

**RTV KLUB MURSKA SOBOTA**

# **DIGITALNI MOSTOVI**

**"HITRI PACKET RADIO"**

**Murska Sobota, september 2000**

# SV360 procesorski modul za SuperVozelj

M. Kovačević, S57MMK

Več razlogov je privedlo do potrebe po razvoju novega, zmogljivejšega procesorskega modula za SuperVozelj. Po nekaj letih rasti hitrega paketnega omrežja smo v S5 prišli do stanja, ko se je na isti frekvenci, na posameznih področjih gnetlo tudi po več kot 10 PSK postaj. Tako je na Ljubljanskem področju delovalo kar 5 vozlišč na isti frekvenci (LJU, LJUH, S50LEA, GOLO, BRKINI) ter cela vrsta uporabnikov. Problem skritih postaj je tako postal zelo pereč, prenosne hitrosti so občutno padle. Hkrati se je vedno bolj uveljavljal TCPIP način dela, kar je za posledico imelo bistveno povečano aktivnost (QRM:) nekaterih vozlišč (LJU, S50LEA) in uporabnikov. Edina rešitev je bila QSY nekaterih linkov, a kaj ko SuperVozelj ni podpiral več kot 2 hitra (1M2bps) kanala. Obstoječa HW oprema (68020 + DMA) ni prenesla nadaljnje širitve zmogljivosti, a tudi vsak napor v tej smeri bi predstavljal izguba časa glede na precejšnjo zastarelost vgrajenih komponent.

Postavljeni smo bili pred zahteven problem izbora ustreznega procesorja (in perifernih vezij), ki bi uspešno reševal že omenjeno problematiko. Zelo mikavni so bili sodobni RISC procesorji (StrongARM, MPC860...). Izbira kakšnega izmed teh bi pomenila praktično začetek iz ničle (brez izkušenj, nov ukazni nabor, nova razvojna orodja, sodobna ohišja problematična za domačo gradnjo... pomanjkanje časa in ljudi, ki so pripravljeni sodelovati...).

Odločili smo se za že rahlo zastarel MC68360 procesor. MC68360 je kompleksno vezje, ki je predvsem komunikacijsko usmerjeno. Razen procesne enote (CPU32), ponuja še štiri hitre HDLC kanale (do 2Mbps) ter celo vrsto drugih funkcionalnih enot, primernih za gradnjo paketnega komunikacijskega vozlišča. Skromne procesorske zmogljivosti kompenzira ustrezna arhitektura perifernih enot ter DMA vezij. Gradnja procesorskega modula z uporabo MC68360 je izredno enostavna. Procesor je združljiv z 32bitnimi procesorji podjetja Motorola (npr. MC68020) na nivoju ukaznega nabora. Tako je prenos obstoječe programske opreme SuperVozlja na nov procesorski modul tudi relativno enostaven.

## Kratek opis vezja MC68360

MC68360 QUICC (Quad Integrated Communication Controller) je integriran procesor (CPU32+), periferne komunikacijske enote (CPM) ter sistemske podporne enote (SIM60) v enem vezju.

Veze je predvsem primerno za komunikacijske aplikacije. Proizvajalec (Motorola) izdeluje vezje v 240 pinskem PGA ali QFP ohišju z osnovnim taktom 25MHz ali 33MHz.

Procesorska enota vezja QUICC doseže 4.5 MIPS pri 25MHz taktu. Enota podpira 32bitno podatkovno in 32 bitno naslovno vodilo ter dinamično prilagajanje širine podatkovnega vodila (8 bit, 16 bit ali 32 bit).

Naslovni prostor je linearen. Procesorska enota je združljiva z drugimi procesorji familije 68K z CPU32 arhitekturo na nivoju strojne kode. Veze podpira 7

zunanjih prekinitvenih linij, 12 I/O pinov z možnostjo generiranja prekinitve, 16 internih prekinitvenih virov z nastavljivo prioriteto. Veze vsebuje HW razhroščevalno vezje (BDM), ki omogoča vpogled ali vpis v memorijo, vpogled ali nastavitvev registrov procesorja ter nastavitvev pasti v program z uporabo enostavnega vmesnika za paralelni port PC računalnika ter ustreznega razhroščevalnega programa.

Komunikacijski procesorski modul (CPM) sestavljajo naslednje funkcionalne enote: RISC procesor, 2.5KB RAM z dvojnimi vrati (dual port RAM), 14 DMA kanalov (SDMA), 3 paralelna I/O registra, 4 SCC večprotokolna serijska sinhrona kanala, 2 UART kanala, 1 SPI serijski kanal, 4 generatorja bitnega takta (BRG).

RISC procesor je centralni procesor komunikacijskega modula (CPM) na ločenem 32 bitnem vodilu. Ta procesor realizira izbran protokol na posameznem serijskem komunikacijskem kanalu ter ima pod kontrolo DMA kanale (SDMA), ki prenašajo podatke med SCC kanali in memorijo. Tako RISC procesor prevzame nase komunikacijske naloge na najnižjem nivoju ter razbremeni glavni procesor. RISC procesor komunicira z glavnim procesorjem na več načinov. Prvič, večina parametrov se izmenjuje preko RAMa z dvojnimi vrati. Drugič, RISC procesor izvaja ukaze glavnega procesorja. Tretjič, RISC procesor lahko generira prekinitve glavnemu procesorju. Četrto, RISC procesor nastavlja statusne registre, ki so dosegljivi s strani glavnega procesorja. RISC procesor nam omogoča še uporabo 16 dodatnih časovnikov, ki so uporabni pri realizaciji posameznih protokolov.

