

# **Il dedispersore digitale del sistema "Pulsar" di Medicina**

*A. Maccaferri, N. D'Amico*

Rapporto interno IRA 198/95

- Introduzione
- Descrizione del dedispersore
- Registri di configurazione del dedispersore
- Procedura d'uso
- Schemi elettrici e descrizione delle varie schede:
  - Buffer di memoria*
  - Sommatore*
  - Controller*
  - Backplane*
  - Dedisp-Interface*
- Software di test
- Bibliografia

## INTRODUZIONE

Per osservare le pulsar a frequenze radio occorre campionare il segnale RF nel piano Frequenza-Tempo con risoluzione sufficiente a rimuovere l'effetto dispersivo. La propagazione degli impulsi nel mezzo interstellare produce un delay dato dalla formula:

$$\Delta t = \frac{DM}{1.2 \cdot 10^{-4}} \frac{\Delta \nu}{\nu^3} \quad \text{dove} \quad \begin{array}{ll} DM = \text{pc cm}^3 & \Delta \nu = \text{MHz} \\ \nu = \text{MHz} & \Delta t = \text{sec} \end{array}$$

La fig.1 mostra per esempio l'effetto nella pulsar PSR 0833-45 osservata a 400MHz.

Il campionamento in frequenza nel sistema pulsar di Medicina é fatto attraverso l'uso di uno spettrometro analogico basato su 128 filtri da 32 KHz. (Rif.1)

per dedispersare gli impulsi, i 128 segnali analogici provenienti dal banco di filtri sono rivelati e campionati individualmente (Rif.2) e poi sommati *off-line* con un opportuno delay. Nel caso della ricerca di pulsar, l'algoritmo di dedispersione viene applicato ripetutamente per un numero NDM di valori di "prova" della misura di dispersione DM.

Questa operazione comporta la lettura, lo spaccettamento e la somma dei dati impacchettati nella matrice Frequenza-Tempo ed é normalmente molto pesante in termini di tempo CPU richiesto.

Nell'acquisizione dati del sistema pulsar di Medicina (fig.2), i singoli canali sono digitalizzati ad 1 bit e l'impacchettamento é fatto sulla basi di 16 canali/word. Questa scelta é efficiente dal punto di vista dei requisiti di trasferimento e memorizzazione dei dati, ma produce un notevole appesantimento nella lettura e spaccettamento dei dati al momento dell'analisi. La lettura, lo spaccettamento e la somma "software" dei dati risulta alquanto inefficiente, ma puó essere semplificata se effettuata a livello hardware su un banco di memoria opportunamente architettato.

Il dedispersore digitale descritto in questa nota tecnica é basato su questa idea e consiste essenzialmente di un banco di memoria di 16MB indirizzabile ad 1 bit. In questa memoria possono essere copiati i dati corrispondenti ad una acquisizione di 1Msample in tempo risolto su 128 canali di frequenza. La lettura dei singoli campionamenti avviene senza spaccettamento (dato che la memoria é ad 1 bit), ed é quindi molto efficiente. Dopo avere estratto per ogni sample i 128 canali da sommare (128 bit), l'hardware utilizza delle *look-up table* in cascata configurate per eseguire direttamente la somma. In questo modo il dedispersore é in grado di fornire in uscita, la serie temporale "*dedispersa*" (1Msample x 128 canali) in un tempo di circa 500 mS. La stessa operazione effettuata su un calcolatore di tipo SPARC-2 richiederebbe circa 20 sec di CPU.

ve1a 400Mhz Pfold= 89.299221ms DM= 69.1

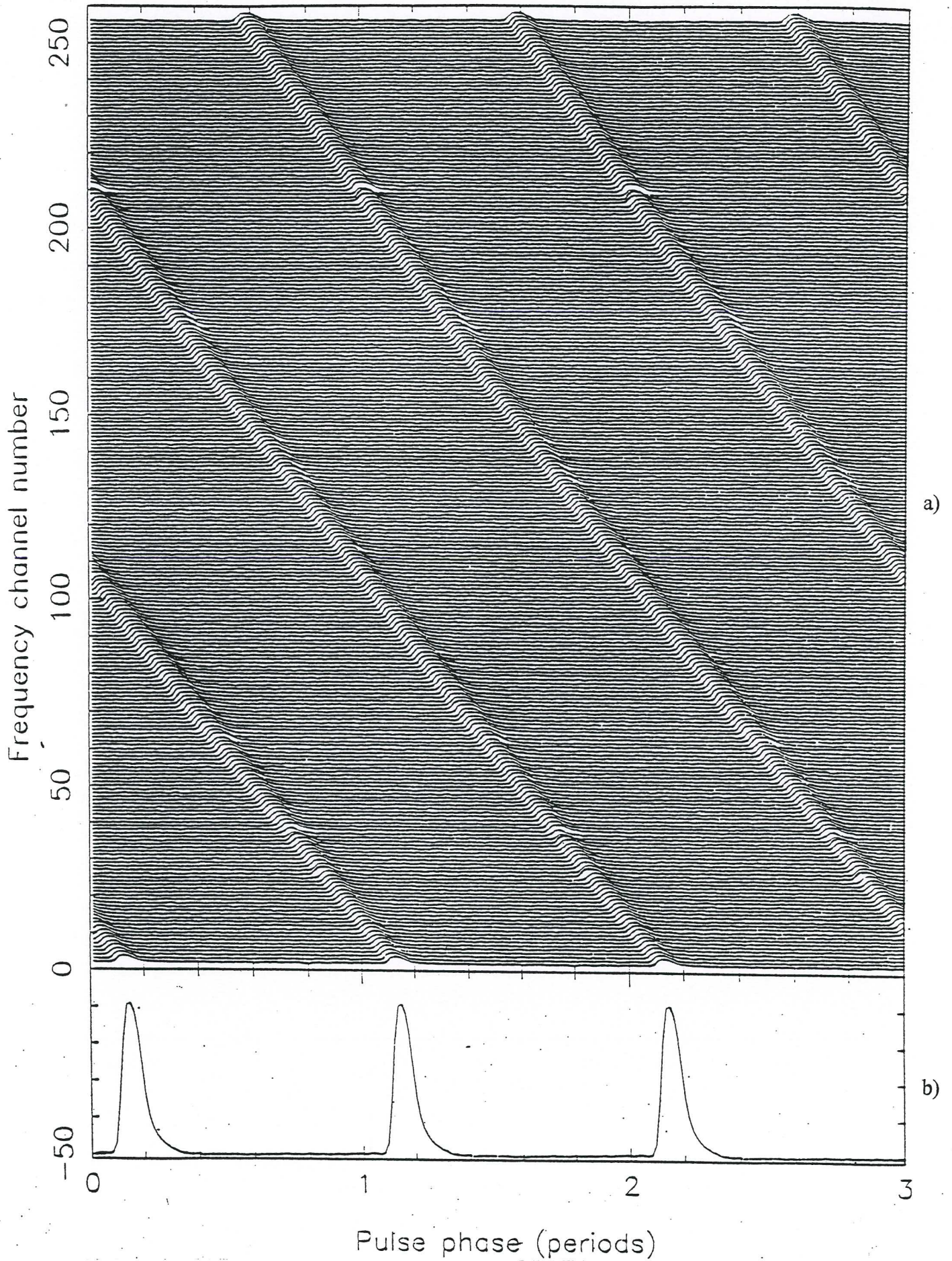


Fig.1 PSR 0833-45 osservata a 400MHz.

a) Osservazione degli impulsi risolti in tempo e frequenza.

b) Impulso dedisperso ed integrato.

