFWORD	CLH	47	2	1	g (R)=s (E)
COMPARE LOGICAL WORD	CLW	57	2	1	g (R)=s (E) [1]
AND DIRECT	ND	94	3	3	g (R2) ∧d → G (R1) [1]
AND HALFWORD	NH	44	3	1	g (R) ∧s (E) → G (R)
AND AND STORE	NS	14	3	2	g (R1) ∧g (R2) → S (E)
AND TO STORE	NTS	24	3	1	s (E) ∧g (R) → S (E) [1]
AND WORD	NW	54	3	1	g (R) ∧s (E) → G (R) [1]
OR DIRECT	OD	95	3	3	g (R2) ∨d → G (R1)
OR HALFWORD	OH	45	3	1	g (R) ∨s (E) → G (R)
OR AND STORE	OS	15	3	2	g (R1) ∨g (R2) → S (E) [1]
OR TO STORE	OTS	25	3	1	s (E) ∨g (R) → S (E) [1]
OR WORD	OW	55	3	1	g (R) ∨s (E) → G (R) [1]
EXCLUSIVE OR DIRECT	XD	96	3	3	g (R2) ∨ d → G (R1)
EXCLUSIVE OR HALFWORD	XH	46	3	1	g (R) ∨ s (E) → G (R)
EXCLUSIVE OR AND STORE	XS	16	3	2	g (R1) ∨ g (R2) → S (E) [1]
EXCLUSIVE OR TO STORE	XTS	26	3	1	s (E) ∨ g (R) → S (E) [1]
EXCLUSIVE OR WORD	XW	56	3	1	g (R) ∨ s (E) → G (R) [1]
Shift Instructions:[2]					
ROTATE LEFT	RLS	85	–	1	g (R)$_0$ → G (R)$_{31}$
ROTATE LEFT DOUBLE	RLD	87	–	1	g (R)$_0$ → G (R+1)$_{31}$; g (R+1)$_0$ → G (R)$_{31}$
ROTATE RIGHT	RRS	84	–	1	g (R)$_{31}$ → G (R)$_0$
ROTATE RIGHT DOUBLE	RRD	86	–	1	g (R+1)$_{31}$ → G (R)$_0$; g (R)$_{31}$ → G (R+1)$_0$
SHIFT LEFT	SLA	89	1	1	g (R)$_{1-31}$ [3]
SHIFT LEFT DOUBLE	SLDA	8B	1	1	g (R)$_{1-31}$; g (R+1)$_{0-31}$ [3]
SHIFT LEFT LOGICAL	SLL	8D	–	1	g (R)$_{0-31}$
SHIFT LEFT DOUBLE LOGICAL	SLDL	8F	–	1	g (R)$_{0-31}$; g (R+1)$_{0-31}$
SHIFT RIGHT	SRA	88	1	1	g (R)$_{1-31}$ [4]
SHIFT RIGHT DOUBLE	SRDA	8A	1	1	g (R)$_{1-31}$; (R+1)$_{0-31}$ [4]
SHIFT RIGHT LOGICAL	SRL	8C	–	1	g (R)$_{0-31}$
SHIFT RIGHT DOUBLE LOGICAL	SRDL	8E	–	1	g (R)$_{0-31}$; (R+1)$_{0-31}$
Control Instructions:					
CHANGE PROGRAM	CP	C0	*)	1	s (P) → PCR; Zero → S (P)
EXECUTE	EX	C1	**)	1	s (E) → IR, E
JUMP ON CONDITION	JC	05	–	1	if r ∧ cc≠0; e → IC
JUMP ONE SUBTRACTED	JOS	02	–	1	g (R)−1 → G (R); e → IC if g (R) ≠0
JUMP TWO SUBTRACTED	JTS	03	–	1	g (R)−2 → G (R); e → IC if g (R) ≠0
JUMP ONE ADDED	JOA	04	–	1	g (R)+1 → G (R); e→ IC if g (R)≠0
JUMP TO SUBROUTINE	JS	01	–	1	ic+2 → G (R); e → IC
JUMP AND STOP	JSP	06	–	1	if r ∧ cc≠0; e → IC; stop [5]
LOAD STORAGE KEY	LSK	C3	–	1	s (E) → SKR [5]
LOAD PROGRAM CONTROL	LPC	C2	*)	1	s (E) → PCR [5]
NO OPERATION	NOP	00	–	1	–
STORE SWITCH REGISTER	STSR	08	–	1	sr → S (E)

[1] The effective address must be an even number
[2] Number of positions to be shifted is specified by the six least significant bits in the effective address
[3] The sign, g(R)$_0$ is not altered
[4] The sign, g(R)$_0$ is not altered but propagated through positions from the left.
[5] Forbidden instructions in normal programs
*) Condition code may change depending on the information loaded to PCR
**) Condition code may change depending on the instruction executed

Tabell 1 Tabell 2

6 Basprogramvara

6.1 Introduktion

I detta kap redovisas de basprogramvaruprodukter som utnyttjades vid skapande av de olika systemkonfigurationer av Censor 932 som förekommit i rgc. Med basprogramvara menas i detta fall dels de programutvecklingshjälpmedel i form av assembler, länkare, laddare, systemgenerator etc. som använts dels vid utveckling av applikationsprogramvaran för rgc och dels de olika versioner av Censor 932 operativsystem som ingått i de olika systemkonfigurationerna.

6.2 Bakgrund

När beslut om utveckling av maskinvarukonceptet Censor 900 togs saknades en plan för ett motsvarande programvarukoncept. Maskinkodning och absolutadressering var den metod som tillämpats vid utveckling av programvaran för Censor 120/220 och ansågs vara tillräcklig även för utveckling av programvara för Censor 932.

Kravet på programvara vid den tiden (mitten på 60-talet) var att den på bästa sätt skulle utnyttja maskinvarans egenskaper och vara så optimerad att det slutliga systemet kunde utgöra ett effektivt realtidssystem. Optimering innebar i detta fall att programvaran inte slösade på mikrosekunder. Bristen på minne krävde dessutom att programmen skulle vara extremt minnessnåla. Enda möjligheten att åstadkomma allt detta var att tillämpa maskinkodning och absolutadressering. Med den metoden kunde man bygga kod- och minnessnåla program och man hade full kontroll på vad som skapats (trodde man). Man visste att man utnyttjat maskinvarans kapacitet på bästa sätt.

Att programutvecklingsarbetet skulle vara effektivt och resultera i underhållbara programvaruprodukter diskuterades inte. En kompilator för något av den tidens högnivåspråk (Algol, Fortran etc.) var inte att tänka på. Dessa kompilatorer gav enligt gjorda undersökningar för dålig objektkod (exekveringsmässigt slöa program) samtidigt som minnesåtgången ökade. Det var inget att använda i ett realtidssystem. Programmeringsarbetet innebar normalt att först ritades ett flödesschema som grafiskt visade hur programmet skulle fungera. Därefter vidtog kodningen av programmet. Detta gjordes normalt så att man på ett linjerat papper skrev en maskininstruktion i hexadecimal form på varje rad. Raderna började med den adress instruktionen skulle ha i minnet. Om man orkade skrevs eventuellt en förklarande text. När bladet var slut fortsatte man med ett nytt. Om programmet var stort blev det till slut en rejäl hög med papper fyllda med hexdecimala tecken. När programmet var färdigkodat var det sedan att för hand med hjälp av maskinens operatörspanel bit för bit knappa in varje maskininstruktion. Därefter var det så dags för testning. Detta genomfördes under intensivt tryckande på operatörspanelens knappar. Det färdiga programmet stansades sedan ut på en pappersremsa (den tidens USB-minne). Allt efter som programsystemet växte i storlek ökade antalet remsor och det gällde att ha god ordning.

Mot den bakgrunden påbörjades utvecklingsarbetet av maskinvaran och i takt med att den färdigställdes uppstod behovet av programvara för verifiering av den framtagna maskinvaran. Kodning av testprogramvara påbörjades under slutet av 1967 med tillämpning av samma metoder som tillämpats för Censor 220, dvs. med programmering i hexadecimal absolutkod (maskinspråkskodning). Man fann snart att det hela inte var så enkelt. Speciellt det moderna adresseringssystemet i Censor 932 med relativ- och indirekt adressering gjorde det besvärligt

för programmerarna att tänka som man gjort vid utveckling av programvaran för Censor 120/220. Programmerarna önskade därför en möjlighet att skriva program med tillämpande av symbolisk adressering.

6.3 MIKRO

Ett "programmeringsspråk" specificerades. Språket fick namnet MIKRO och dess huvuduppgift var att ge programmeraren möjlighet att i en maskinkodad instruktion för Censor 932 utnyttja symbolisk adressering istället för C932 ”besvärliga” sätt att beräkna en absolutadress. MIKRO’s huvuduppgift blev i detta sammanhang att beräkna operandens slutliga absolutadress. Vid sidan om detta gav MIKRO programmeraren möjlighet att förse sin källkod med kommentarer (mycket viktigt i en maskinkodad miljö), deklarera värden för en adressymbol samt att förse källkoden med olika typer av styrkommandon för styrning av MIKRO’s hantering av källkodsremsan. Med MIKRO fick programmeraren även ett verktyg för grundläggande kontroll av källkoden genom att felrapporter kunde skrivas ut på konsolskrivare.

MIKRO kom att utgöra ”det första spadtaget” i Censor 932 framtida programutvecklingskoncept. MIKRO färdigställdes under 1968. MIKRO skrevs i absolutkod för Censor 220 dvs. program för Censor 932 utvecklades först på Censor 220 innan de laddades i Censor 932. Programmen ”stansades” in på en TeleType ansluten till Censor 220. Rättning och komplettering av koden gjordes i Censor 220 varefter Censor 220 producerade en hålremsa som var laddbar med Censor 932 IPL-funktion (Initial Program Loading) via dess remläsarutrustning, PTR eller med programmet ”Binary Tape Loader”. Detta var det första hjälpmedelsprogrammet i Censor 932. Det andra hjälpmedelsprogrammet var Binary Tape Puncher. Därefter kunde programkoden provköras i sin riktiga miljö med tillämpande av den då förhärskande metoden med stegning av programmet instruktion för instruktion via Censor 932 operatörspanel OCP.

6.4 Assembler

Även om MIKRO löste sin uppgift på ett utmärkt sätt så var man naturligtvis medveten om att detta var en nödlösning och att ett riktigt assemblerspråk måste utvecklas för Censor 932. Detta arbete påbörjades under 1968. Ursprungligen skrevs assemblern i MICRO-språket, men så snart den fungerade hjälpligt konverterades den med hjälp av ett särskilt konverteringsprogram till assemblerspråk och i fortsättningen kunde assemblern således "lyfta sig själv i håret". Den första versionen av assemblern blev klar under 1969. Till assemblern ingick även en ”linking-loader” (se nedan avsnitt 6.8).

I och med detta kunde MIKRO pensioneras och Censor 932 blev nu utvecklingsmiljö i stället för Censor 220. Med assemblern hade Censor 932 fått sitt första riktiga programutvecklingsverktyg. Med assemblern fick programmeraren möjlighet att tillämpa symbolisk adressering, namnsätta och definiera symboliska variabler samt att benämna den aktuella instruktionen med dess mnemoniska kod. Källkoden blev igenom detta mer läsbar än den tidigare stilen med hexadecimal instruktionskod. Andra viktiga egenskaper i C 932 assembler var att programvaran kunde delas upp i ”moduler” som kunde assembleras oberoende av övriga program till vilka referenser gjordes. Referenserna kunde vara lokala eller globala. I det första fallet löser assemblern själv upp referenserna helt och hållet med utnyttjande av Censor 932 relativadresseringsfunktion. I det andra fallet genererar assemblern ofullständigt ifyllda instruktioner jämte viss tilläggsinformation, som gör det möjligt för det separata länknings- och ladd-

ningsprogrammet att vid länkningen lösa upp de globala referenserna. Länkningen kunde därvid göras i godtycklig ordning.