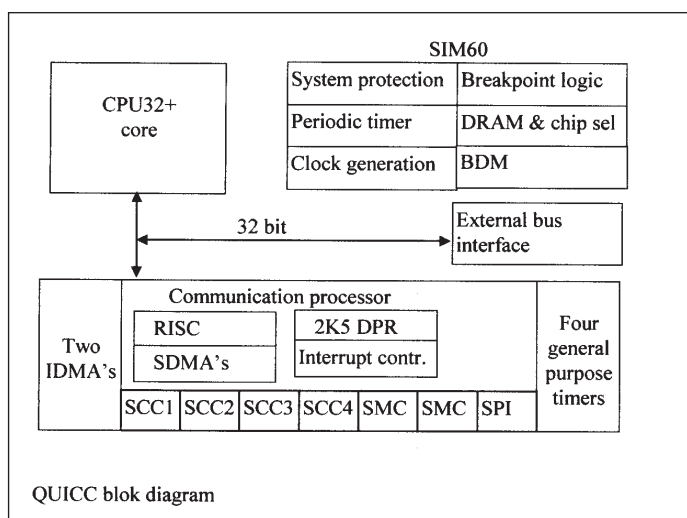
RAM z dvojnimi vrati (DPM) velikosti 2560 zlogov je dosegljiv s strani RISC procesorja, glavnega procesorja ali zunanjega gospodarja systemskega vodila. DPM se uporablja za nastavitve parametrov vezanih na posamezni komunikacijski kanal, nastavitvev buffer deskriptorjev (BD) ki določajo kam se shranjujejo sprejeti podatki in kje se nahajajo podatki za oddajo. DPM se lahko uporablja tudi kot navadni RAM za uporabniški program ali za shranjevanje sprejetih podatkov. Nadaljnja uporaba je za hranjenje RAM mikrokode RISC procesorja. To omogoča implementacijo dodatnih protokolov.

SDMA kanali so namenjeni izključno prenosu podatkov med RAMom in serijskimi komunikacijskimi kanali. Za vsak kanal imamo na razpolago po 2 SDMA kanala, kar nam omogoča duplex delovanje. Če so podatki za sprejem ali oddajo v sistemskem spominu, potem posamezni SDMA zahteva systemsko vodilo od arbitražne enote QUICC. Če se pa podatki nahajajo v DPM potem so DMA cikli nevidni.

Večprotokolni serijski komunikacijski kanali (SCC) podpirajo naslednje protokole: HDLC/SDLC, HDLC bus, BISYNC, UART, AppleTalk in transparentni protokol (TTP). Dodatno, v izpeljanki vezja MC68EN360 je podprt tudi 10Mbps Ethernet/IEEE 802.3 na kanalu SCC1. Možna je implementacija dodatnih protokolov. SCC kanali omogočajo 2Mbps HDLC duplex istočasno na vseh kanalih. Veze vsebuje DPLL enoto za izločanje bitnega takta z NRZ, NRZI, FM0, FM1, Manchester kodiranjem. Bitni takt lahko generiramo na osnovi zunanjega pina, DPLL vezja ali generatorja bitnega takta (BRG). Veze

vsebuje 32 zlogovni FIFO vmesnik za SCC1 ter 16 zlogovni FIFO vmesnik za ostale kanale.

Sistemiški integracijski modul (SIM60) vsebuje celo vrsto funkcij ki poenostavijo gradnjo procesorskega



modula z QUICC vezjem za komunikacijske aplikacije. Med te funkcije sodijo: kontrola porabe vezja (low power, slow go), sinteza glavnega urnega takta vezja, razhroščevalno vezje (BDM), logika za izbiro perifernih enot (chip select... z možnostjo nastavitve začetnega naslova, velikosti, števila čakalnih stanj ter pravice do pisalnih ciklov), vezja za dinamične RAME, vezja za arbitražo sistemskega vodila, programski kužapazi, kužapazi ciklov na sistemskem vodilu itd. Vse te funkcije so programsko nastavljive po resetu ali vklopu napajanja.

### Zasnova CPU modula

CPU modul je zasnovan na ploščici Evropskega formata in vsebuje procesorsko vezje MC68360, 64KB EPROM, 512KB ali 2MB statični RAM, uro realnega časa ter vmesnik za enostavno 16 bitno vodilo SuperVozlja. Ploščica je izdelana v štiri slojni tehniki. Predvidena je uporaba SMD komponent, razen za EPROM (DIL podnožje).

Procesorsko enoto poganja 4.9152MHz kvarc. Interno DPLL vezje poskrbi za generiranje nazivnega procesorskega takta (24.576MHz ali 33.024MHz). Izbrana frekvenca kvarca omogoča uporabo glavnega procesorskega takta za generiranje bitnega takta (BRG) za serijske vmesnike na vseh standardnih prenosnih hitrostih. Vezje MC68360 se resetira na osnovi /RESET signala SV vodila. Število čakalnih stanj za posamezno spominsko enoto ali enoto na SV vodilu je določeno interno (programska nastavitve). Tako ni potrebna uporaba signalov /DSACK0 ali /DSACK1, ki sta predvidena za asinhrono zaključitev procesorskih ciklov.

Na tiskanini je predvideno 28 pinsko DIL podnožje za EPROM. EPROM je priključen na procesor kot 8 bitni zunanji spomin. Kot izborni signal (chip select) EPROMa je uporabljen /CS0 izhod vezja MC68360. V podnožje lahko vstavimo EPROMe tipa 27C256 ali 27C512 z maksimalnim časom dostopa 200 nsec.