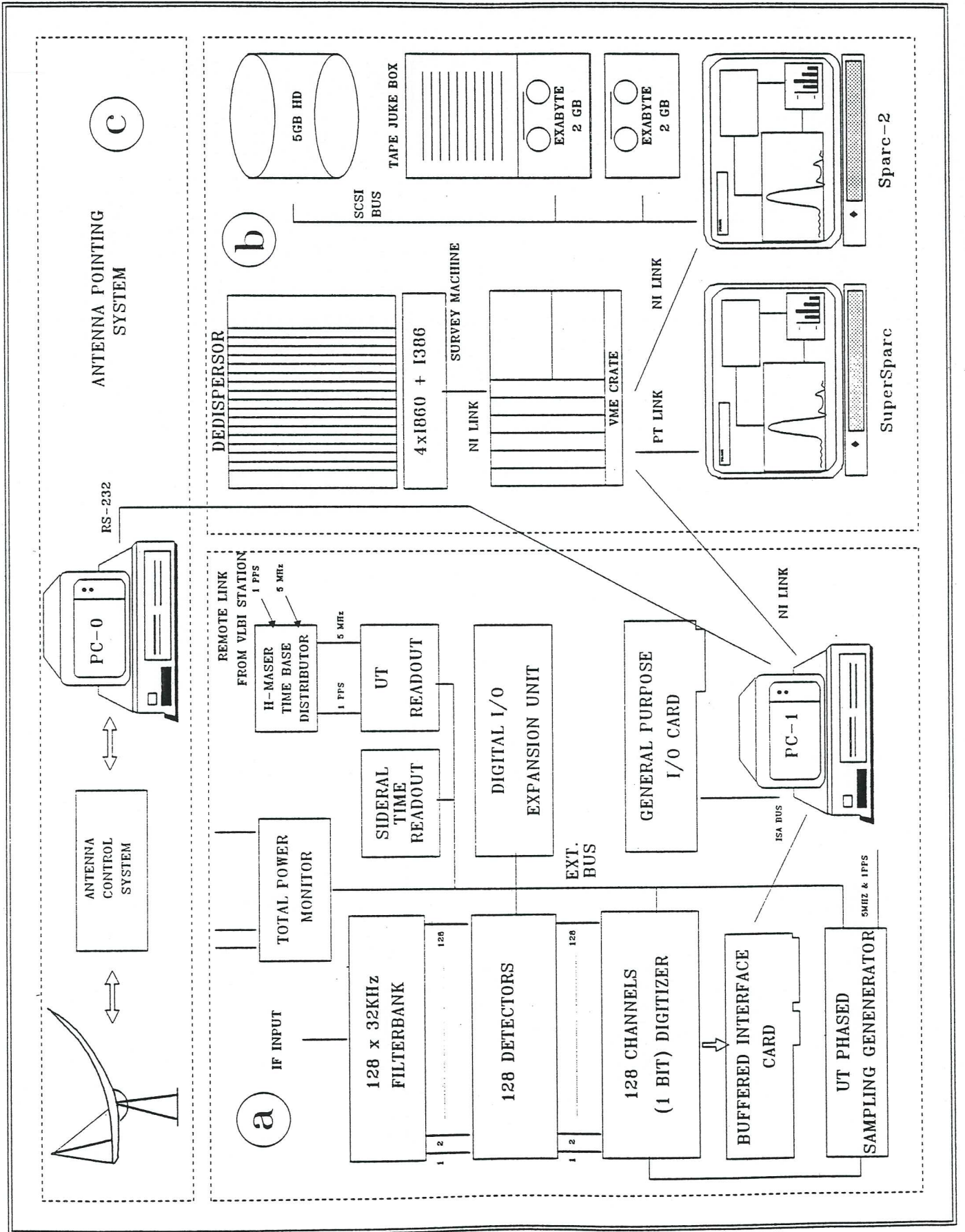
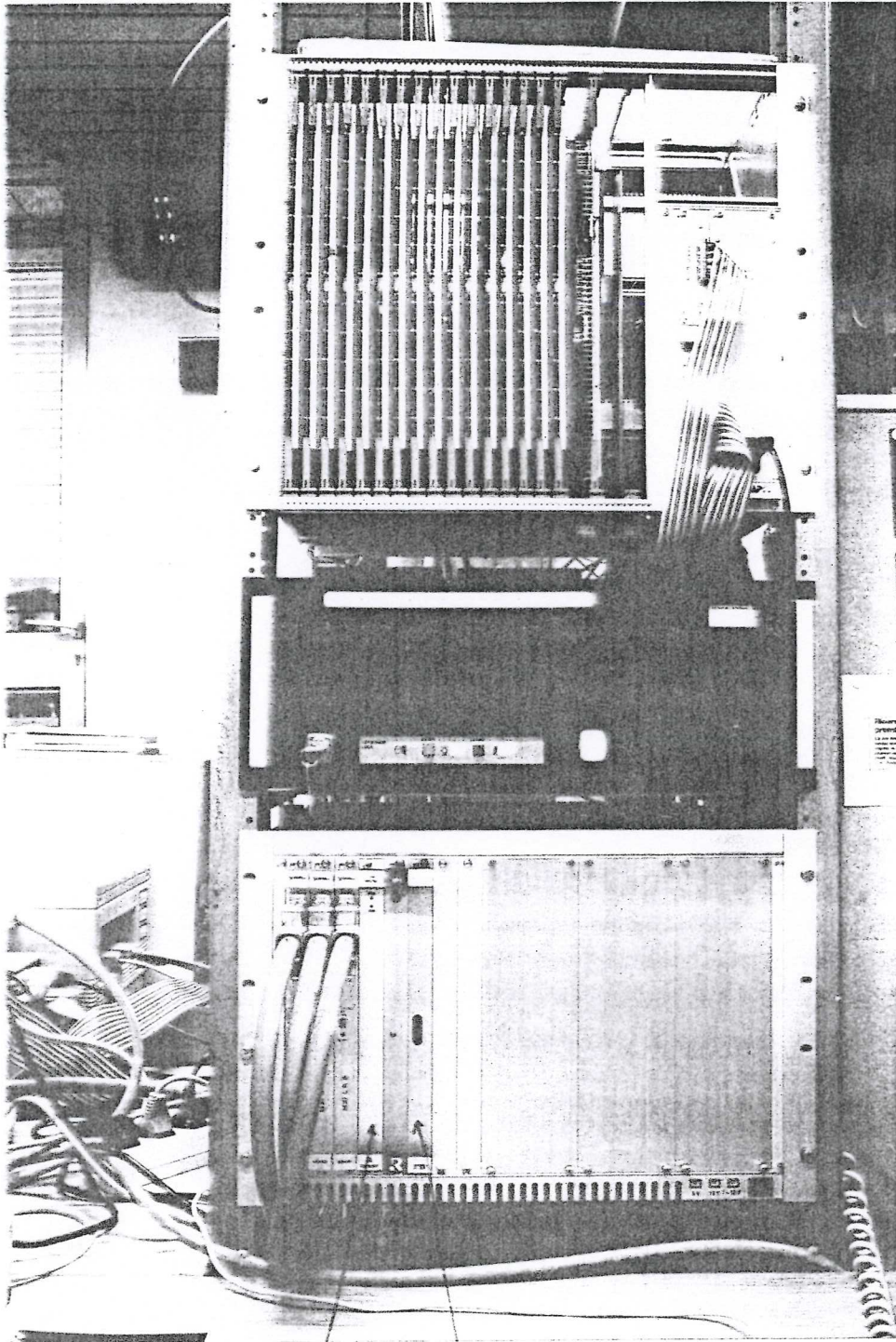