Över tid utvecklades assemblern främst genom tillägg av makrofaciliteter. Den har förekommit i tre olika versioner:

- 1 pass, drift under OS
- 2 pass, drift under OS
- 1 pass, drift i TSS eller BOS

Med assemblern tillhands och Censor 932 remsutrustning (PTR och PTP) och konsolskrivare (IBM Selectric, den s.k. kulskrivmaskinen) kunde man nu (slutet 1969) påbörja utveckling av applikationsprogramvara. Emellertid hade man börjat inse att Censor 932 borde förses med ett operativsystem för att på rätt sätt utnyttja maskinens egenskaper. Man kunde inte fortsätta med den ”frekvensstyrda” jobhanteringen som tillämpats i Censor 220. Innan fullskalig applikationsprogrammering kunde starta så måste därför ett operativsystem definieras.

6.5 Operativsystem version 1

Under 1969 växte sig tankarna på ett operativsystem för Censor 932 allt starkare. Flera orsaker till detta fanns. För det första fanns det ett uttalat behov av en tidsstyrd jobbadministration som ersättning för den gamla klockpulsmetoden i Censor 220. Den nya maskinen hade en inbyggd programmerbar digitalklocka med vilken programsystemet kunde skaffa sig avbrottssignaler vid valbara tidslägen. För det andra var det nödvändigt att ha vissa standardiserade funktioner för I/0-hantering via skrivmaskin, remsläsare och stans. Till I/O-hanteringen hörde självklart att administrera brytsignalerna från respektive yttre enhet samt att möjliggöra multiprogrammering genom överlappning mellan bearbetning och in/ut-matning. För det tredje, hade den nya maskinen möjlighet att detektera vissa allvarligare fel och automatiskt anropa en felhanteringsrutin (maskinavbrott) vilken därigenom blev en självklar del av operativsystemet.

Det första operativsystemet skisserades under 1969 och implementerades under 1970. Det omfattade förutom ovan nämnda funktioner även en intern köhantering för styrning av jobb och I/O-beställningar, och en enkel funktion för dynamisk allokering av arbetsutrymme i kärnminnet.

Grundprincipen för detta operativsystem är att merparten av applikationsprogrammen exekveras under OS kontroll på en brytsignalnivå som kallas Main Processing Level (MPL). Drivrutiner för olika yttre enheter och systemfunktioner ligger på högre prioriteter. På en prioritet under MPL kan operatören beordra vissa hjälpprogram till exekvering, ett och ett i taget. Denna nivå kallas Batch Processing Level (BPL).

Under 1970-1971 vidareutvecklades OS så att det totala programsystemet kunde delas in i delar (partitioner), vilka kunde lagras på skivminne och automatiskt hämtas in av OS vid behov. Partitionerna prepareras i en separat systemgenereringsprocess, då alla referenser mellan programmen löses upp och en pointerarea skapas. Pointerarean måste ligga resident i kärnminnet under exekveringen och på fast plats. När partitionerna länkats, skrivs de ut på skivminnet som rena binära data. Partitionerna kunde därigenom laddas om igen genom enklast tänkbara I/O-operation och systemets overhead blev därigenom minimal.
Allt efter tillkomsten av nya yttre enheter (magnetband, radskrivare etc.) så kompletterades OS med drivrutiner för dessa enheter.

Följande namngivna varianter av OS version 1 har förekommit:

- OS 0.5 (1970) Användes i DBU 205. Motsvarar det som beskrivits ovan utom partitionering.
- OS 1.5 (1971) Motsvarar ovan beskrivna system
- OS 1.6 (1973) Motsvarar OS 1.5 med mindre interna förbättringar
- OS 1.7 (1974) Motsvarar OS 1.6 men anpassat för virkortsvarianten av Censor 932 V.

6.6 Operativsystem version 2

Under 1973 uppkom en serie krav på modifieringar av operativsystemet:

- en ny CPU med delvis ändrade gränsytor med I/O-enheter och maskinavbrottsfunktion
- kassettbandspelare tillkom som ny enhet (med denna kunde remsutrustningen pensioneras)
- behov av multipla MPL-nivåer för att göra det möjligt för högre prioriterade jobb att bryta jobb med lägre prioritet
- behov att kunna partitionera program även på bakgrundsnivå
- behov av symbolisk filhantering på skivminne och kassettband
- behov av en mer generell WAIT-funktion

Operativsystem version 1 hade dessutom ett antal olägenheter, t ex:

- ingen möjlighet att ändra längden på parameterlistor vid OS-anrop
- ingen möjlighet att skriva reentrant program innehållande OS-anrop med variabler i parameterlistorna
- Dessa saker tillsammans ledde till att ett nytt operativsystem definierades (1974). Målsättningen var:
- En mycket strikt programmeringsstandard skulle tillämpas för att uppnå säkerhet och likformighet
- Systemet skulle medge framtida utbyggnader
- Gränsytan mot användarprogrammen skulle standardiseras och dokumenteras väl

Under 1974-1975 implementerades detta operativsystem vilket kallades OS.2.

6.7 Hjälpprogram

Med hjälpprogram avses här sådana program som normalt används i samband med produktion av programsystem, programtestning och drift av programsystem.

6.8 LOADER

Det första viktigare hjälpprogram som togs fram för Censor 932 var laddnings- och länkningsprogrammet, den s.k. LOADERN. Den var nödvändig eftersom assemblern skapade en laddmodul med objektkod och ”oupplösta” referenser som måste lösas upp innan objektprogrammet kunde fungera på tänkt sätt. LOADERN sköter denna uppgift genom att ersätta de oupplösta referenserna med absolutadresser. LOADERN var ursprungligen gjord för att läsa laddmoduler från hålremsa, men har kompletterades senare för att kunna läsa även från skivminne. Vid sidan av sin huvuduppgift skriver LOADERN ut en lista över de laddade pro-

grammen och pointrarna på konsolskrivmaskinen. Med hjälp av Loadern kunde program laddas in tillfälligt i systemet, t ex för testning eller programproduktion. En förutsättning var dock att Loadern utgjorde en del av ett genererat programsystem, lagrat i IPL-laddningsbar form (IPL = Initial Program Loading, d.v.s. hardware funktion för laddning av program till helt tom maskin).

6.9 LIST

Det normala sättet att framställa källkodsremsor var från början genom stansning på telexskrivare. Denna ersattes senare med Teletype ASR33. I båda fallen var det tidsödande att få en snygg formatering av källkodslistorna. Därför togs ett listningsprogram fram, först i en variant för listning på konsolskrivmaskin eller hålremsa, och senare för utmatning till radskrivare.

6.10 EDIT

Det var mycket tidsödande och besvärligt att ändra källkodsremsor. Därför konstruerades ett generellt editeringsprogram, vilket läste källkod från remsa till en buffert i kärnminnet. Operatören kunde sedan genom olika direktiv via konsolskrivmaskinen beordra valfria ändringar i koden. Till sist kunde han beordra utmatning av det editerade resultatet på hålremsa. Detta program togs i bruk i början av 1970.

6.11 SSG

Tidigt kom kravet att man skulle kunna välja bort inte önskade funktioner i operativsystemet på ett enkelt sätt. Detta arrangerades så att man på lämpliga ställen i operativsystemets källkod lade in s.k. funktionsparenteser. Med hjälp av programmet Source Code System Generation (SSG) kunde den kod som fanns mellan dessa parenteser tas bort. SSG kan således betraktas som en specialversion av editeringsprogrammet EDIT.

I och med tillkomsten av OS version 2 (OS.2) (1974) pensionerades SSG. I detta operativsystem finns inga som helst funktionsparenteser. De möjligheter som där finns till val av OS-funktioner grundar sig i stället på val av hela källkodsmoduler vid assembleringen av den centrala OS-laddmodulen samt genom val av laddmoduler vid genereringen.

6.12 MERGE

Operativsystemet för Censor 932 var ursprungligen skrivet som ett stort antal separata program, vilket var mycket lämpligt ur uttestningssynpunkt och underhållssynpunkt, men mindre önskvärt ur den synpunkten att det skapade en stor pointerarea. Eftersom minne ansågs dyrbart, tog man fram ett MERGE-program som kunde slå ihop flera program (på källkodsnivå) till ett enda program. I senare utgåvor av den första OS-generationen (OS 1.6) och i den första utgåvan av OS-2, har operativsystemet från början utformats som ett enda program. Behovet av MERGE har därigenom upphört och MERGE har därför pensionerats på samma sätt som SSG.

6.13 OSG

OSG bestod väsentligen av Loadern. Med hjälp av OSG kunde alla de laddmoduler som skulle ingå i ett programsystem länkas ihop, partitioner skapas och olika tabeller och konstanter definieras. Output från OSG var ett programsystem i objektformat, lagrat i en form som tillät inläsning med hjälp av IPL.

OSG arbetade i princip i dialog med operatören. Vid rutinmässig generering av större system blev detta emellertid ineffektivt och en möjlighet fanns därför att lagra operatörens inmatningar i en fil på skivminne eller hålremsa för snabbare generering.

6.14 TRACE

Uttestning av program har alltid varit besvärligt. Från början fanns det bara en enda metod, nämligen stegning av programmet via maskinens kontrollpanel. För att underlätta detta arbete, konstruerades under 1969 ett spårningsprogram, kallat TRACE. Med dess hjälp kunde ett program genomlöpas i enlighet med operatörens önskemål och programmets beteende, inkluderande förändringar i variabler o.d. kunde loggas på en yttre enhet som exempelvis Censor 932 konsolskrivmaskin.

Den ursprungliga versionen av TRACE präglades naturligtvis starkt av den tid i vilken den föddes. Detta innebar dels att gränsytan mot användaren inte var idealisk, dels också att programmets interna utformning var mindre lyckad. Dessa brister ledde under 1975 till att ett helt nytt och avsevärt förbättrat TRACE-program togs fram. Detta anpassades för drift i TSS-miljö (Time Sharing System).

6.15 TSS

Tanken att man skulle använda Alfaskopet som terminal i ett tidsdelningssystem för programproduktion föddes samtidigt med Alfaskopet i slutet av 60-talet. Under 1970-1972 utvecklades ett sådant system (TSS) på Censor 932. Systemet omfattade följande funktioner:

- in- och utmatning via hålremsa och radskrivare
- lagring av data i filer i bibliotek på skivminne
- editeringsfunktion
- assembler
- interpretativ modultestmonitor (TRACE)

Systemet togs i drift i slutet av 1972. Med hjälp av detta system blev det möjligt att stansa, editera, assemblera och testa programmoduler utan att resa sig från terminalen. Detta innebar ett mycket stort steg i utvecklingen mot en rationellare programproduktion än vad som varit möjligt att genomföra med befintliga hjälpmedel.

7 Censor 932V (”virkortscensorn”)

7.1 Bakgrund

Censor 932K utnyttjades från dess introduktionsår 1968 i en rad olika systemtillämpningar. Över lag var systemkonceptet tekniskt väl anpassat för att möta krav på olika typer av systemtillämpningar. En besvärande faktor var dock den höga tillverkningskostnaden. Produktionstekniken gav dessutom upphov till kvalitetsproblem. Eftersom samma teknologi användes av det ”priskänsliga” produktområdet Alfaskop ingrep dåvarande ägarbolag ITT med en direkt uppmaning till SRT att sänka tillverkningskostnaden drastiskt.