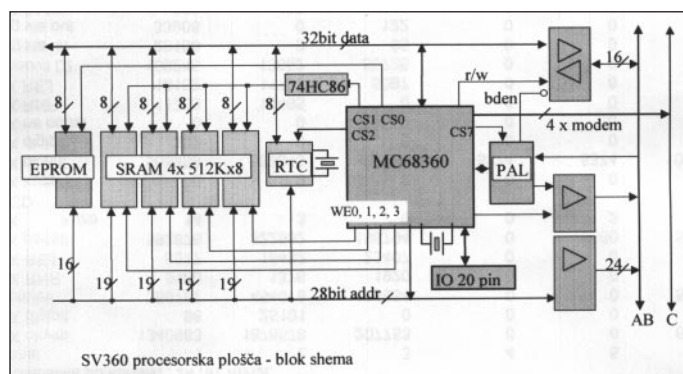
Statični RAM kapacitete 512KB ali 2MB je priključen na procesor 32 bitno. Predvidena je uporaba štirih SRAM vezij 128Kx8 (32 pin) ali 512Kx8 (36 pin) v SOJ ohišju (400 mil SOJ center power), ki je znano tudi kot

“revolucionarno” ohišje. V tem ohišju so pakirani hitri SRAMI (čas dostopa 35 nsec ali manj). Ker je SOJ ohišje “simetrično”, lahko v isto podnožje vstavimo SRAME različne kapacitete in števila nopic. Kot izborni signal (chip select) SRAMov je uporabljen /CS1 izhod vezja MC68360. Procesor MC68360 omogoča dostop do zunanjega spomina minimalno v dveh periodah procesorskega takta (80 nsec pri 25Mhz procesorju, 66nsec pri 33Mhz procesorju). Izbrani SRAMI omogočajo uporabo takšnega načina dostopa. Izbrana zloga pri pisalnih ciklih poteka na osnovi dvofunkcijskih signalov A31/WE0, A30/WE1, A29/WE2, A28/WE3 procesorja. Funkcijo teh signalov določimo programsko pri inicializaciji procesorja. Statični RAM je napajano baterijsko (+CMOS napajalna linija SV vodila). Tako je zagotovljeno hranjenje pomembnih nastavitvev SV tudi ko izpade glavno napajanje (+5V). Za nizko porabo SRAMov pri izklopljenem glavnem napajanju skrbi vezje z dvemi XOR vrati (74HC86) ki zagotavlja +CMOS nivo na /CE pinih. V vezje lahko vstavimo kaksnega izmed naslednjih SRAMov (primeri samo za 512Kx8) : Samsung KM684002AJ-20, NEC uPD434008ALE20, Alliance AS7C4096-25JC, Cypress CY7C1049-20VC...

Za uro realnega časa sem izbral vezje DP8573A (National Semiconductors). Vezje je priključeno na procesor 8 bitno. Kot izborni signal (chip select) ure realnega časa je uporabljen CS2 izhod vezja MC68360. /RD in /WR signala za čitanje vsebine oz. vpis v registre ure realnega časa sta generirana na osnovi /CS7, /OE in /DS signalov procesorja v programabilnem vezju GAL22V10. Vezje ure realnega časa zasede 32 zlogov naslovnega prostora procesorja. Vezje je baterijsko napajano, poganja ga pa kvarc 32.768 kHz. Vezje lahko generira prekinitvev (periodično, alarm ali izguba napajanja) na IRQ6 prekinitvenem vhodu procesorja. Vezje vsebuje trenutni čas od stotink sekunde do leta (tudi prestopnega leta) v BCD formatu.

Vmesnik za 16 bitno SV vodilo omogoča dostop do spominskih ali perifernih naprav na SV vodilu 8 bitno ali 16 bitno. Vsi signali SV vodila so priključeni na CPU modul preko ločilnih vmesnikov in speljani na A in B vrstici 96 pinskega konektorja vodila. Vmesnik za SV vodilo ne omogoča obstoja dodatnih gospodarjev SV vodila (npr. DMA kartica). Procesor dostopa spominske lokacije na SV vodilu kadar je aktiven izborni signal /CS7 vezja MC68360. Linije podatkovnega vodila SV (D0 - D15) so

priključene na procesor modula preko dvosmernih vmesnikov 74HC245 ki sta v stanju visoke impedance kadar je neaktiven signal /CS7. Smer vmesnikov 74HC245 določa signal R/W procesorja. Naslovne linije procesorja (A1 - A23) so speljane na SV vodilo preko treh enosmernih vmesnikov 74HC244. Vmesniki so vedno



odprti v smeri proti vodilu. Krmilna logika (GAL22V10) vmesnika SV vodila skrbi za generiranje /LDS in /UDS signalov vodila na osnovi /DS, A0, SIZ0 in SIZ1 signalov procesorja. Tako je omogočen 8 bitni ali 16 bitni dostop do spominskih ali perifernih enot na vodilu. Signali /AS, R/W, FCO-2 so speljani iz procesorja na SV vodilo preko vmesnika 74HC244. Krmilna logika SV vodila generira /RESETH signal kadar je aktiven signal /MR (master reset - reset tipka ali vklop napajanja) na SV vodilu. Istočasno se generira tudi /RESET signal na SV vodilu. /RESET signal SV vodila se generira tudi takrat, kadar se procesor resetira interno (fatalna napaka ali reset ukaz). Celotno logiko krmiljenja vmesnika SV vodila vsebujejo naslednje logične enačbe (Abel):