Fig.2 Sistema Pulsar di Medicina. a) Sottosistema di acquisizione dati b) Sottosistema di analisi e memorizzazione c) Sistema di puntamento

# THE SURVEY MACHINE



Digital  
dedispersor

4 x i860  
vector  
processor

VME Crate

## Descrizione del dedispersore

Il buffer di memoria del dedispersore é suddiviso in 16 schede di formato triplo Euro lungo (366.8x220). Ciascuna scheda (Fig.3), é composta di 8 buffer di memoria (4 x 256Kx1 static ram cadauno) e di 8 contatori di indirizzo a 20 bit (16 dei quali programmabili). Una scheda sommatore (adder) ed un controller completano il dedispersore (fig.4).

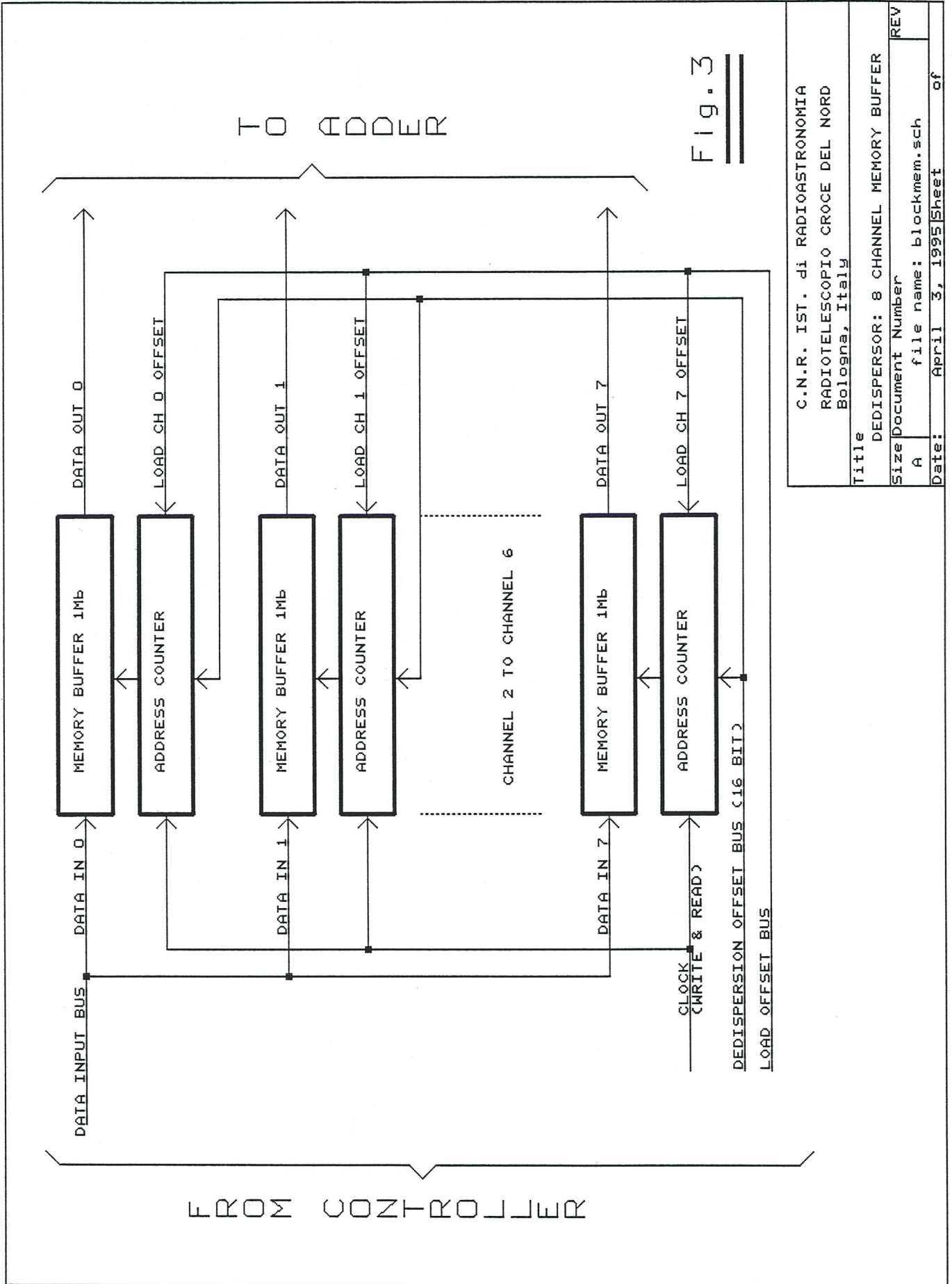
Il controller si interfaccia esternamente con una scheda posta nel bus del PC (Dedisp. Interface) e con la scheda multiplexer, mentre internamente attraverso il backplane implementato nel cestello del dedispersore colloquia con i buffer di memoria e con la scheda sommatore. Attraverso il link "Controller-Dedisp.Interface" si possono trasferire i dati da dedisperdere nei buffer, configurare il dedispersore, rileggere i dati dedispersi ed effettuare vari test diagnostici;

il link con il multiplexer invece permette solamente la lettura dei dati dedispersi. Uno slot libero a lato del controller principale é già predisposto per alloggiare un secondo controller che potrà consentire l'accesso al dedispersore anche a schede poste nel VME o ad altri dispositivi. A questo scopo é già stato implementato sul controller principale un apposito circuito che effettua la funzione di semaforo.