En intensiv mötesverksamhet startades. Efter ett stort antal konstruktionsgenomgångar föddes det s.k. virkortet (se beskrivning i avsnitt Elektrisk uppbyggnad nedan). En mycket flexibel lösning som passade för både enstycks- och serietillverkning. Dessutom fungerade virkortet, med sin separata spänningsstabilisator, utmärkt tillsammans med det befintliga systemet för strömförsörjning. Med virkortet hade SRT nu ett kostnadsmässigt bättre alternativ till den dyra ”kortlådan”. Nu var det till och med ekonomiskt försvarbart att serieproducera ”prototyper”. Nödvändiga ändringar kunde införas i efterhand. Produktområde Alfaskop var först på plan att utnyttja virkortet.

Genom den utveckling som skedde inom mikrokretsområdet sedan introduktionsåret 1968 fanns det ett par år in på 1970-talet nya typer av komponenter (MSI-kretsar, PROM etc.). Om dessa kunde användas vid uppbyggnad av Censor 932 skulle det vara möjligt att halvera antalet ”kortlådor” i Censor 932K. Det säger sig själv att detta skulle innebära en drastisk sänkning av tillverkningskostnaden. Med virkortstekniken skulle dessutom en ny CPU kunna rymmas i en dubbel ISEP-ram istället för ett helt skåp.

Eftersom kärnminnestekniken under samma tid genomgått samma kraftiga utveckling som mikrokretstekniken fanns det nu snabbare (750 ns) kärnminnesmoduler som samtidigt var mycket mindre utrymmeskrävande än de tidigare använda minnena. Sammantaget skulle man nu kunna packa in en CPU 932 med 32 kord minne och en extra ramplats i ett stativ. Jämfört med tekniken som använts för Censor 932K fick man nu en komplett dator i ett skåp (!).

1973 var tiden inne för Stansaab (bildat 1971) att ta tillvara den tekniska utvecklingen i form av nya komponenter och ny produktionsteknik vid lösandet av kraven på en kostnadsreducerad CPU. Konstruktionen av Censor 932V (”virkortscensorn”) påbörjades. Censor 932V fick senare produktnamnet 9103 när Stansaab hade fått ordning på sina produktprogram. Censor 932V kom att tillhöra Stansaab’s produktfamilj Censor 900 Computer System.

7.2 Censor 932V, introduktion

Målet med Censor 932V var primärt en ur tillverkningssynpunkt kostnadsoptimerad produkt. Eftersom processorn skulle kunna användas även för uppgradering av redan levererade system som exempelvis rgc var det självklart att den nya produkten blev fullständigt programkompatibel med Censor 932K. På detta sätt skulle den också kunna köra den nya versionen av operativsystemet (OS 2) som var under konstruktion inom Stanssab. Kompatibilitetskravet innebar dessutom att den nya Censorn måste kunna hantera den remsutrustning (PTP/PTR) som ingick i Censor 932K.

Av kostnadsskäl ersattes den tidigare använda konsolskrivaren IBM Selectric ersattes med den betydligt billigare konsolskrivaren Teletype (TTY).

Tillkomsten av ”virkortsCensorn” innebar dessutom att minnesdatabussen och I/O-bussen Channel Selector (CS) i Censor 932K databuss ersattes av det nya databussystemet Censor 900 bussystem (den s.k. ”TTL-bussen”). Bytet var en direkt anpassning till den nya DIP-baserade virkortstekniken. Bussgränsytan utgjordes nu enbart av integrerade kretsar.

Med de metoder och verktyg för utveckling och test av enskilda program och programsystem som tillämpats för Censor 932K hade behovet av en OCP (Operators Consol Panel) av den typ som ingick i CPU 932K över tid minskat. För CPU 932V utvecklades därför en enklare panel till stöd för främst maskinvarunära test och övervakning. Övrig styrning och övervakning utfördes istället alltmer via den nya konsolskrivaren TTY.

7.3 Elektromekanisk uppbyggnad

7.3.1 Allmänt

Det avgjort viktigaste beslutet i kostnadsreduceringsprojektet var att CPU 932V skulle byggas upp på virkort med integrerade kretsar och virkortsramar av typ ISEP (International Standard Equipment Practice). I likhet med Censor 932K användes vild kabeldragning enligt konceptet ”wire wrap” som förbindningssystem. I de fall kretslösningar måste baseras på diskreta komponenter (motstånd, transistorer etc.) förbands dessa komponenter på folieplatta.

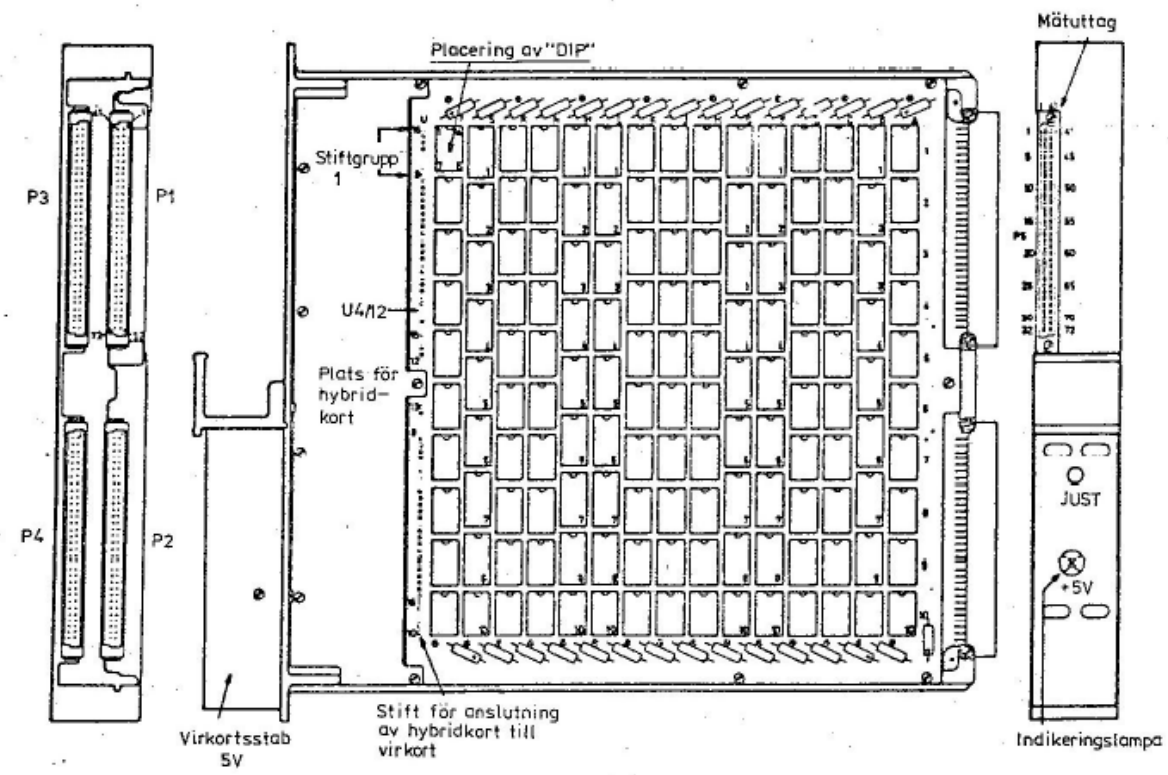
7.3.2 Virkortet

Virkortets utförande visas i bilden nedan. Kortet består av en metallram i vilken ett foliekort är monterat. På foliekortet monteras hållare för integrerade kretsar i DIP-utförande (Dual in-line Package). Hållarna, som är försedda med förbindningsstift för ”wire wrap” monteras på folieplattans ena sida. Hållarnas förbindningsstift är sedan åtkomliga på folieplattans baksida. Upp till 96 14-stifts och 48 16-stifts hållare kan monteras på en folieplatta. DIP-hållarna bestyckas med integrerade kretsar och kretsarna signalstift kopplas samman med ”virtråd”.

I de fall som den elektriska lösningen kräver att diskreta komponenter måste utnyttjas används den s.k. hybriplattan på vilken komponenterna monteras. Hybridplattan är foliekort som monteras bakom virkortets handtag och kan innehålla både diskreta komponenter och integrerade kretsar. Hybridplattan förbinds sedan med kretsarna på virkortsplattan med virtråd.

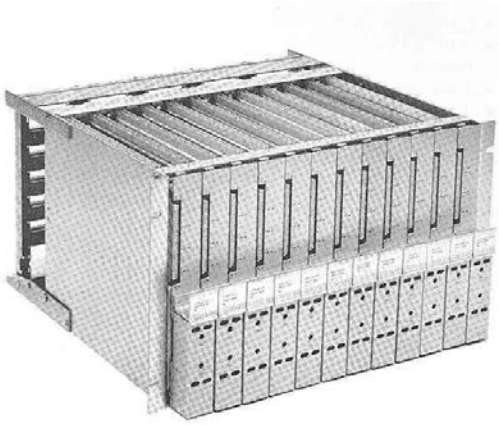
På virkortets handtag monteras virkortets strömförsörjningsenhet, en stabilisator för + 5V. Stabilisatorn (stabben) matades med råkraft från det kraftmatningssystem som togs i bruk vid konstruktionen av Censor 932K.

Virkortets anslutning till virkortsramen omfattar fyra stycken 64-poliga kontaktdon. På virkortets framsida finns ett 64-poligt kontaktdon för mätuttag.



7.3.3 Virkortsramen

Virkorten monterades i ramar (se bild nedan) som hade plats för upp till tolv stycken virkort. Ramens bakplan bestod av ett kontaktfält innehållande 12 x 4 x 64 = 3072 virstift. Virkortsramen monterades i stativ av ISEP-typ. I stativet för Censor 932V ingick 2 stycken ramar. CPU 932V omfattade 1 ram inklusive anpassare för remsutrustning och konsolterminal.



7.3.4 Elektronik

Censor 932V baserades helt på den tidens (1973) mest moderna (state of the art) integrerade TTL-kretsar i DIP-utförande (Dual Inline Package). Allt från 14 till 24-pinnars kretsar användes. Kretsarna placerades i DIP-hållare på virkortet.

Med de nya MSI (Medium Scale Integration) kretsarna fick man nu exempelvis en komplett adder för fyra bitar i en (1) krets. Jämfört med uppbyggnadstekniken i Censor 932K omfat-

tade en sådan adder ca 15 stycken kretsar. Den höga packningstätheten innebar dessutom att det i jämförelse med CPU 932K blev betydligt enklare att få kontroll på det tidigare stora problemet med styrsignalers tidsfördröjningar. Genom det mindre antalet kretsar som användes för att distribuera en styrsignal kunde man vara säker på tidsfördröjningen alltid låg inom givna tidsmarginaler. I CPU 932K var detta inte alltid fallet. Resultat av den nya kretstekniken blev därför väsentligt högre driftsäkerhet.

7.3.5 Mikroprogrammering

Konstruktionen av exempelvis en CPU förenklas väsentligt om man på ett enkelt sätt kan generera de styrsignaler som behövs för utförande av de olika typer av mikrooperationer en komplex funktion är sammansatt av. Vid slutet av 60-talet var det normalt att generera styrsignaler antingen med hjälp av invecklade logiska grindnät eller att lagra dessa styrsignaler i någon form av minne baserat på dioder. I avsaknad av lämpliga minneskomponenter fick man i C 932K generera styrsignalerna med hjälp av invecklade logiska grindnät, med varierande tidsfördröjningar. I C220 genererades styrsignaler i diodmatriser styrda med pluggar för ”hand”. Packningstätheten blev inte särskilt hög.

Under början av 70-talet började olika typer av s.k. Programmable Read Only Memory (PROM) lanseras på marknaden. Ett PROM var en högintegrerad ”diodmatris” som kunde programmeras i ”fält” med hjälp av en datorstyrd PROM-programmerare. När beslutet togs att starta utvecklingen av en kostnadsreducerad CPU 932 kunde marknaden erbjuda prestandamässigt mycket snabba (kort accesstid) PROM-varianter som samtidigt var packningsmässigt kraftfulla.