Podatkovni signali (TxD, RxD), urni signal bitnega takta (RTxC) ter kontrolni signali (RTS, DCD) za SCC kanale 1 -

```
module _sv360 (DEV);
    flag 'r3';
title 'SV360 local logic & bus cntrlr, by Marko Kovacevic, S57MMK'
    sv360 DEVICE 'P22V10';

"Input pins
    CLK0,SIZ0,SIZ1,_DS,A0,_OE,_SRAM,_EPROM,_CSXX,_DSACK,_MR,_AS
pin 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13;
"I/O pins
_UDS,_LDS,_BDEN,_RESET,_DSACK1,_RSTH,_RSTS,_TRD,_TWR,_CSBUS
pin 14, 15, 16, 17, 18, 19, 20, 21, 22, 23;
"declarations
    DS = !_DS; OE = !_OE; SRAM = !_SRAM; EPROM = !_EPROM; CSXX = !_CSXX;
    DSACK = !_DSACK; MR = !_MR; AS = !_AS; UDS = !_UDS; LDS = !_LDS;
    BDEN = !_BDEN; RESET = !_RESET; DSACK1 = !_DSACK1; RSTH = !_RSTH;
    RSTS = !_RSTS; TRD = !_TRD; TWR = !_TWR; CSBUS = !_CSBUS;
equations
    UDS = !A0 & DS;
    LDS = DS & (A0 # SIZ1 # !SIZ0);
    BDEN = CSBUS & AS & DS;
    RESET = RSTS # RSTH # MR;
    DSACK1 = FALSE;
    RSTH = MR;
    ENABLE RSTH = MR;
    TRD = CSXX & OE & DS;
    TWR = CSXX & !OE & DS;
end _sv360;
```

4 so speljani na C vrstico 96 pinskega trovrstičnega konektorja procesorskega modula za priključitev na SV vodilo in modeme (manchester, scrambler) direktno iz vezja MC68360. Signal bitnega takta je skupni za oddajnik in sprejemnik in je zvezan na dva CLK pina vezja MC68360. CLK pine vezja MC68360 programsko (inicializacija BRGjev) povežemo na izbrani SCC kanal.

Signali paralelnega porta B (18 bitov) so povezani na poseben 20 pinski dvovrstični konektor. Vsak posamezni izhodni signal porta B je programsko nastavljen kot vhodni ali izhodni, pri čemer izhodne signale lahko programsko nastavimo kot aktivne ali open drain izhode. Na ta konektor sta zvezani tudi napajalni liniji +5V in GND. Vhodno izhodni signali paralelnega porta so uporabni za priključitev različnih digitalnih ali analognih senzorjev za ugotavljanje stanja posameznih podsistemov vozlišča.

### Prireditev programske opreme Super Vozlja

Za delovanje programske opreme Super Vozlja (npr. SV79d) so potrebne naslednje dopolnitve in spremembe: inicializacija SIM60 modula vezja MC68360, inicializacija SCC kanalov 1 - 4 glede na tabelo parametrov kanalov (parblok), inicializacija ure realnega časa in izdelava podprogramov za nastavitve in čitanje stanja ure

realnega časa, izdelava nove DMA zanke (dmazank).

Program SV79dx zahteva definicijo nekaj osnovnih parametrov, na osnovi katerih se izvršijo nastavitve SIM60 modula ter organizacija delovnega spomina. Te parametri so: sysclk - sistemski takt procesorja, ki mora biti deljiv z 76800, ram - začetni naslov statičnega RAMa, ramdlg - velikost statičnega RAMa (512K ali 2M), QUARTZ - nazivna frekvenca sistema kvarca, risctck - perioda časovnika RISC procesorja. Naslovi ure realnega časa, EPROMa in perifernih naprav na vodilu SV so fiksni. Po resetiranju procesorja se najprej izvrši nastavitve osnovnih parametrov sistema integracijskega modula (SIM60). Ta vključuje nastavitve osnovnega naslova internega spomina z dvojnimi vrati (MBAR), nastavitve registrov DPLL vezja za generiranje sistema takta na osnovi kvarca 4.9152 MHz, brisanje vsebine spomina z dvojnimi vrati (DPR), reset RISC procesorja, izbira funkcij večfunkcijskih pinov vezja MC68360 (PEPAR), nastavitve izbornih signalov za periferna vezja (chip select... GMR, ORx, BRx) ter nastavitve načinov delovanja paralelnih portov. Aktualna verzija programske opreme za procesor MC68360 (SV79dx) nastavi omenjene parametre na naslednji način:

sistemski takt:	24.576.000 Hz ali 33.024.000 Hz (sysclk)
perioda RISC časovnika:	125 usec
paralelni port B:	vsi pini aktivni izhodi
naslov DPR:	\$70000
naslov statičnega RAMa (CS1):	\$200000 velikost: 512K ali 2M (ramdolg), r/w
naslov EPROMa (CS0):	\$0 velikost 64KB, samo čitanje
naslov ure realnega časa (CS2):	\$10000 velikost 64KB, r/w
naslov periferije na SV vodilu (CS7):	\$E0000 velikost 64KB, r/w

Inicializacija internih serijskih sinhronih kanalov SCC1-4 se izvede na osnovi podatkov vpisanih v tabeli parametrov kanalov (parblok). Tabela je razširjena tako, da podpira maksimalno 10 kanalov. Pri tem so kanali 1 - 4 v tabeli parametrov obvezno DMA kanali (prvi parameter ima vrednost 4), kanali 5 - 10 pa kanali na zunanji SCC kartici (Z8530). Tretji parameter (hitrost) kanalov 1 - 4 določa ali bo kanal inicializiran za scrambler (hitrost = 0) ali za manchester modem (hitrost = 2400 - 76800bps). Pri kanalih z hitrostjo enaki 0 se uporabi zunanji signal bitnega takta (RTXC7-10). Pri kanalih z hitrostjo različno od 0 se uporabi interni BRG z DPLL za izločanje bitnega takta.