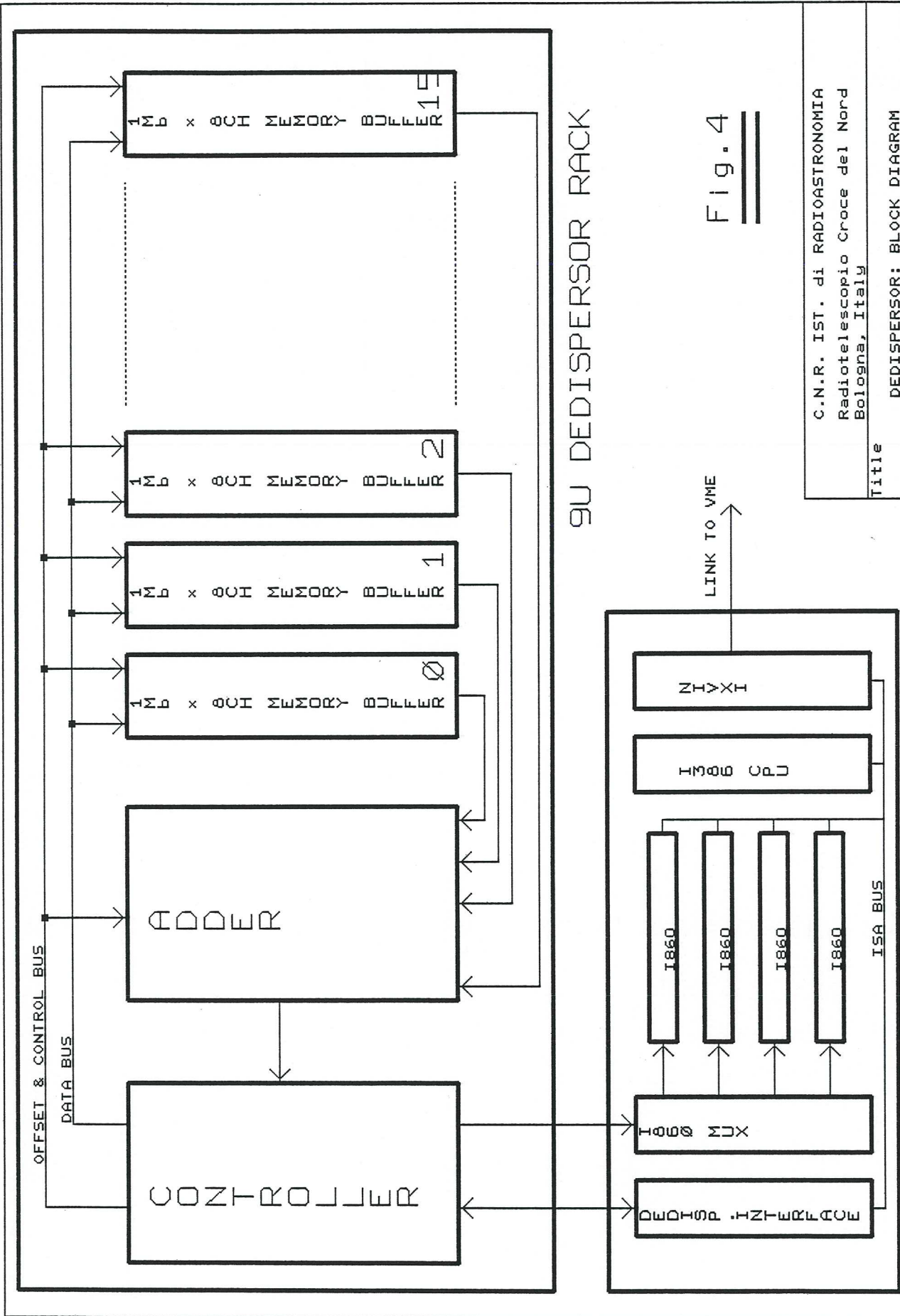
L'architettura adottata prevede che la scrittura dei dati in memoria, dei 128 diversi valori di dedispersione e la lettura dei dati dal sommatore, sia effettuata in modo blocco con postincremento automatico, ciò permette di sfruttare la massima velocità di trasferimento consentita dal BUS ISA del P.C. e comunque di ogni dispositivo al quale possa essere interfacciato.

Il sommatore non é in realtà una semplice tabella perchè ciò avrebbe richiesto una singola memoria con campo di indirizzamento pari a  $2^{128}$  Byte, (i più grossi dispositivi esistenti hanno la taglia di  $2^{15}$ Byte), siamo quindi ricorsi ad una architettura a 3 stadi (fig.5), utilizzando 12 EPROM da 128KByte, raggiungendo comunque considerevoli velocità nel calcolo della somma (1 dato disponibile ogni 500 microsecondi inclusi i tempi di trasferimento dalla memoria al sommatore e da questo all'utilizzatore).

Il primo stadio è sostanzialmente costituito da sommatore di 15 parole da 1 bit, ciascuna di peso  $2^0$ , il secondo ed il terzo stadio sono composti da sommatore di un numero di parole vario, ma soprattutto di peso variabile, il tutto ottimizzato considerando che per ogni sommatore si possono avere al massimo 15 parole in ingresso e che il risultato non deve superare gli 8 bit.