Med denna möjlighet var beslutet givet. CPU 932V skulle generera alla sina styrsignaler genom mikroprogrammering. Varje komplex funktion (exempelvis en maskininstruktion) som skulle utföras i CPU motsvarade en sekvens av mikroprograminstruktioner lagrade i ett PROM. Varje mikroprograminstruktion innehöll koder för de styrsignaler som behövdes för att utföra en mikrooperation. Styrlogiken i CPU försågs med ett s.k. mikroprogramverk som likt en dator läste mikroprograminstruktioner i PROM och utifrån mikroinstruktionens innehåll genererade de styrsignaler som krävdes för att genomföra den aktuella mikrooperationen. Med denna mikroprogrammeringsmöjlighet som blev införd i CPU 932V blev det nu bl.a. enkelt att i efterhand utöka datorns instruktionslista. Ett exempel på detta var införandet av flyttalsinstruktioner i CPU 932V. Flyttalsinstruktioner utnyttjades dock inte i rgc.

7.3.6 Förbindningssystem

Samma wire wrap-teknik som användes för Censor 932K användes även för Censor 932V.

7.4 Kraftmatningssystem

Kraftmatningssystemet som använts för Censor 932K användes utan ändring för Censor 932V. Se motsvarande avsnitt för Censor 932K.

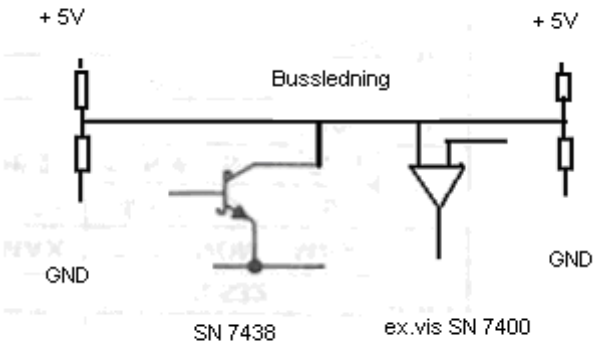
7.5 Censor 900 bussystem

7.5.1 Bakgrund

Med beslutet att konstruera CPU 932V på virkort minskade möjligheten att använda diskreta komponenter. Det mest effektiva blev nu att använda integrerade kretsar i kretslösningarna. Detta innebar att den på diskreta komponenter baserade databussen (”kniptångsbussen”) i Censor 932K inte på ett ekonomiskt skulle kunna gå att använda i Censor 932V. Man måste därför finna en busslösning som var helt baserad på integrerade kretsar. Vid valet av busslösning spelade även det faktum att behovet av en ”lång buss” hade minskat genom att den nya byggtekniken innebar att ett komplett systems fysiska utbredning kunde minskas radikalt jämfört med kassettcensorn. Detta innebar i sin tur att den nya bussens längd skulle kunna begränsas till 3 m. Efter en lång rad tester specificerades den s.k. TTL-bussen och konceptet fick namnet ”Censor 900 databus system”.

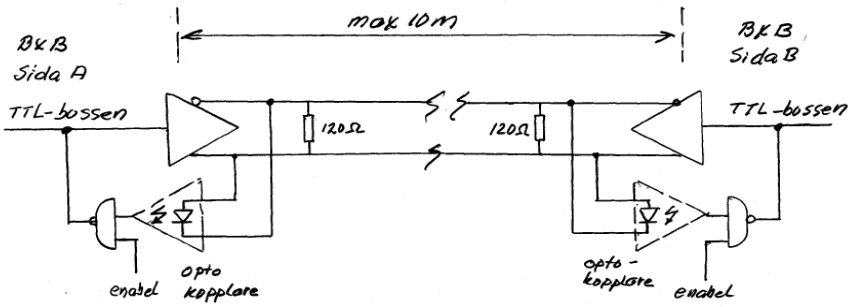
7.5.2 Elektrisk uppbyggnad

Bussens huvudkomponent utgjordes av TTL-kretsen SN 7438 (se bilden nedan). Kretsen var en 2-ingångars grind med öppen kollektor och användes som bussdrivare. Varje bussledning avslutades i sina båda ändar med anpassningsmotstånd via vilka bussledningen strömförsörjdes. Som mottagare användes grindar av typen SN 7400. För att minska överhörningen mellan bussystemets bussledningar tvinnades varje bussledning med sin returledning



Komponenter i bussen

Bussledningen enligt bilden ovan var dimensionerad för en största längd om 3 m. I de fall en systemkonfiguration krävde större längd än 3 m mellan olika bussegment kopplades dessa segment samman med hjälp av en balanserad transmissionslänk. Varje bussposition på TTL-bussen kopplades samman via ett interface enligt bilden nedan. Funktionellt är länken dubbelriktad. Optokopplare används för att få så god isolation som möjligt mellan länkens båda sidor. Länkens elektriska komponenter ingick i virkortsenheten Bus Exchange Balanced (BXB).

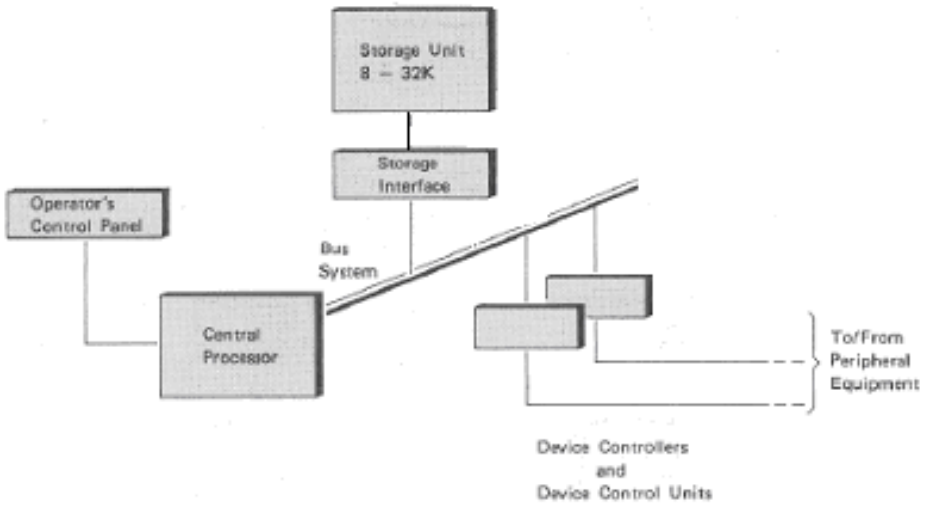


Balanserad transmissionslänk

7.5.3 Censor 900 Bus System

Konceptmässigt är 900-bussen en replik av databusskonceptet för Censor 932K. Dock övergavs den särskilda I/O-bussen Channel Selektor. Alla enheter blev ”minnesmappade” d.v.s man angav ett speciellt minnesområde för I/O-enheter såsom anpassare för remsläsare och stans (PTE), skrivare (TPW), bandspelare (DCRC), skivminne (CI) osv. Med minnesmapp kunde man även anropa andra bussar via Bus Exchange Unit. BIXA och BXB ingick i rgc bussystem. Bussystemet kontrollerades av en enhet Storage Interface, SI, som styrde och kontrollerade bussaccesser. Detta innebar att 900-bussen kunde hantera både DMA-enheter och Program I/O-enheter.

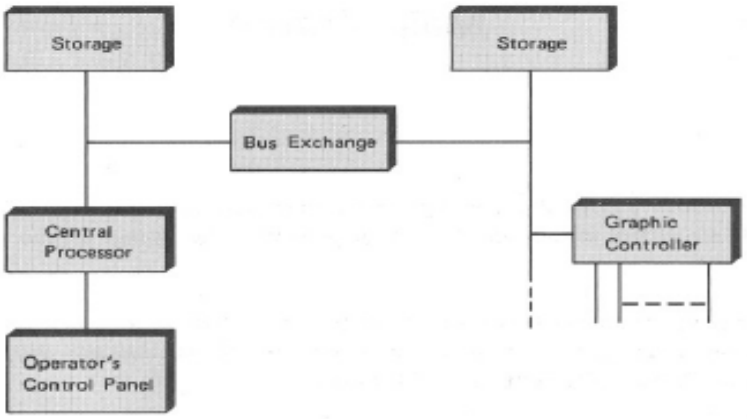
I bilden nedan visas uppbyggnaden av en typisk busskonfiguration. CPU och Device Controllers är ansluts direkt till bussen. Minnesmoduler ansluts via enheten Storage Interface. Inkluderat i enheten Storage Interface finns enheten för tilldelning och kontroll av bussaccesser (jfr enheten AM i bussystemet för Censor 932K). Funktionellt konkurrerar CPU och Device Controllers med varandra om bussaccesser. Konsekvensen blir att CPU inte alltid kan arbeta med högsta prestanda.



Typisk busskonfiguration

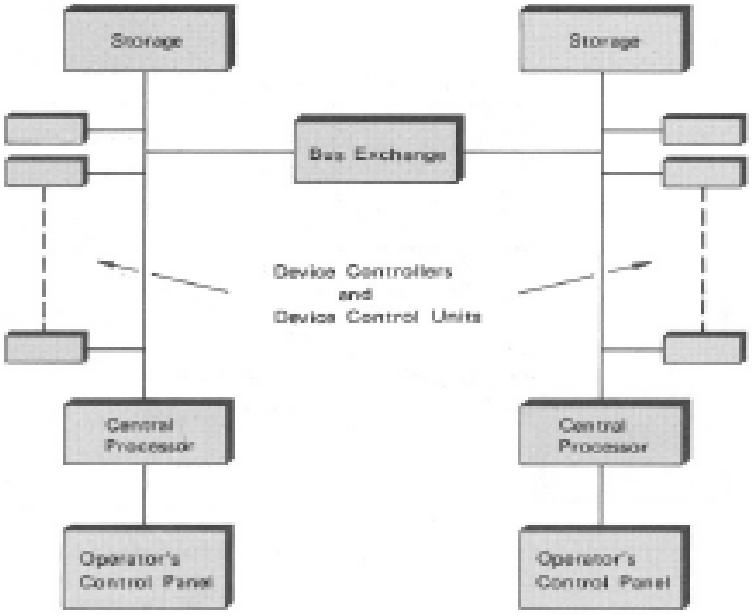
Med ett bussystem enl. bilden nedan har man med enheten Bus Exchange (BX) skapat två segment, ett internt för CPU och ett externt för Device Controllers (i bilden en Graphic Controller). Funktionellt kan nu CPU arbeta med full kapacitet på sitt segment samtidigt som

Graphic Controller arbetar med full kapacitet på sitt segment. I de fall CPU adresserar det externa minnet gör den det i konkurrens med Graphic Controller. Priset är en viss prestanda-förlust. Med enheten BX kunde man genom sektionering av ett bussystem på ett effektivt sätt öka systemprestanda jämfört med ett osektionerat system.



Exempel på bussegmentering

I bilden nedan ges ett annat exempel på möjligheten med sektionering av ett bussystem. I detta fall har man skapat ett dubbeldatorsystem.