Inicializacijo DMA kanala začnemo z določitvijo načina delovanja kontrolnih pinov kanala (RTS, DCD, CTS). Signala RTS in DCD sta nastavljena kot I/O pina. Komunikacijski modul vezja MC68360 omogoča tudi avtomatično generiranje RTS signala. Ta način ne izberemo pri inicializaciji ker komunikacijski modul ne pozna možnosti za nastavitve zakasnitve oddaje podatkov po aktiviranju RTS signala, zakasnitve deaktiviranja RTS signala po končani oddaji podatkov ter naključnega izbora trenutka prehoda na oddajo (persistenca, slot). Vse te funkcije na žalost mora opraviti glavni procesor. Signal CTS je nastavljen kot vhodni pin in je vedno aktiven tako da ne vpliva na oddajnik.. Inicializacijo nadaljujemo z nastavitvijo periodičnega časovnika RISC procesorja komunikacijskega modula, ki nam daje osnovo za časovni potek pri oddaji okvirjev (txdelay, txtail, persistence, slottime). Periodo tega časovnika nastavimo na 125 usec.



Po nastavitvi izvora bitnega takta (ter eventuelno BRG) za kanal nastavimo še strukturo verige vmesnikov za sprejem in oddajo za vse interne SCC kanale v spominu z dvojnimi vrati. Za vsak kanal dodelimo (dodeli) 16 sprejemnih blokov katerih številke vpisemo v tabelo blokov (blok, glej strukturo spodaj) ter naslove kot kazalec (\*data) in jim nastavimo stanje "pripravljen" (rx\_status). Ravno tako dodelimo eden blok za oddajo, ki mu nastavimo status na "ni pripravljen" (tx\_status). Zadnjemu sprejemnemu bloku (RxB[16]) v strukturi KANAL\_BD ter oddajnemu bloku še nastavimo status "zadnji". Tako bo RISC procesor "vedel" kje je konec verige vmesnikov in bo lahko začel nanovo od prvega. Strukturo, ki opisuje vmesnike za sprejem in oddajo lahko definiramo na naslednji način ( C ):

```
struct rx_bd {
    unsigned rx_status, // struktura ki opisuje vmesnik za sprejem
    int length, // statusna beseda sprejemnega vmesnika
    far *data // dolžina sprejetega okvirja
    // kazalec na spominski blok za sprejem
    // podatkov
};
struct tx_bd {
    unsigned tx_status, // struktura ki opisuje vmesnik za oddajo
    int length, // statusna beseda oddajnega vmesnika
    far *data // dolžina okvirja za oddajo
    // kazalec na podatke za oddajo
};
struct kanal_bd {
    // struktura ki opisuje podatkovne vmesnike
    // za posamezni kanal
    struct rx_bd [16], // sprejemni podatkovni vmesniki
    struct tx_bd, // podatkovni vmesnik za oddajo
    int prazno[49], // nezaseden prostor
    int index, // indeks v tabeli RxB
    int blok[16] // številke dodeljenih blokov za podatke
    // sprejetih okvirjev
};
struct kanal_bd Kanal_bd[4]; // vmesniki za 4 kanale na naslovu $70000
// (DPR).. zasede 1KB
```

Po inicializaciji strukture, ki opisuje sprejemne in oddajni vmesnik, nam preostane samo še to da RISC procesorju damo ukaz, naj začne sprejem in oddajo na osnovi podatkov v opisani strukturi. RISC procesor nato začne sprejem podatkov, jih vpisuje v spomin, kamor kaže kazalec \*data prvega bloka za sprejem (RxB[1]). Ko se sprejem okvirja zaključi uspešno ali z napako, nastavi rx\_status na "zaključeno" ter eventuelno ustrezne bite o vrsti napake in dolžino sprejetega okvirja (length). Nato RISC procesor nadaljuje z sprejemom naslednjega okvirja na osnovi podatkov iz RxB[2]. Ko RISC procesor tako zaključi zadnji blok (RxB[16]) nadaljuje z prvim. Če pri tem pridemo do bloka, ki ima stanje "zaključeno", RISC procesor zavrže vse prihajajoče okvirje, dokler se ta blok ne sprost ("pripravljen"). Dolžnost glavnega procesorja je, da zaključene bloke posreduje naprej glavnemu sprejemniku in jih nadomešča z nanovo dodeljenimi bloki. Kot je razvidno iz zgornje podatkovne strukture, je za vsak SCC kanal predviden samo eden blok za oddajnik. Glavni procesor nastavi status (tx\_status) temu bloku na "pripravljen", potem ko je potekel čas zakasnitve oddaje okvirja po vklopa oddajnika. RISC procesor reagira na ta status tako, da takoj začne z oddajo okvirja iz spominskega bloka, na katerega kaže kazalec \*data..