C.N.R. IST. di RADIOASTRONOMIA RADIOTELESCOPIO GROCE DEL NORD Bologna, Italy	
Title	DEDISPERSOR: 8 CHANNEL MEMORY BUFFER
Size	Document Number
A	file name: blockmem.sch
Date:	April 3, 1995
Sheet	of
REV	



9U DEDISPERSOR RACK

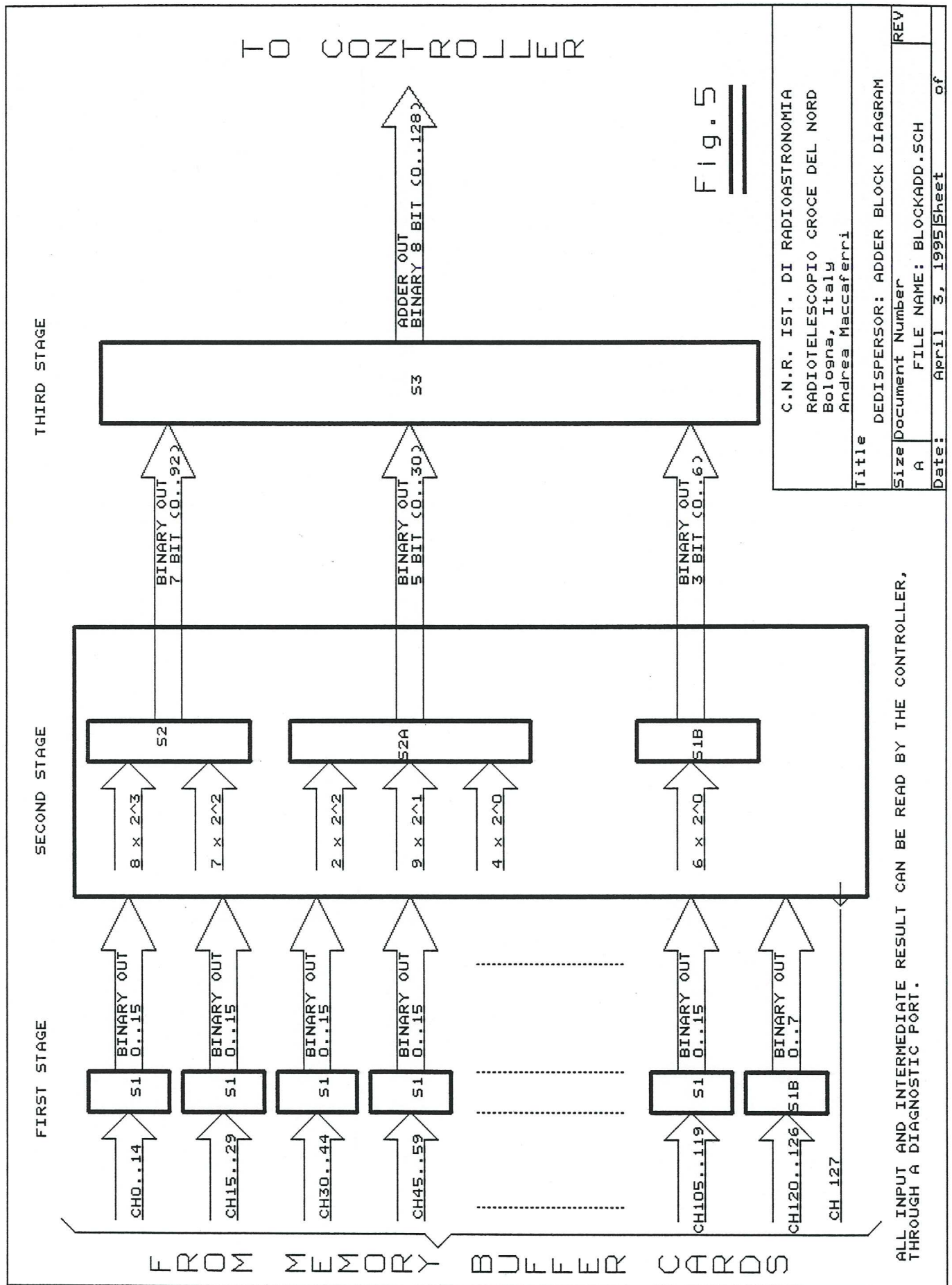
Fig. 4

C.N.R. IST. di RADIOASTRONOMIA  
 Radiotelescopio Croce del Nord  
 Bologna, Italy

Title	DEDISPERSOR: BLOCK DIAGRAM
Size	Document Number
A	file name: blockded.sch
Date:	April 3, 1995
Sheet	of
REV	

PERSONAL COMPUTER ISA BUS





C.N.R. IST. DI RADIOASTRONOMIA  
 RADIOTELESCOPIO CROCE DEL NORD  
 Bologna, Italy  
 Andrea Maccaferri

Title  
 DEDISPERSOR: ADDER BLOCK DIAGRAM

Size Document Number  
 A FILE NAME: BLOCKADD.SCH REV

Date: April 3, 1995 Sheet of

ALL INPUT AND INTERMEDIATE RESULT CAN BE READ BY THE CONTROLLER, THROUGH A DIAGNOSTIC PORT.

## Registri di configurazione del dedispersore

Tramite la scheda "Dedisp.Interface" collegata al Controller attraverso un link bidirezionale, é possibile gestire il dedispersore ed effettuare tutte le operazioni necessarie: scrivere e leggere la memoria, leggere i dati già dedispersi, configurare il tutto in base al numero di canali e al fattore di dedispersione desiderato ed eseguire varie operazioni di diagnostica.

Il controller dal punto di vista della programmazione dal PC attraverso la scheda Dedisp.Interface equivale a 5 registri di scrittura/lettura a 16 bit. Base é l'indirizzo di I/O a cui si configura la scheda stessa (attualmente 280hex, 640 decimale)

Indirizzo	Scrittura/Lettura	Descrizione
Base	S	Scrittura memoria dedispersore con postincremento.
Base	L	Lettura dati dal dedispersore con postincremento (usare solo il byte basso).
Base+2	S	Scrittura offset di dedispersione di ciascun canale con postincremento. (iniziando dal canale 0)
Base+2	L	Lettura gruppo di un gruppo di canali o stadi intermedi Adder con postincremento. (diagnostica)
Base+4	S	Scrittura pattern di abilitazione canali a gruppi di 8.
Base+4	L	Lettura flag di status
Base+6	S	Selezione dati di lettura per diagnostica e gruppi di canali in scrittura.
Base+6	L	Non usato
Base+8	S	Bit di controllo
Base+8	L	Non usato

### Base+0 :

In scrittura permette di riempire il buffer di memoria con i dati da dedisperdere, in modo sequenziale (Campionamento dopo campionamento, considerando ogni campionamento (128 ch) composto dalla sequenza di 8 word a 16 bit; viene effettuato un postincremento automatico dell'indirizzamento del buffer di memoria dopo la scrittura di ogni word).

Bit Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	c15	c14	c13	c12	c11	c10	c9	c8	c7	c6	c5	c4	c3	c2	c1	c0
1	c31	c30	c29	c28	c27	c26	c25	c24	c23	c22	c21	c20	c19	c18	c17	c16
2	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
3	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
4	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
5	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
6	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
7	c127	c126	c125	c124	c123	c122	c121	c120	c119	c118	c117	c116	c115	c114	c113	c112

In lettura permette di accedere al risultato della dedispersione, in modo sequenziale con postincremento automatico ad ogni lettura. (E' valida solo la parte bassa della word).