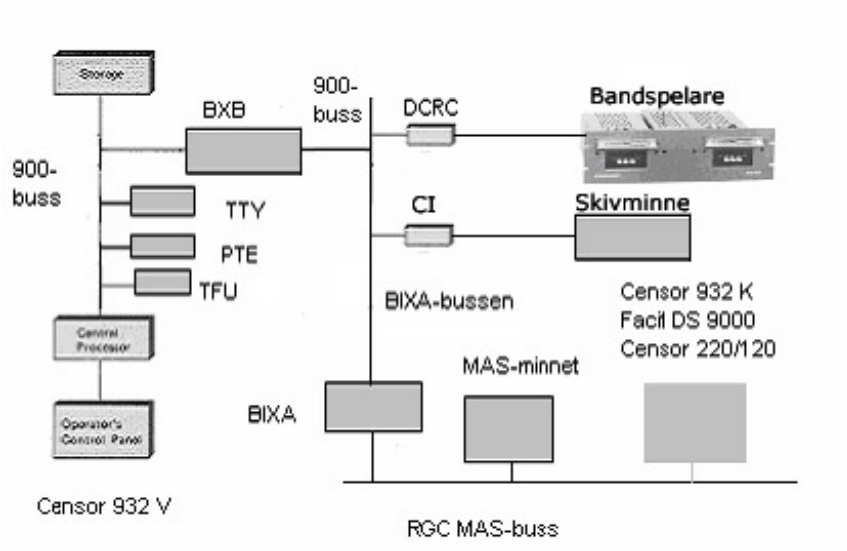
Exempel på sektionering

7.5.4 Censor 900 Bussystem i rgc

I bilden nedan visas hur Censor 932V integrerades med befintlig utrustning i rgc. Bussystemet består av två stycken TTL-bussegment. I det ena segmentet finns anpassare för kassettband-spelare (DCRC) och Singerdisk anpassat till MAS-bussen via den för rgc speciellt konstruerade busväxeln BIXA (installerades i etapp 3). I det andra segmentet ingår Censor 932V med kärnminne och anpassare för konsolskrivare och remsutrustning samt den trigonometriska beräkningsenheten TFU (installerades i etapp 4). Eftersom det fysiska avståndet mellan stati-

vet för Censor 932V och stativet där BIXA ingick (BG-stativet) överskred 3 m användes bussväxelenheten BXB (se avsnitt 7.5.2) för sammankoppling av de båda 900-segmenten.

Rgc MAS-buss ”skapades” när rgc uppgraderades med Censor 932K och fungerade som sammankopplingsbuss för Censor 932K, Facit DS 9000 och Censor 120/220.



Censor 900 Bussystem i rgc

7.6 Teknisk beskrivning

7.6.1 Systemöversikt

CPU 932V omfattade följande 10 virkort:

- Styrenhet CU1-CU4 (4 kort)
- Aritmetisk logisk enhet AU1-AU4 (4 kort)
- Programväljare del 1 och digitalklocka PDC (1 kort)
- Programväljare del 2, adresslogik, I/O-styrning PDA (1 kort)

Korten placerades i en virkortsram. De återstående 2 kortplatserna i ramen var förberedda för anpassningsenhet för remsutrustning (Paper Tape Equipment, PTE) och anpassare för konsolterminal Teletype (TTY) samt för optionen Floating Point.

I linje med konstruktionsförutsättningarna för CPU 932V gjordes den fullständigt maskininstruktionskompatibel med Censor 932K. Detta innebar samtidigt att CPU 932V försågs med exakt den registeruppsättning som beskrivs i avsnitt 4.2, Censor 932 systemuppbyggnad. Den stora skillnaden mellan CPU 932V och CPU 932K låg i styrenhetens uppbyggnad. I V-maskinen baserades styrsignalgenereringen på mikroprogrammering och PROM-lagrade styrsignaler. I K-maskinen skapades styrsignaler med logiska grindnät.

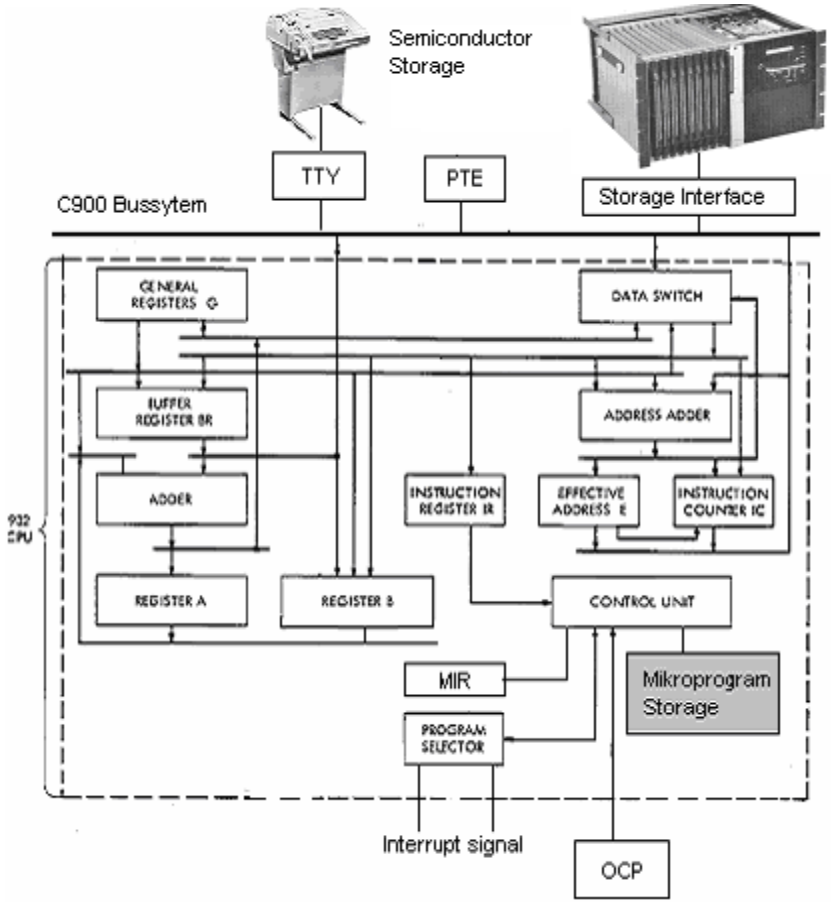
En konstruktionsfiness värd att nämna var den s.k. Instruction Look Ahead-funktionen (ILA). Den kan sägas vara ett första steg mot att implementera en ”instruction pipe” d.v.s. en funktion som kan hantera flera instruktioner samtidigt i avsikt att öka en centralenhets prestanda. I detta fall omfattade ”pipen” dock endast en position. Med ILA hämtades en ny instruktion

innan den föregående slutbehandlats. På så sätt kunde centralenheten utnyttja minnet maximalt. Fortare kunde det inte gå. Det blev minnets cykeltid som avgjorde hur fort processorn kunde arbeta.

7.6.2 Systemblockschema Censor 932V

Bilden nedan visar Censor 932V systemblockschema med de till CPU anslutna enheterna:

- Konsolskrivare (TTY)
- Flyttbart minne
- Primärminne
- Operators Control Panel (OCP)



Censor 932V systemblockschema

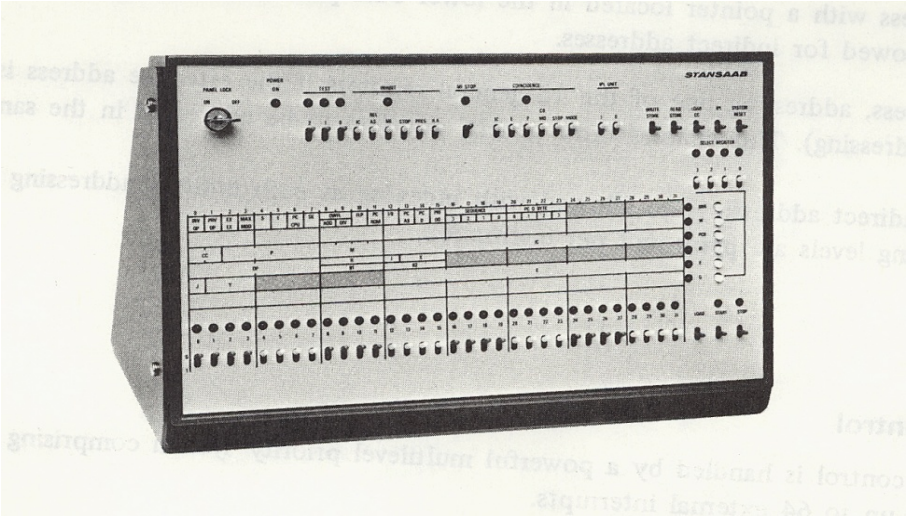
Konsolskrivare (TTY)
Som konsolskrivmaskin användes en TTY (TeleType) för Censor 932V. Billigare och enklare då denna till skillnad från den parallellanslutna konsolskrivaren (IBM Selectric, ”kulan”) hade ett allmänt använt serieinterface. Med detta blev man fri att använda i princip vilken ASCII-terminal som helst som konsol. I rgc utnyttjades detta så att man till slut använde en PC som konsol.

Flyttbart minne
Med anpassaren för remsläsare/stans (PTE) återanvändes för en tid det ”remssystem” som använts i Censor 932K. I takt med att nya typer av flyttbara minnesenheter blev tillgängliga ersattes remsläsare/stans med en kassettbandspelare (Digital Cartridge Recorder) och skivminne.

Primärminne
I Censor 932V användes halvledarminne som datorns primärminne. Sedan introduktionen av Censor 932K skedde en mycket kraftig utveckling av minnestekniken. Under inledningen av 70-talet förbättrades kärnminnestekniken kontinuerligt men mot mitten av 70-talet tycktes det dock som att tekniken nått vägs ände. Det var nödvändigt för dåvarande Stansaab att finna lämplig ersättare för kärnminnet. Ett omfattande utvecklingsprogram startade därför i avsikt att utveckla en kärnminnesersättare som var baserad på den tidens kompakta halvledarminnen av N-MOS-typ. Resultatet blev en komplett minnesmodul innehållande minneskort, batteri-backup och refreshlogik. Som extra tillbehör ingick även en felkorrigeringsenhet som kunde korrigera en-bitsfel. Halvledarminnet var anslutningsmässigt kompatibel med de tidigare använda kärnminnena.

I blockschemat ovan visas den halvledarminnesmodul som användes i rgc. Mekaniskt var det en ISEP-ram med 32 kord minne och kraftenhet. Motsvarande minne i Censor 932K utgjordes av 8 minnesmoduler om 4 kord vardera samt fyra kraftaggregat allt packat i två stativ. Med den nya tekniken ockuperade kärnminnet i V-datorn endast ett fjärdedels stativ.
.