Glavni procesor obdeluje sprejete okvirje na osnovi skaniranja stanja tabele rx\_bd za posamezni kanal. Pri tem so vse prekinitve serijskega kanala prepovedane. Na poenostavljen način (zaradi preglednosti) lahko opišemo postopek obdelave sprejetih okvirjev na naslednji način (C):

Po prvih poskusih sem ugotovil, da se tabela rx\_bd zelo hitro polni z kratkimi okvirji ali okvirji z napako (abort, CRC...). Na posmeznem kanalu dobimo neveljaven okvir

```
for (i = 0, i < 4, i++) {
    index = Kanal_bd[i].index; // i je indeks po kanalih 1 - 4
    if ( Kanal_bd[i].rx_bd[index].rx_status != READY ) {
        // rx_bd je zaključen, preverimo status na napako in min dolžino okvirja
        if ( (Kanal_bd[i].rx_bd[index].rx_status != NAPAKA) &&
            (Kanal_bd[i].rx_bd[index].length >= MINDOLG) ) {
            // številko bloka z podatki posredujemo sprejemniku
            sprejemnik[naslednji] = Kanal_bd[i].blok[index];
            // dodeli blok in dopolni tabelo rx_bd
            blok = dodeli();
            Kanal_bd[i].blok[index] = blok;
            Kanal_bd[i].rx_bd[index].data = naslov(blok);
            // spremeni status rx_bd na "pripravljen"
            Kanal_bd[i].rx_bd[index].rx_status = READY;
            // premakni index v tabeli rx_bd za 1... začnem z 0 ko pridem do konca tab.
            index += 1;
            Kanal_bd[i].index = index % 16;
        } else { // napaka ali prekratek okvir
            // spremeni status rx_bd na "pripravljen"
            Kanal_bd[i].rx_bd[index].rx_status = READY;
            // premakni index v tabeli rx_bd za 1... začnem z 0 ko pridem do konca tab.
            index += 1;
            Kanal_bd[i].index = index % 16;
        }
    }
}
```

približno vsakih 40 usec. Iz tega razloga sem moral dvigniti prioriteto dma zanke v primerjavi z glavno zanko programa. Tako je zagotovljeno, da se tabela rx\_bd nikoli (ali samo izjemoma) popolnoma ne zapolni.

Glavni procesor skanira vrste blokov za oddajo za posamezni kanal in posreduje bloke za oddajo na tx\_bd vmesnik, pri čemer so vse prekinitve serijskega kanala prepovedane. Pri tem mora glavni procesor poskrbeti tudi za časovni potek aktiviranja oddajnika (PTT, txdelay, txtail, persistence, slottime). Časovna osnova za kontrolo časovnega poteka aktiviranja oddajnika je izvedena na osnovi enega izmed časovnikov RISC procesorja in znaša 125 usec Poenostavljeno, potek oddajnega dela dma zanke lahko prikazemo na naslednji način ( C ):

### Pripomočki za razhroščevanje modula in programske

```
for (i = 0, i < 4, i++) {
    // i je indeks po kanalih 1 - 4
    if ( PTT(i) aktiven && !txdelay(i) ) { // txdelay je 1 medtem ko teče txdelay timer
        if ( (Kanal_bd[i].tx_status != READY) && !tail(i) ) {
            // oddaja okvirja končana... sprost in startaj tx tail delay
            sprost (); // sprost že oddani blok
            txtail_delay(i); // start txtail delay
            tail(i) = 1; // tail je 1 medtem ko teče txtail timer
            break; // nadaljuj z naslednjim kanalom
        } else if ( tail && txtail_delay(i) potekel ) {
            // oddaja okvirja končana... tx tail potekel... vzemi naslednji blok
            tail(i) = 0;
            if ( blok_za_oddajo_pripravljen(i) ) {
                blok = izokvir(i);
                Kanal_bd[i].tx_bd.data = naslov(blok);
                Kanal_bd[i].tx_status = READY; // start tx
            } else ptt_off(i); // ni naslednjega bloka... ugasni PTT
            break; // nadaljuj z naslednjim kanalom
        }
    } else if ( PTT aktiven && txdelay(i) ) { // tx delay potekel.. začni z oddajo
        blok = izokvir(i); // blok iz vrste za oddajo
        Kanal_bd[i].tx_bd.data = naslov(blok); // nastavi kazalec na podatke
        Kanal_bd[i].tx_status = READY; // start tx
        txdelay(i) = 0; break;
    } else if ( dcd(i) break; // DCD aktiven... počakajmo na neaktivnega
    else if ( (blok_za_oddajo_pripravljen(i) && ( slot_delay(i) potekel ) ) ) {
        if ( psevnak > PERSISTENCE ) { // začnimo postopek persistenca/slot/txdelay
            ptt_on(i);
            tx_delay(i); // start tx delay
            txdelay(i) = 1;
            break; // nadaljuj z naslednjim kanalom
        } else { slot_delay(i); // start slot delay
            break;
        }
    }
}
```

### opreme

Procesorski modul je opremljen tudi z BDM konektorjem za razhroščevanje (Background Debug Mode). Razhroščevanje je hardwareško podprto na vseh Moto-

rolinih procesorjih z CPU32 arhitekturo. Na BDM konektor priključimo posebni vmesnik za paralelni port PC računalnika. Z uporabo programa BD32, ki teče na PC računalniku lahko zasledujemo stanje registrov procesorja, vpisujemo ali čitamo vsebino spomina na procesorski plošči ali spomina z dvojnimi vrati, nastavljamo kontrolne točke v programu (breakpoint), resetiramo, ustavimo ali zažemo procesor. Takšno orodje je zelo primerno za učinkovito ugotavljanje hardwarskih napak na procesorskem modulu ali odpravljanju napak v programski opremi.