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x	Add	Add	Add	Add	Add	Add	Add	Add
									7	6	5	4	3	2	1	0

#### Base+2 :

E' il registro dove scrivere l'offset di ciascun canale (in unità di campionamenti) per la dedispersione. Il registro é a 16 bit, si può quindi impostare un offset da 0 a 65535 campionamenti. Prima di scrivere i 128 valori di offset, occorre inizializzare a zero il contatore di canale, effettuando le transizioni 1.0.1 sul bit di controllo Init\_Ded\_Counter.

Ad ogni scrittura, il contatore di canale viene automaticamente postincrementato, la scrittura dei 128 offset può quindi avvenire sequenzialmente iniziando dal canale 0 fino al canale 127.

In lettura é il registro dove andare a leggere o il pattern in uscita ad un gruppo di 16 canali, o le uscite intermedie del sommatore, a scopo di diagnostica ( dopo ogni lettura avviene il postincremento del buffer del dedispersore).

L' informazione a cui si accede viene selezionata tramite il registro Base+6 (MA[0..3]), i dati letti in funzione di queste linee sono indicati nella tabella.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Porta Base+6																	
Descrizione																	
0	canali	c15	c14	c13	c12	c11	c10	c9	c8	c7	c6	c5	c4	c3	c2	c1	c0
1	canali	c31	...	...	...	...	...	...	...	...	...	...	...	...	...	...	c16
2	canali	c47	...	...	...	...	...	...	...	...	...	...	...	...	...	...	c32
3	canali	c63	...	...	...	...	...	...	...	...	...	...	...	...	...	...	c48
4	canali	c79	...	...	...	...	...	...	...	...	...	...	...	...	...	...	c64
5	canali	c95	...	...	...	...	...	...	...	...	...	...	...	...	...	...	c80
6	canali	c111	...	...	...	...	...	...	...	...	...	...	...	...	...	...	c96
7	canali	c127	...	...	...	...	...	...	...	...	...	...	...	...	...	...	c112
8	Primo stadio	P15	P14	P13	...	...	...	...	...	...	...	...	...	...	...	...	P0
9	Primo stadio	P31	...	...	...	...	...	...	...	...	...	...	...	...	...	...	P16
10	Secondo stadio	x	S14	...	...	...	...	...	...	...	...	...	...	...	...	...	S0
11	Risultato e Prino stadio	x	x	x	x	x	P34	P33	P32	R7	R6	R5	R4	R3	R2	R1	R0

#### Base+4 :

Il pattern scritto nel registro Base+4 abilita a gruppi di 8 canali le sottobande da utilizzare nella dedispersione secondo la seguente tabella (Sottobanda abilitata con bit corrispondente a 0).

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Canali	120	104	88	72	57	40	24	8	112	96	80	64	48	32	16	0
	127	111	95	79	63	47	31	15	119	103	87	71	55	39	23	7

In lettura permette di conoscere lo stato di possesso del dedispersore.

Bit	15..2	1	0
Descrizione	x	Master a 0 indica che sono padrone del dedispersore	Busy a 0 indica che il dedispersore é in uso

**Base+6 :**

In scrittura serve per indirizzare quali dati leggere per diagnostica secondo la tabella 3, tramite le linee MA0..MA3; e per impostare in blocchi di 16 il numero di canali in fase di riempimento del buffer, tramite le linee WSEL0..WSEL2.

Bit	15	14	13	12	11	10	9	8	7	6..1	0
	x	WSEL2	WSEL1	WSEL0	MA3	MA2	MA1	MA0	x	x	x

In lettura non é utilizzato.

**Base+8 :**

In scrittura permette di inizializzare alcune funzioni e di controllare il possesso del dedispersore, permettendo cosí l'accesso al dedispersore ad altri sistemi durante i tempi morti, per un piú efficiente utilizzo.

Bit	15..8	7	6	5	4	3	2	1	0
Descrizione	x	Init Ded. Counter	Offset Write	Memory Write	Master Reset	Ctrl_Clr	Ctrl_Rqst	x	Init Wcount

In lettura non é utilizzato.

## Procedura d'uso

- Inizializzazione:**
- ◆ Effettuare un master reset, facendo un impulso 1-0-1 sul segnale Master Reset del registro (Base+8) Nessuno ha il possesso del dedispersore.
- Riempimento:**
- ◆ Richiedere l' uso del dedispersore, generare un impulso 1,0,1 sul segnale Ctrl\_Rqst di Base+8.
  - ◆ Verificare che si abbia il possesso del dedispersore, leggendo il bit di status Master in Base+4
  - ◆ Programmare il numero di canali con i quali riempire il dedispersore, tramite i segnali WSEL0..WSEL2, in Base+6 (7 = 128 Ch, quindi tutte le schede; 6 = 112 Ch, quindi usa solo le schede 0..6 e 8..14 etc..)
  - ◆ Inizializzare il contatore di indirizzamento coppie di schede per la scrittura postincrementata (effettuare un impulso 1,0,1 sul segnale Init Wcount in Base+8)
  - ◆ Abilitare il Bus di scrittura dati in memoria e dell'offset. (Impostare a zero i segnali Memory Write e Offset Write in Base+8)
  - ◆ Azzerare l'offset dei 128 canali, scrivendo 128 volte 0 in Base+2 (scrittura postincrementata)
  - ◆ Riempire il buffer del dedispersore scrivendo i dati dei canali in parole di 16 bit in Base+0, secondo la sequenza illustrata in tabella 2 (Nell'eventualità che i dati siano meno di una potenza di 2, riempire di zeri fino alla potenza di 2 immediatamente superiore.)
  - ◆ Disabilitare il Bus di scrittura dati rimettendo a 1 il segnale Memory Write in Base+8
- Letture dati dedispersi:**
- ◆ Richiedere l' uso del dedispersore, generare un impulso 1,0,1 sul segnale Ctrl\_Rqst di Base+8.
  - ◆ Verificare che si abbia il possesso del dedispersore, leggendo il bit di status Master in Base+4
  - ◆ Abilitare il Bus di scrittura dell'offset. (Impostare a zero il segnale Offset Write in Base+8)
  - ◆ Azzerare il contatore di postincremento scrittura contatori offset di dedispersione (effettuare un impulso 1,0,1 sul segnale Init Ded.Counter in Base+8)
  - ◆ Scrivere in sequenza i 128 diversi valori di offset nei contatori di indirizzamento iniziando dal ch0.
  - ◆ Abilitare le sottobande di 8 canali desiderate mettendo ad 1 il bit corrispondente del registro Base+4
  - ◆ Disabilitare il Bus di scrittura dati e offset rimettendo a 1 il segnale Memory Write e Offset Write in Base+8
  - ◆ Leggere i dati dedispersi in Base+0 (lettura con postincremento automatico del buffer di memoria)
  - ◆ Cedere l'uso del dedispersore nel caso che si voglia concedere l'accesso al dedispersore al secondo controllore generando un impulso 1-0-1 sul segnale Ctrl\_Clr di Base+8
- Diagnostica**
- ◆ Effettuare l'inizializzazione ed il riempimento come sopra.
  - ◆ Richiedere l' uso del dedispersore, generare un impulso 1,0,1 sul segnale Ctrl\_Rqst di Base+8.
  - ◆ Verificare che si abbia il possesso del dedispersore, leggendo il bit di status Master in Base+4 in caso negativo attendere un tempo ragionevole, quindi rieffettuare la richiesta ed il controllo
  - ◆ Abilitare il Bus di scrittura dell'offset. (Impostare a zero il segnale Offset Write in Base+8)
  - ◆ Azzerare il contatore di postincremento scrittura contatori offset di dedispersione (effettuare un impulso 1,0,1 sul segnale Init Ded.Counter in Base+8)
  - ◆ Scrivere in sequenza i 128 diversi valori di offset nei contatori di indirizzamento iniziando dal ch0.
  - ◆ Abilitare le sottobande di 8 canali desiderate mettendo ad 1 il bit corrispondente del registro Base+4
  - ◆ Selezionare i canali da rileggere per testare il buffer di memoria o il risultato di uno stadio intermedio del sommatore, scrivendo il valore opportuno in Base+6 (Vedi tabella)
  - ◆ Disabilitare il Bus di scrittura dati e offset rimettendo a 1 il segnale Memory Write e Offset Write in Base+8
  - ◆ Leggere i dati desiderati alla porta Base+2 ( lettura con postincremento automatico del buffer di memoria )