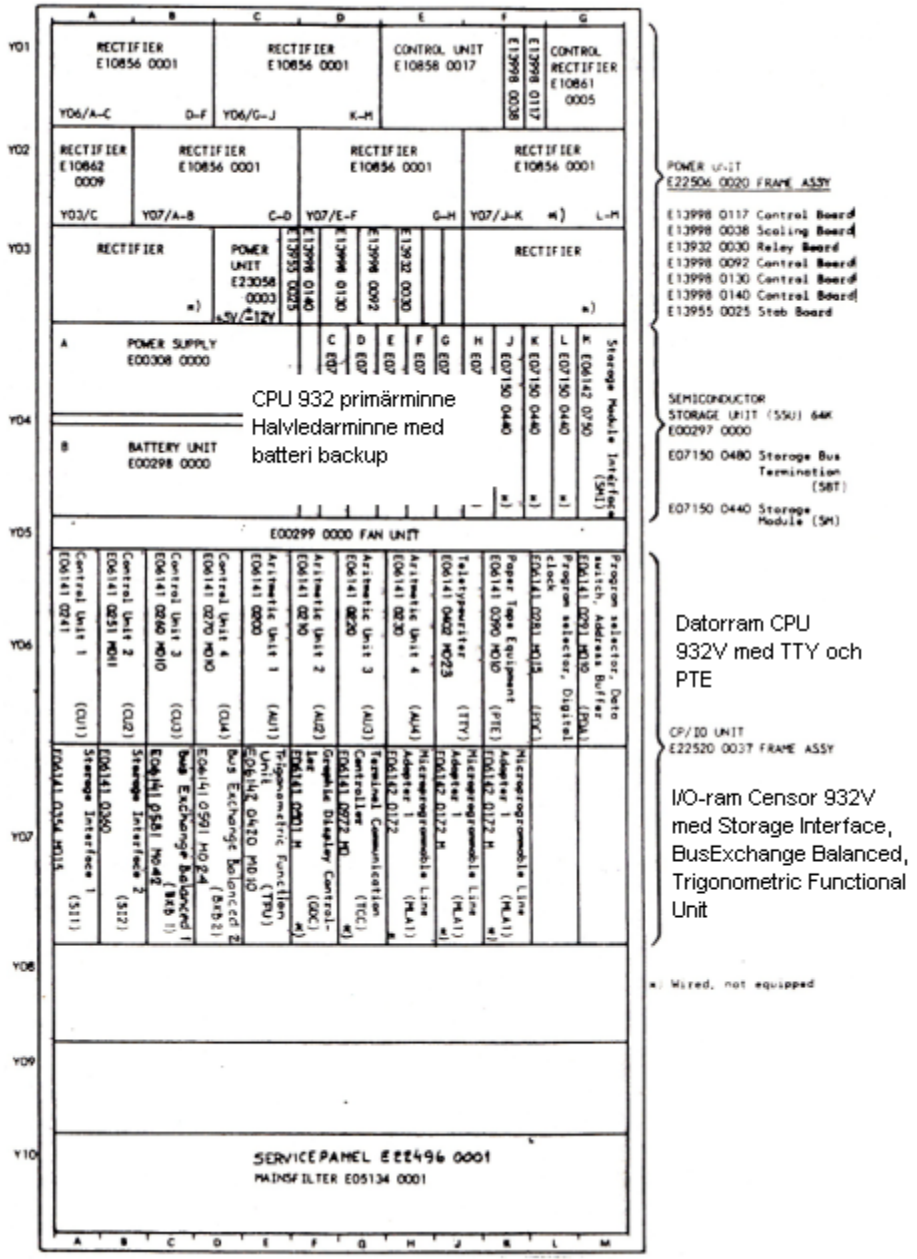
Operators Control Panel (OCP)
I CPU 932V ingick som seden var vid den tiden en Operators Control Panel (OCP) (se bilden nedan). Den hade samma funktioner som OCP för CPU 932K men hade inte en lika omfattande presentationsfunktion. Mekaniskt var den möjlig att placera både på bord och på vägg.
Med de metoder och verktyg för utveckling och test av enskilda program och programsystem sedan introduktionen av Censor 932K hade behovet av en OCP minskat. Dock behövdes fortfarande någon form av maskinnära manöver- och övervakningsorgan. För CPU 932V utvecklades därför en enklare panel. För presentation av registerinnehåll användes exempelvis endast en lamprad (röda lysdioder). Med omkopplare på OCP valdes sedan vilket register som skulle presenteras. Ett krets-och kabelbesparande serieinterface användes som kommunikationslänk mellan CPU och OCP. Trots de förenklingar som gjordes var det möjligt att utföra samma funktioner som på den mer omfattande panelen för Censor 932K.



Censor 932V operatörspanel

7.6.3 Stativkonfiguration

C 932V stativkonfiguration framgår av nedanstående bild. Stativet är bestyckat med det kärnminne som visas i punkt 7.6.2.


RECTIFIER E10856 0001
RECTIFIER E10856 0001
CONTROL UNIT E10858 0017
E13998 0038
E13998 0117
CONTROL RECTIFIER E10861 0005
RECTIFIER E10862 0009
RECTIFIER E10856 0001
RECTIFIER E10856 0001
RECTIFIER E10856 0001
RECTIFIER
POWER UNIT E23058 0003 +5V/-12V
E13955 0025
E13998 0140
E13998 0130
E13998 0092
E13932 0030
RECTIFIER
POWER SUPPLY E00308 0000
BATTERY UNIT E00298 0000
CPU 932 primärminne
Halvledarminne med batteri backup
E00299 0000 FAN UNIT
Control Unit 1 E06141 0241 (CU1)
Control Unit 2 E06141 0251 M01 (CU2)
Control Unit 3 E06141 0260 M010 (CU3)
Control Unit 4 E06141 0270 M010 (CU4)
Arithmetic Unit 1 E06141 0200 (AU1)
Arithmetic Unit 2 E06141 0210 (AU2)
Arithmetic Unit 3 E06141 0220 (AU3)
Arithmetic Unit 4 E06141 0230 (AU4)
Teletypewriter E06141 0402 M023 (TTY)
Paper Tape Equipment E06141 0390 M010 (PTE)
Program selector, Digital clock E06141 0281 M015 (PSC)
Program selector, Data switch, Address Buffer E06141 0291 M010 (PSA)
Storage Interface 1 E06141 0354 M015 (SI1)
Storage Interface 2 E06141 0360 (SI2)
Bus Exchange Balanced 1 E06141 0581 M042 (BxB1)
Bus Exchange Balanced 2 E06141 0591 M024 (BxB2)
Trigonometric Function Unit E06142 0420 M010 (TFU)
Graphic Display Control-ler (GDC) *)
Terminal Communication Controller (TCC) *)
Microprogrammable Line Adapter 1 (MLA1) *)
Microprogrammable Line Adapter 1 (MLA1) *)
Microprogrammable Line Adapter 1 (MLA1) *)
SERVICE PANEL E22496 0001
MAINSFILTER E05134 0001
Storage Module Interface (SMI)
E07150 0480
E07150 0440
E07150 0440
E06142 0790
POWER UNIT E22506 0020 FRAME ASSY
E13998 0117 Control Board
E13998 0038 Scaling Board
E13932 0030 Relay Board
E13998 0092 Control Board
E13998 0130 Control Board
E13998 0140 Control Board
E13955 0025 Stab Board
SEMICONDUCTOR STORAGE UNIT (SSU) 64K E00297 0000
E07150 0480 Storage Bus Termination (SBT)
E07150 0440 Storage Module (SM)
Datorram CPU 932V med TTY och PTE
CP/IO UNIT E22520 0037 FRAME ASSY
I/O-ram Censor 932V med Storage Interface, BusExchange Balanced, Trigonometric Functional Unit
*) Wired, not equipped

7.6.4 Censor 932V maskininstruktionslista

Av kompatibilitetsskäl gjordes instruktionslistan för Censor 932 V helt lik med Censor 932 K. Man tog dock chansen att effektivisera vissa ofta återkommande programloopar genom att utöka listan med följande två instruktioner:

- Load Address s(E) > G(R)
- Test Masked Word s(E) <> g(R); g(R) utgör mask, resultat till CC

Angående beteckningarna ovan se kapitel 5.

7.6.5 Basprogramvara

Eftersom målet med CPU 932V skulle vara en med Censor 932K fullständigt programkompatibel CPU utvecklades ingen särskild basprogramvara, med undantag av drivers för konsolterminalen TTY och kassettbandspelaren DCRC. All basprogramvara som använts i Censor 932K kunde därför ”köras” på CPU 932V.

8 Censor 932E (”europakortscensorn”)

8.1 Bakgrund

I slutet av 70-talet ändrades marknadsbilden för Censor 932. Systemet hade sedan introduktionen 1974 använts i en rad olika militära och civila projekt (exempelvis ATC-system) där datorsystemets realtidsegenskaper och kundanpassade komponenter passade väl in i stationära och centraliserade uppställningar. Mot slutet av 70-talet började dock marknadsläget förändras för denna typ av specialanpassade system. Man ställde nu krav på att systemen skulle baseras på generella och kommersiellt tillgängliga system. Det skulle bli billigare så, trodde man. Datorsystem från Digital Equipment (Vax-datorerna), HP och under en tid Norsk Data var vanliga krav. System för den militära marknaden ställde krav på litet fysiskt utrymme, elektrisk och mekanisk miljötålighet samt mobilitet.

Kraven på kommersiella system och miljötålighet innebar för dåvarande Datasaab att det var dags att avsluta den ”hemtillverkade” virkortseran och planera för den nya situationen.

Eftersom tillgången på kommersiella mobila och miljötåliga system var ytterst begränsad så fann man att trots marknadens krav på kommersiella datorer att det skulle vara ekonomiskt försvarbart att gå vidare med Censorkonceptet men i en ny teknisk kostym. Grunden för detta beslut var att Datasaab under 70-talet förvärvat värdefulla kunskaper om vad som krävdes för att göra datorer för krävande miljöer samt att man efter flera undersökningar visste att det skulle i princip vara omöjligt att ”rugga” en kommersiell dator. Beslutet blev inte förvånande; vi gör en kompakt och miljötålig Censor. Censor 932E med produktnamnet 9107 var född. Om man dessutom gjorde den fullständigt programkompatibel med tidigare versioner av Censor 932 så hade man en stor mängd programvara att återanvända.

Implementeringen av beslutet drog dock ut på tiden eftersom Datasaab efter beslutet uppgick i SRA som därefter blev Ericsson. Eftersom Ericsson redan hade ett eget systemkoncept gynnades beslutet dock i så motto att nu fanns det plötsligt ett systemkoncept med tillbehör som man kunde dra nytta av vid utformningen av den nya Censorn.

8.2 Censor 932E, introduktion

Censor 932E, eller som den brukar benämnas ”europakortscensorn”, är den tredje ”moderniseringen” av Censor 932. Den är funktionellt fullständigt programkompatibel med tidigare versioner av Censor 932. Genom tillkomsten av ett antal nya instruktionsklasser (exempelvis Bit-operationer) har den nya datorn möjliggjort att prestandamässigt effektivare program kan skapas jämfört med tidigare Censor-versioner.

Elektromekaniskt skiljer sig E-versionen dock avsevärt från tidigare versioner. Datorn baseras genomgående på folierade Europakort (dubbel Europa) i flerlagersutförande som placeras i 19”-ramar med folierade bakplan. Genom ett maximalt utnyttjande av den ”state of the art”-teknologi (företrädesvis LSI-kretsar) som var tillgänglig under konstruktionstiden (1984-1986) kunde centralenheten inrymmas på enbart tre kretskort. Genom denna utveckling hade den fysiska volymen av CPU 932 på 15 år minskat från ett fullbestyckat 19”-tums skåp till tre folierade Europakort (!). Övergången till Europakortstandarden innebar dessutom att Censor 900 bussystem i det utförande som använts i virkortscensorn ersattes med industristandarden Versa-bus eller VME-bussen som den vanligen benämns. Med E-versionen ersattes dessutom den klassiska lampbestyckade operatörspanelen (OCP) med en serieport till vilken en ”dum” ASCII- terminal (t.ex. Digital Equipments VT100 eller en PC) kunde anslutas vid behov.

Pappersremsutrustningen som använts i virkortscensorn blev med den nya versionen helt förpassad till historien och kassettbandspelare (Digital Cartridge Recorder) och ”hårddisk” blev datorns standardutrustning.

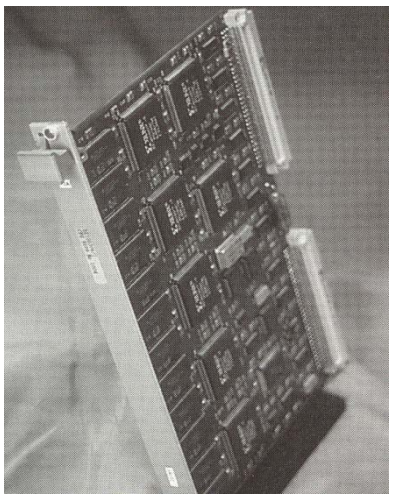
I den nya Censorn infördes flyttalsaritmetik som standard till skillnad mot virkortscensorn där denna funktionalitet utgjorde ett valbart tillval.