Shemo BDM vmesnika za paralelni port PC računalnika in program BD32 z uporabniškim navodilom lahko najdemo na Motorolini domači strani

(<http://www.mot.com/SPS/RISC/netcomm/tools/freeware/360debug.html>).

Že izdelano tiskanino za BDM vmesnik lahko dobimo pri Franciji (S51RM).

### **Izdelava procesorskega modula**

Tiskanina procesorskega modula je izdelana v štiri slojni tehniki. Vse komponente so v ohišjih za SMD montažo. Sestavljanje modula je relativno zahtevno opravilo in zahteva primerne izkušnje in orodje za spajkanje. Posebno težavno je spajkanje procesorskega vezja MC68360 z 240 nogicami v QFP ohišju z rasterjem nogic 0,5mm in statičnih RAMov v 36 pinskem SOJ ohišju z rasterjem nogic 1.27mm. Enostaven, a zelo občutljiv način spajkanja teh vezij lahko opravimo z uporabo plinskega spajkalnika z nastavkom za vroč zrak ali ploščati plamen. Najprej natančno postavimo vezje na

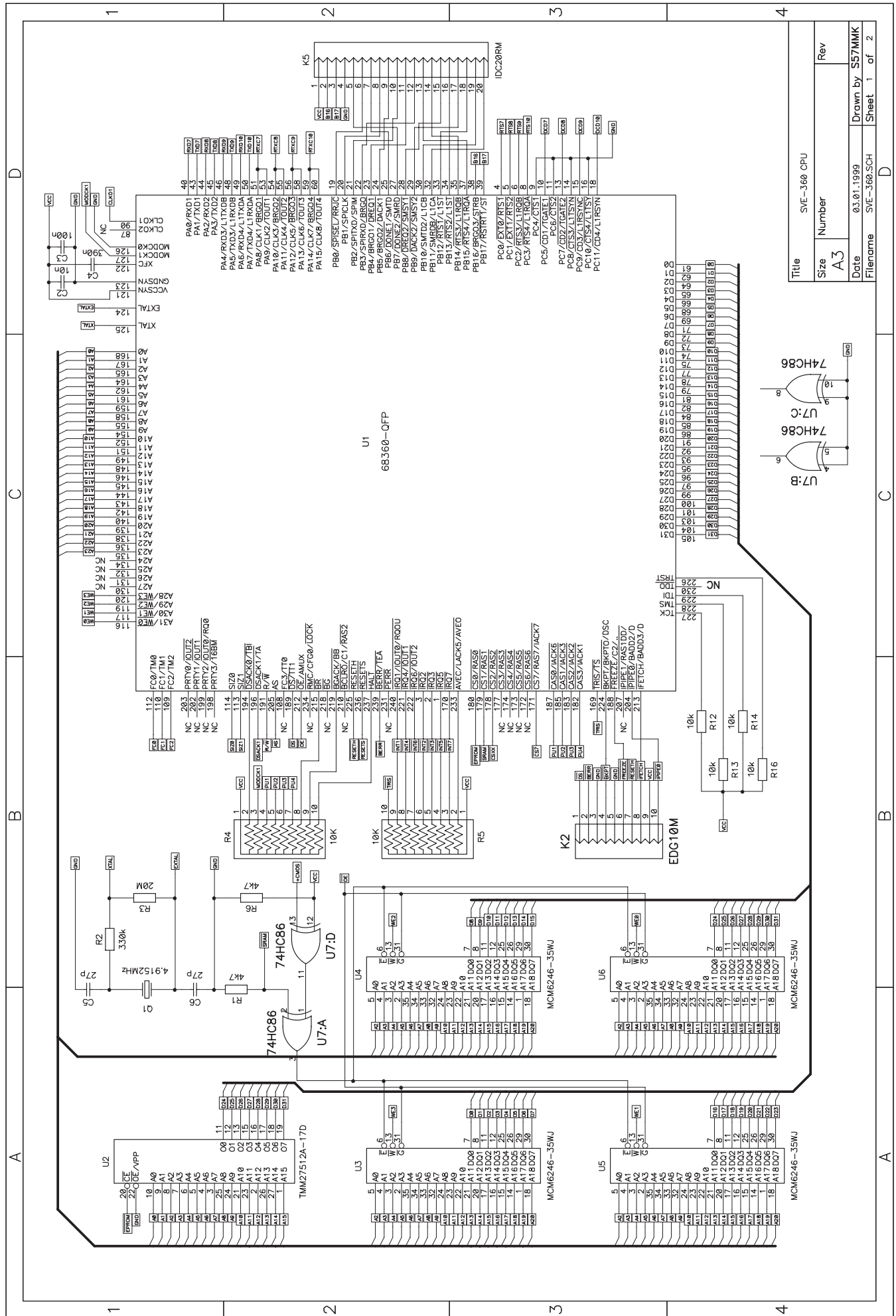
tiskanino, prispajkamo vogalne pine vezja ali obtežimo, nato vse pine premažemo z tenkim slojem spajkalne paste primerne za spajkanje z vročim zrakom. Potem s plamenom plinskega spajkalnika enakomerno segrevamo pine vezja do temperature približno 200°C. Pri tem moramo biti izredno pazljivi, da pinov ali vezja ne pregrejemo ter da ne poškodujemo vezic na tiskanini. Tiskanino za procesorski modul SV360 je načrtoval Franci (S51RM) in Sine (S53RM).