## Schemi elettrici e descrizione delle varie schede

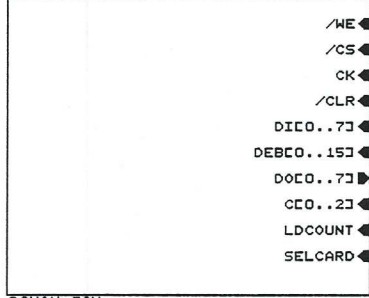
### *- Buffer di memoria*

La scheda buffer di memoria si compone di 8 sottoblocchi (Memory Channel) ciascuno corrispondente ad un canale ad un bit, da alcuni buffer U39, U40 ed U41 che limitano il carico della scheda verso il backplane per i segnali comuni ad 1 solo carico TTL, da un circuito di decodifica U42 che seleziona il contatore di indirizzo da presetare al valore richiesto per la dedispersione e da 2 monostabili che rendono visibile tramite dei led lo svolgimento di alcune funzioni sulla scheda quali la scrittura, il postincremento, la selezione di un contatore all'interno della scheda per la scrittura dell'offset di dedispersione e la abilitazione della sottobanda di 8 canali della scheda.

Ogni blocco Memory Channel é composto da un contatore di indirizzo a 20 bit, formato da 5 contatori 74HCT193 in cascata, presetabile ad un valore compreso fra 0 e 65535, da 4 chip di memoria ram da 256Kx1 per un totale di 1Mx1 e da un circuito di decodifica per la selezione dei chip di memoria. In origine il progetto prevedeva la possibilità di fare un clear di tutti i contatori di indirizzamento tramite un'unica operazione, ma purtroppo questi contatori hanno il clear attivo alto, sono quindi molto sensibili al rumore, si é quindi dovuto eliminare questa funzione, e presetare i contatori singolarmente a 0.

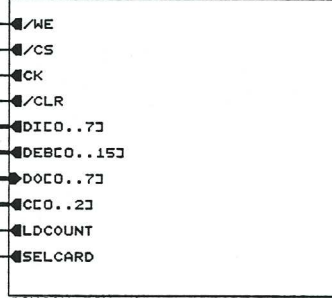
Nel dedispersore vengono utilizzate 16 di queste schede per un totale di 128 canali. Visto il numero relativamente considerevole di queste schede, é stato realizzato il circuito stampato effettuando anche la progettazione del *master*, é quindi agevole procedere ad una replica del dedispersore presso altri osservatori impegnati nelle osservazioni di pulsar.