8.3 Elektromekanisk uppbyggnad

8.3.1 Europakort

När Datasaab under början av 80-talet så småningom inlemmades med Ericssons militärdivision hade man virkortscensorn i flyttbagaget och ett behov av att modernisera Censorn. Efter - som Ericsson redan hade baserat mycket av sin utveckling på standarden Europakort så var beslutet enkelt. Den nya Censorn skall basera på den mekaniska standarden IEC-60297-3. Den kom senare att bli känd som IEEE 1101.10. Standarden beskriver bl. a. det modulära måttsystemet som bestämmer kortformaten. För Censor 932E bestämdes kortstorlek till 160 x 220 mm eller ”dubbel Europa”, se bilden nedan.

Två stycken 96-poliga kontaktdon (enligt DIN41612) används för kortets signalförbindelser med ”omvärlden”. Den använda kontakttypen valdes bl. a för att passa standarden för datorns nya databuss (se avsnittet nedan angående VME-bussen). En av de två kontakterna används enbart för databussen. Den andra kontakten innehåller ”övriga ”signaler.

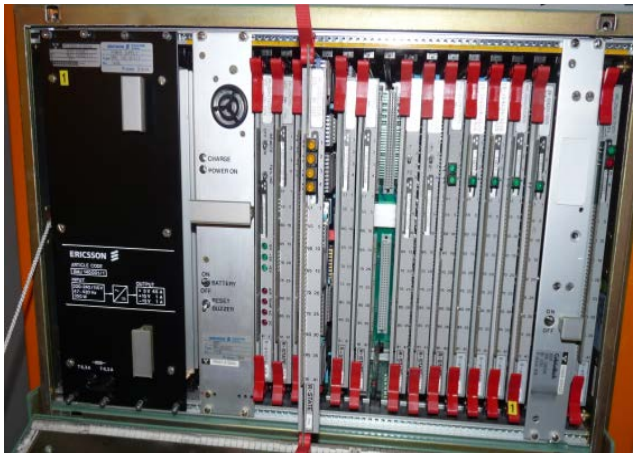


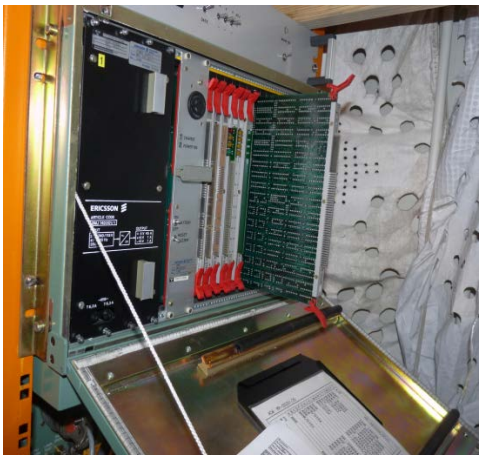
Dubbelt Europakort.

8.3.2 Europaramen

De foliekort som ingick i Censor 932E placerades i en 19”-ram med plats för upp till 20 kort. Kortramens kontaktdon ingår i ett folierat bakplan utformat för databussen VME som sammanbinder samtliga kortplatser i ramen.
I bilden nedan till vänster nedan visas en typisk ramkonfiguration för Censor 932E. Längst till vänster finns ramens strömförsörjningsaggregat. Närmast aggregatet finns en batteribackupenhet för det halvledarminnesbaserade primärminnet. Därefter från vänster till höger följer de tre foliekort som utgör CPU 932E, primärminnesmoduler och övriga kortmoduler som ingår i den aktuella konfigurationen.

Korten skjuts in i ledade korthållare och trycks fast i bakplanets kontaktdon. Korten låses sedan i ramen med de ”röda” kortlåsen. Hela ramen är sedan inbyggd i ett mekaniskt tåligt hölje som samtidigt gjorts elektriskt tät (EMI-tät). Framsidan på ramen är försedd med en låsbar lucka. På bilden till höger nedan visas luckan i nerfällt läge. Luckans kanter är försedd med speciella s.k. EMI-tätningsdon för att ”stänga inne” den elektromagnetiska störstrålningen som datorn genererar. Allt i avsikt att förhindra att datorns störstrålning kan påverka omgivande enheter i det totala systemet.





Datormagasin Censor 932E

8.3.3 Elektronik

Eftersom ett av målen med Censor 932E var att minimera datorns fysiska dimensioner blev komponentvalet enkelt. Man valde att utnyttja de TTL-komponenter i LSI-utförande (Large Scale Integration) som fanns tillgängliga i mitten av 80-talet. Ett avgörande bidrag till att minimeringen lyckades bra, var valet av de effektiva bit-slicekretsarna i bipolär teknik som användes för att realisera CPU’ns aritmetiska och logiska enhet. De ytterst kompakta och snabba minneskretsarna (PROM, Programmable Read Only Memory) som användes för att realisera CPU’ns mikroprogramminne var ett annat viktigt bidrag till minimeringen.

8.3.4 Mikroprogrammering

I likhet med ”virkortscensorn” baserades all styrsignalgenerering i CPU 932E på mikroprogrammering. Snabba PROM-kretsar används för lagring av maskininstruktionerna styrsignalsekvenser.

8.4 Censor 932E bussystem

8.4.1 VME-bussen.

Kraven på en ”minimiserad” Censor innebar dessutom att det tidigare använda databusskonceptet Censor 900 Bus system (se Censor 932V) inte längre kunde användas. Det skulle kräva för många komponenter dvs. ta för stor kortyta. Ett nytt komponentsnålare busskoncept måste väljas. I kravbilden på den nya datorn låg dessutom att en komplett datorkonfiguration av Censor 932E skulle kunna inrymmas i en 19”-ram. Detta innebar att den nya Censorns databussystem endast behövde klara en fysisk utbredning på några decimeter till skillnad från Censor 900 bussystem som måste klara flera meters utbredning. Skulle man trots allt behöva

en ”databuss” med större fysisk utbredning kunde detta lösas genom att skapa ett lokalt nätverk (LAN) baserat på Ethernet.

Genom Datasaabs samgående med Ericsson och valet av den mekaniska kortstandarden Europa blev valet av databusskoncept enkelt då databusstandarden Versa-bus eller VME-bus (Versa Module European) så att säga var barn i samma familj. Eftersom Ericsson redan byggde moduler i detta koncept var det logiskt att den s.k. VME-bussen fick bli den nya Censorns databusskoncept.

Standarden specificerar en databuss, ursprungligen utvecklad för mikroprocessorn Motorola 68000, som senare blev fastlagd av standardiseringsorganet IEC (International Electrotechnical Commission) för att därefter bli ANSI-standard (ANSI/IEEE 1014-1987). Den ursprungliga standarden definierade en 16 bits bus och en 24 bits adressbuss för att passa i en 64-pin DIN41612-kontakt. Över tid har bussen utvecklats för större bussbredder. Idag finns en standard för 64-bits buss.

8.4.2 SCSI-bussen

På samma sätt som ”marknaden” strävade efter att standardisera databussar så pågick en motsvarande standardisering av gränsytan till olika typer av yttre enhet (skivminnen, diskettenheter, magnetbandsenheter etc.). Standardiseringen innebar att de yttre enheterna blev ”intelligentare”. För utvecklingen av den nya Censor innebar detta att det gamla sättet att ansluta yttre enheter med specialkonstruerade styrenheter kunde överges. Nu räckte det med att välja lämpligt standardkoncept för anslutning av exempelvis skivminnen och sedan förse datorn med en styrenhet för detta koncept.

Vid tiden för Censor 932E tillblivelse var standarden SCSI (Small Computer System Interface) den dominerande standarden för yttre enheter. En annan standard var den komponentsnåla IDE/ATA som främst användes för yttre enheter i persondatorer (PC). För Censor 932E fick det bli en styrenhet för SCSI då detta koncept redan används inom Ericsson. En finess med SCSI är att den vid sidan om den rena enhetsstyrningen även implementerar ett bussystem för yttre enheter. Med en SCSI-styrenhet kan man till dess bussystem (SCSI-bussen) ansluta många olika typer av enheter. Styrning av de olika enheterna på bussen sker vi det s.k. SCSI-protokollet.

8.4.3 Censor 900-bussen

Övergången till det nya bussystemet innebar att den nya Censorn inte direkt kunde kopplas in i det tidigare använda bussystemet Censor 900. Detta löstes dock med en specialutvecklad bussväxelenhet benämnd VMEC (VME Converter) vilken fungerade som brygga mellan Censor 900 bussystem och VME-bussen. VMEC konverterar alla adress-, data- och kontrollsignaler utom programanrop (detta görs av enhet PA) mellan Censor 900-bussen och C932E interna buss VME.

VMEC användes i rgc för sammankoppling av Censor 932E med databussen BIXA som utgjorde en Censor 900-buss.

8.5 Censor 932E i rgc

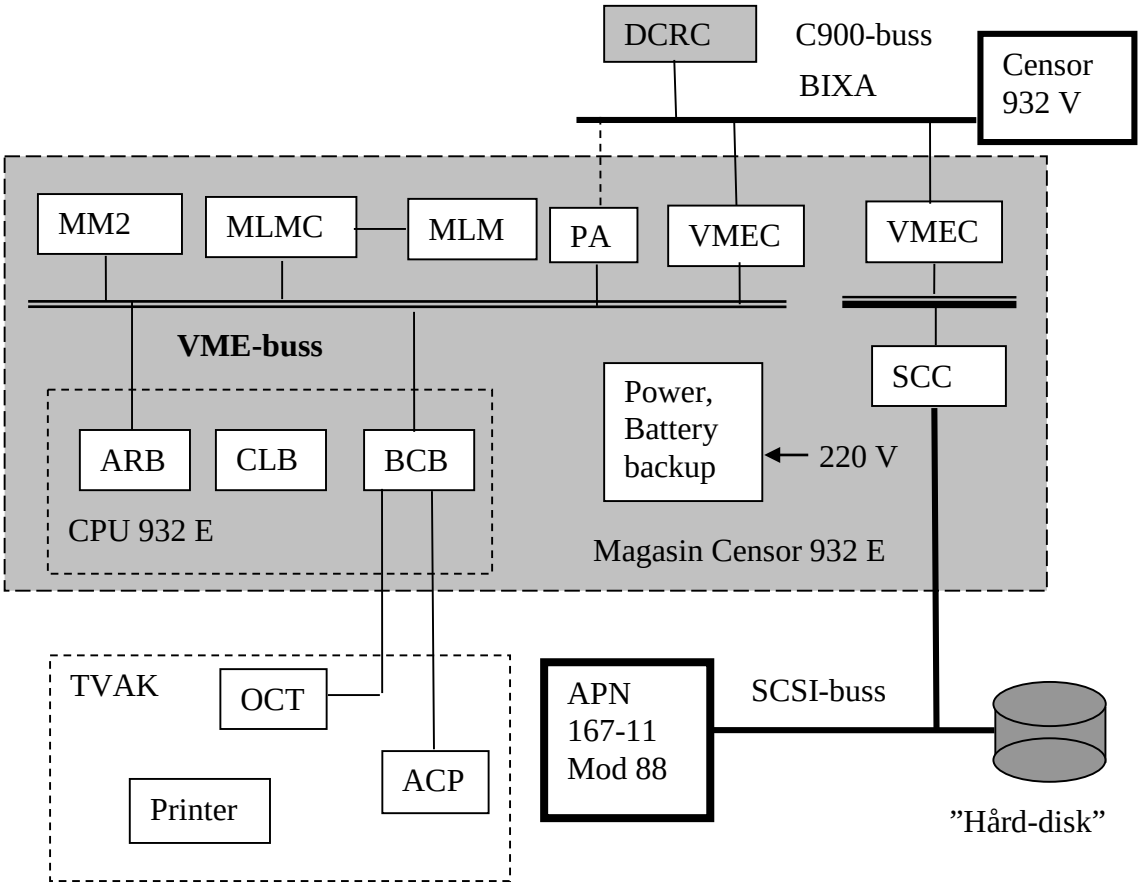
I bilden nedan visas hur Censor 932E integrerades med befintlig utrustning i rgc. Datorn inrymdes i en 19”-ram (kallat Magasin) och innehöll förutom de tre CPU-korten, kraftaggregat, två minneskort (MM2, 512 kbyte DRAM), styrenhet för SCSI-bussen (SCC), bussväxelenhet VMEC, styrenheterna för seriedatakommunikation MLMC och MLM samt slutligen ett anpassningskort (Priority Adaptor, PA) för programanropssignalering mellan Censor 932E och befintliga Censor 932-datorer.