Tako izdelan procesorski modul za Super Vozelj vstavimo v SV vodilo na pozicijo predvideno za (staro) DMA ploščo. Vozlišče lahko deluje samo s procesorsko ploščo in štirimi modemi, pri čemer je potrebno ustrezno prilagoditi tabelo parametrov kanalov v programu SV360dx (parblok). V SV vodilo lahko vstavimo še šest kanalno SCC (Z8530) ploščo. Tako lahko dobimo maksimalno deset kanalno vozlišče. Dodatni RAM moduli na SV vodilu niso potrebni. Starih DMA modulov ni možno vstaviti v vozlišče z procesorsko ploščo SV360. Priključitev analognih vhodov ali slikolova (še) ni programsko podprta v programu verzije SV360dx.

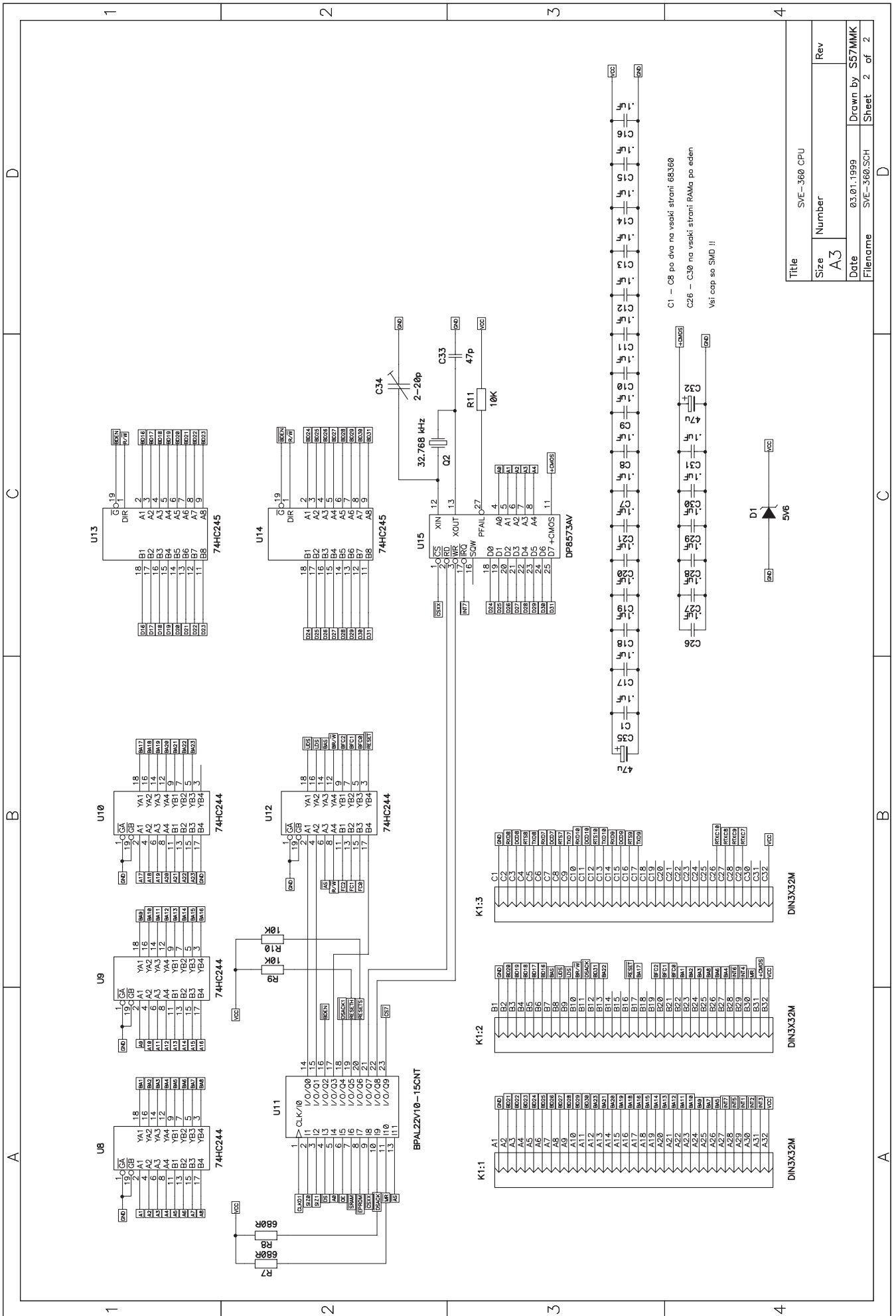
SuperVozelj z vgrajenim procesorskim modulom SV360 omogoča priključitev štirih hitrih PSK postaj s sprejemljivo majhno izgubo okvirjev pri sprejemu zaradi zasedenosti glavnega procesorja (pod 0.1 promila).

### **Literatura:**

1. Motorola: MC68360 Quad Integrated Communication Controller, User's Manual,
2. Matjaž Vidmar: Packet-radio vozlišče SuperVozelj (2), CQ ZRS št. 3/1995,



Title	SVE-360 CPU
Size	Number
A3	
Date	03.01.1999
Filename	SVE-360.SCH
Sheet	1 of 2



Title	SVF-360 CPU		
Size	Number	Rev	
A3			
Date	03.01.1999		
Filename	SVF-360.SCH		
Sheet	2	of	2

C1 - C8 po dva na vsaki strani 68360  
 C26 - C30 na vsaki strani RAMa po eden  
 Vsi cap so SMD !!