Funktionellt utgör Censor 932E och Censor 932V ett dubbeldatorsystem med skivminne (”hårddisk”) och kassettbandspelare (DCRC) som gemensamma massminnesenheter. Censor 932E når skivminnet via sin styrenhet SCC för SCSI-bussen och kassettbandspelaren på BIXA-bussen via sin bussväxelenhet VMEC. Censor 932V når skivminnet på SCSI-bussen via BIXA och bussväxeln VMEC medan kassettbandspelaren (DCRC) nås direkt via det egna bussystemet BIXA. Genom att SCSI-bussen dessutom är ansluten till rgc’s modernaste datorsystem APN 167-11 så kan de nya datorsystemen överföra data mellan varandra med Dator – Dator-kommunikation på SCSI-bussen.

Med en manuell anropsomkopplare (3 st. vred placerade på baksidan av Bildgeneratorstativet, BG) bestäms vilken dator som skall vara huvuddator och erhålla programanrop från yttre enheter. I programsystemen ingår funktioner för delad last, där huvuddatorn kan lägga arbetsuppgifter i den andra datorn.

Laddning av programsystemet i C932E sker först från kassettbandspelare till skivminnet. Från skivminnet kan programsystemet sedan laddas in i C932E internminne efter initiering från ACP. Kommunikationen med skivminnet sker via VMEC och SCC.

Operatörsutrustningen OCT (Facit Twist) och ACP samt skrivare placerades vid TVAK-positionen.



Datorkonfiguration

8.6 Teknisk beskrivning

8.6.1 Systemöversikt

CPU 932E omfattar följande foliekort i dubbel Europaformat och manöverfunktioner:

- ARB - Aritmetik- och registerkort (Arithmetic and Register Board)
- CLB - Kontroll- och logikkort (Control and Logic Board)
- BCB - Busskontrollkort (Bus Control Board)
- Kontrollterminal OCT
- Kontrollpanel ACP

Korten placeras i en 19”-ram med ett bakplan som är anpassat mot databussen VME. Kontrollterminal OCT och kontrollpanel ACP placerades i anslutning till TVAK-positionen.

8.6.2 Systemblockschema Censor CPU 932E

I blockschemat nedan visas CPU 932E interna uppbyggnad. CPU 932E är i linje med konstruktionsförutsättningarna för CPU 932E fullständigt maskininstruktionskompatibel med tidigare versioner av Censor 932 och har samma registeruppsättning som dessa versioner (se avsnitt 4.2 Censor 932 systemuppbyggnad).

Av uppbyggnadstekniska skäl skiljer sig dock den fysiska implementeringen av dessa register mot tidigare versioner genom att dessa numera ingår som data i CPU’ns internminne (RAM)

dvs. de ”syns” inte på samma sätt i blockschemat som de gjorde i blockschemat för tidigare versioner av Censor 932.

Fysiskt omfattar CPU 932 E följande Europa-kort:

ARB - Aritmetik- och registerkort (Arithmetic and Register Board) utför beräkningar i sina tre register med hjälp av den aritmetisk/logiska enheten samt läser och skriver mot VME-bussen.

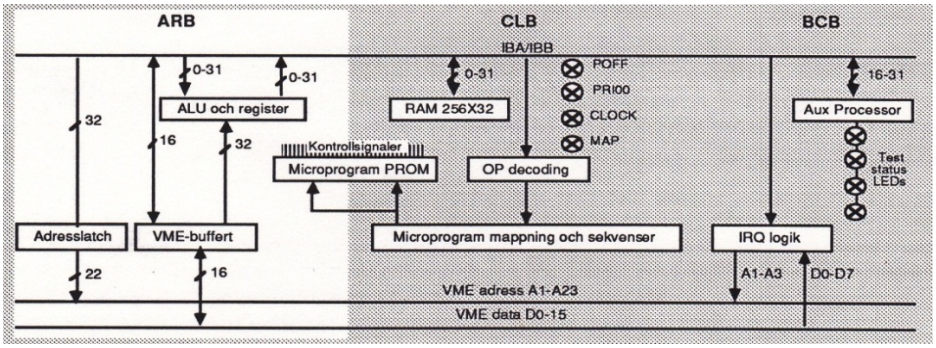
CLB - Kontroll- och logikkort (Control and Logic Board) avkodar programinstruktionerna och ser till att motsvarande sekvens i mikroprogrammet exekveras. Under exekveringen används ett 256 x 32 bitars RAM för lagring av interna variabler och parametrar. CLB hanterar även programanropssignaler och utför programväxling. På CLB finns det fyra lysdioder som användas för att presentera status och resultat från enklare diagnostik.

BCB - Busskontrollkort (Bus Control Board) har hand om busarbitrering och avbrottshantering. På kortet finns även mikroprocessor av typen Motorola MC6803 (Auxillary Processor) som är programmerad att hantera in och utmatning från/till operatörsterminalen OCT, inmatningar från ACP samt utföra enkel diagnostik av datorn.

Mikroprocessorn MC6803, som innehåller 192 bytes RAM av vilka 32 bytes har batteribackup, är försedd med en asynkron full duplex (9600 bit/s) serieport och en parallell I/O-port. Till serieporten ansluts operatörsterminalen Operators Control Terminal (OCT). OCT är vanligtvis en ASCII-terminal av typ VT100. Via den kan operatören utföra de uppgifter som han tidigare utförde via den lamp-och knappbestyckade manöverpanelen OCP (jfr Censor 932 V OCP).

För att möjliggöra programladdning (IPL, Initiell Program Load) och omstart av datorn används kontrollpanel ACP (Auxillary Control Panel). Från denna panel kan datorn ”bootas” från olika s.k. boot-media, t ex bandkassett eller skivminne.

Terminalen OCT möjliggör operatörskommunikation och kontroll av C932E. Med denna terminal kan operatören utföra kontroll av innehåll i censorns arbetsminne (placerat på CLB),
samt beordra exekvering av tester lagrade i MC6803 PROM eller i CPU mikroprogram PROM.



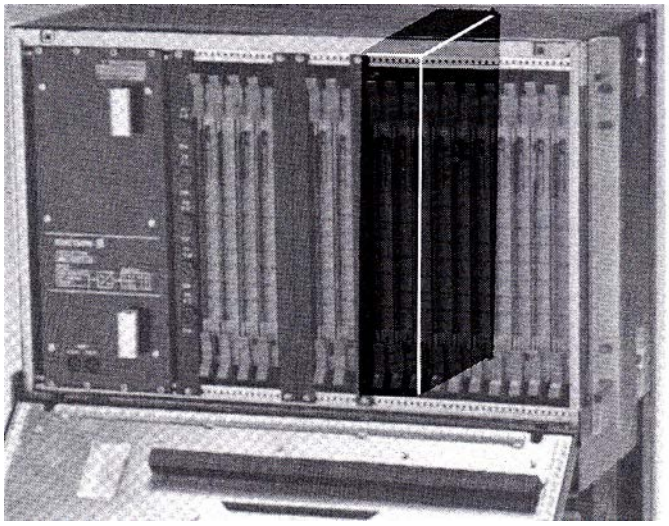
CPU 932 E blockschema

Tester kan utföras när C932E är off-line. MC6803 kan utföra tester på sig själv och ge CPU order att utföra tester på sig själv, VME-bussen och det första minneskortet. Fyra lysdioder

på BCB indikerar binärt resultatet av testerna. Följande testfall förekommer:

Test 1	Kontroll av checksumma för MC6803 PROM
Test 2	Minnestest av MC6803 externt RAM
Test 3	Kommunikationstest MC6803 – Censor mikroprogram
Test 4	Test av C900E-bussen
Test 5	Test av aritmetisk/logiska enheten
Test 6	Test av censorns arbetsminne (på CLB)
Test 7	Test av censorns shift-funktion (avancerad operation)

8.6.3 Stativkonfiguration



Stativritning C 932E

8.6.4 Censor 932E maskininstruktionslista

Av kompatibilitetsskäl gjordes maskininstruktionslistan för Censor 932E helt lik med de tidigare versionerna Censor 932K och Censor 932V (virkortscensorn). Programutvecklingserfarenheterna från de tidigare versionerna av Censor 932 visade att man skulle kunna göra betydligt effektivare programvara om instruktionslistan utökades med ett antal nya instruktionsklasser. Särskilt bit- och byte-instruktioner var efterfrågade. Efter moget övervägande utökades maskininstruktionslistan till slut med följande instruktioner:

LB	Load Byte
LBI	Load Byte Increment Index
STB	Store Byte
STBI	Store Byte Increment Index
SB	Set Bit
RB	Reset Bit
TB	Test Bit
SHA	Shift Arithmetically
SHDA	Shift Double Arithmetically
SHL	Shift Logically
SHDL	Shift Double Logically

MB	Move Block
LHN1	Load Halfword Neighbour 1
LTN1	Load and Test Neighbour 1
LDN1	Load Double Word Neighbour 1
LHN2	Load Halfword Neighbour 2
LTN2	Load and Test Neighbour 2
LDN2	Load Double Word Neighbour 2
LHN3	Load Halfword Neighbour 3
LTN3	Load and Test Neighbour 3
LDN3	Load Double Word Neighbour 3
STWN1	Store Word Neighbour 1
STWN1	Store Word Neighbour 2
STWN1	Store Word Neighbour 3
STR	Store Ram
LR	Load Ram

8.6.5 Basprogramvara

Allt sedan introduktionen av Censor 932V (virkortsmaskinen) pågick en kontinuerlig utveckling av programutvecklingsverktygen för Censor 932. Inledningsvis var mycket baserat på pappersremsor men med införandet av magnetbandskassetten DCRC kunde pappersremsorna överges.

Systemen som byggdes under början av 70-talet var ofta en ”primärminnesresident” realtidsapplikation som hanterade tidskritiska radardata. Allt fokus låg på prestanda. Allt var skrivet i ”optimerad” assembler. I takt med att radardata kombinerades med tyngre objektinformation än bara kurs och fart ökade kraven på att tyngre informationsbehandling skulle kunna göras i ”radardatasystemet”. Exempelvis utgör den informationsbehandlingstunga färdplanhanteringen i ett flygtrafikledningssystem (ATC) exempel på detta. Väderinformation och simuleringsdata är andra exempel.

Sammanfattningsvis innebar den förändrade kravbilden att radardatasystemen i fortsättningen måste innehålla någon form av skivminnesbaserad databas. Den naturliga utvecklingen blev att utvecklingsverktygen och operativsystemet moderniserades i enlighet med den nya kravbilden. Resultatet av denna omvandlingsprocess blev att Censor 932 som från slutet av 60-talet varit ett remshanteringssystem femton år senare blivit ett modernt skivminnesbaserat realtidssystem. Det var därför logiskt att Censor 932 E försågs med den senaste systemprogramvaran.

Sammanfattningsvis omfattar programsystemet för Censor 932 E följande huvuddelar:

- Censor 932 E operativsystem (OS 2.1 NUCLEUS)
- OS 2.1 systemprogramvara (”drivers”, ”managers” etc.)
- Applikationsprogramvara