MEMORY BUFFER 8 CANALI



BCHAN.SCH

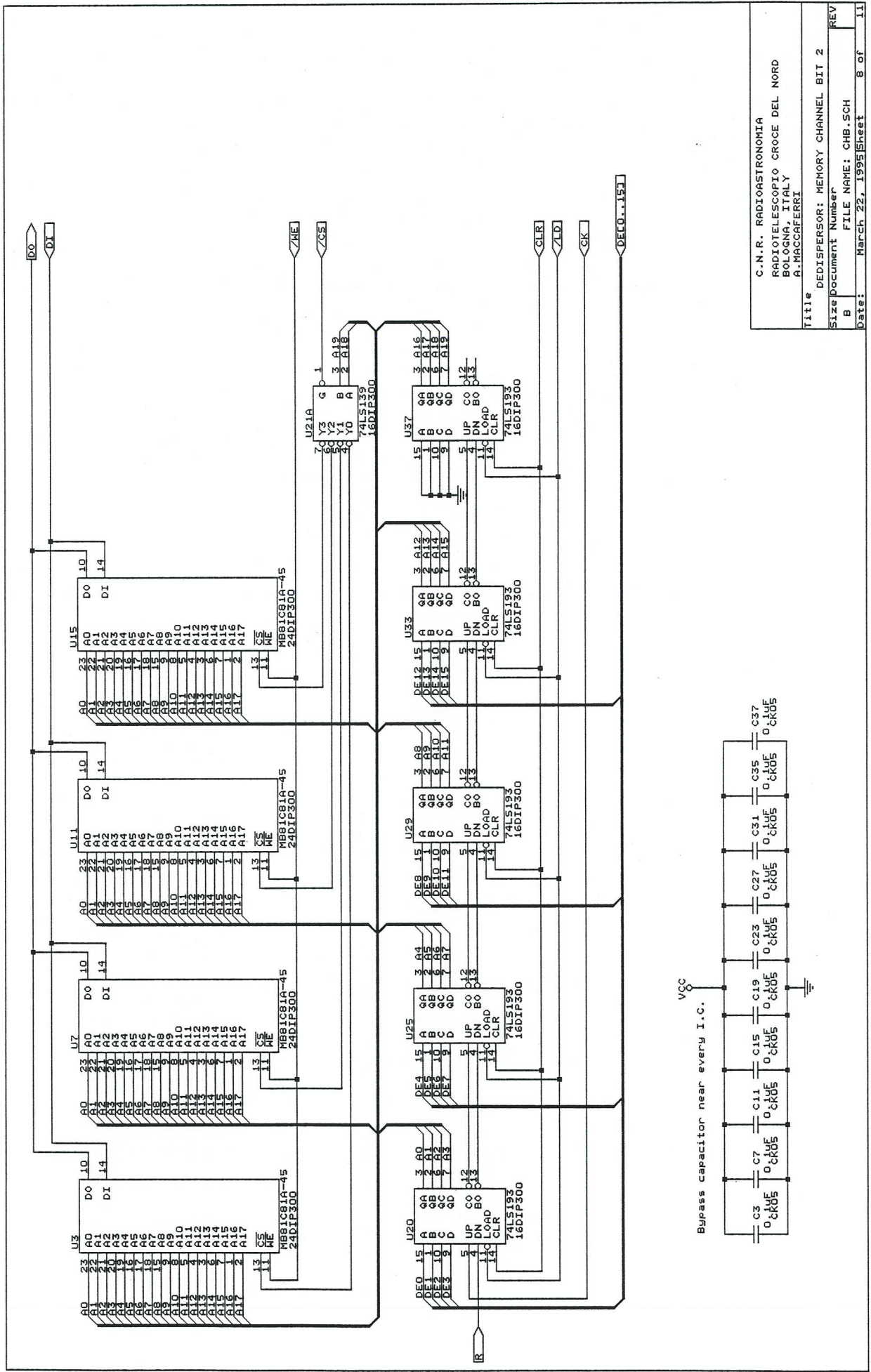
CONNETTORI MEMORY BUFFER 8 CHAN



BCHCON.SCH

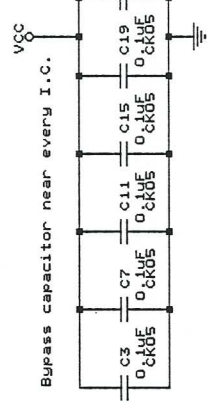
C.N.R. RADIOASTRONOMIA  
RADIOTELESCOPIO CROCE DEL NORD  
BOLOGNA, ITALY  
A.MACCAFERRI

Title		
DEDISP.:MEMORY BUFF. 8 CHANNEL CARD BLOCK		
Size	Document Number	REV
A	FILE NAME: MEMORY.SCH	
Date:	March 22, 1995	Sheet of

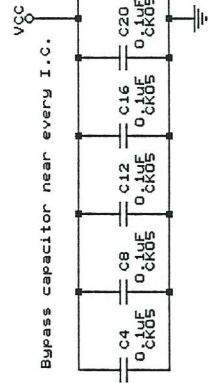
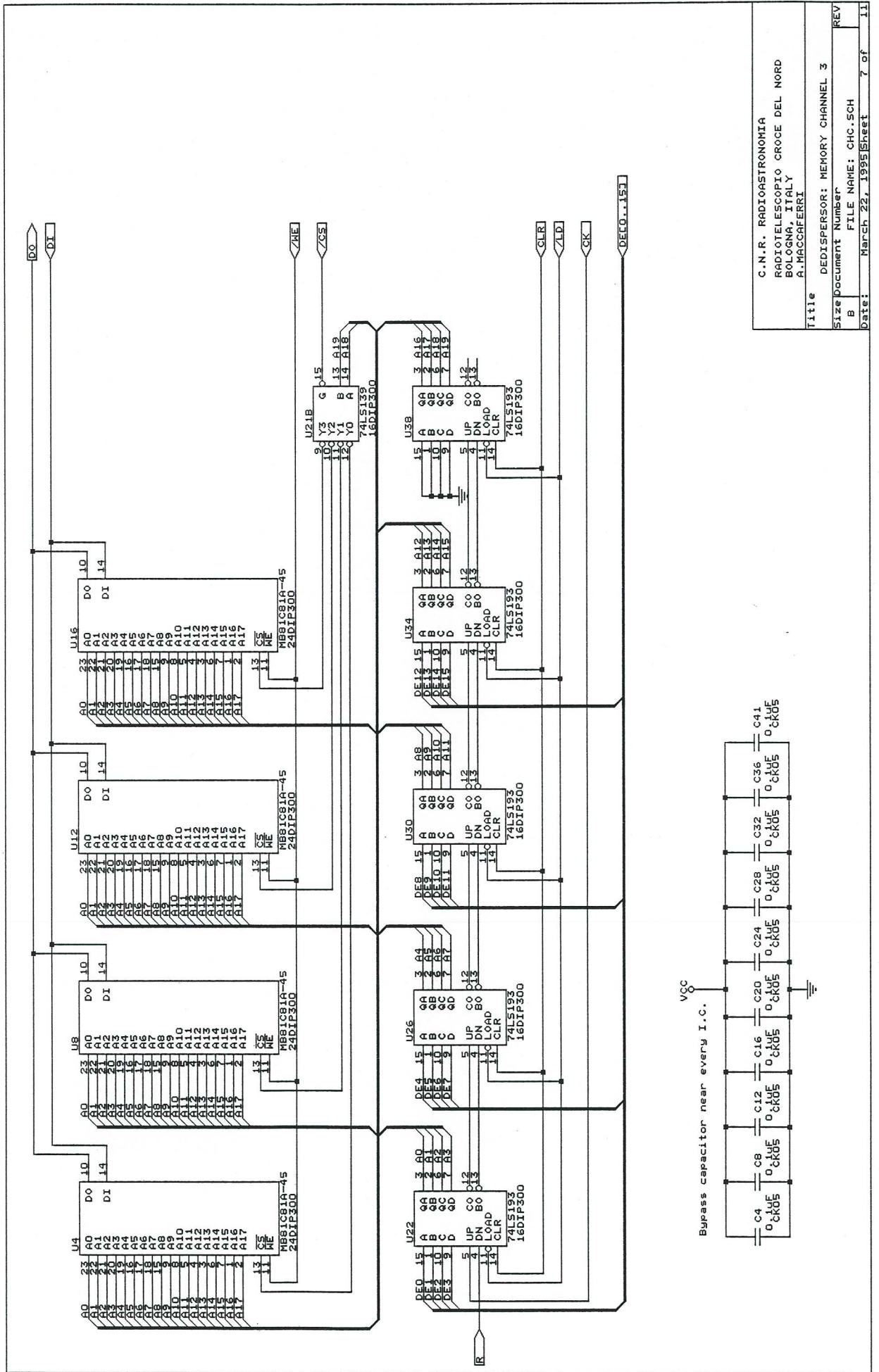


C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO CROCE DEL NORD  
 BOLOGNA, ITALY  
 A.MACCAFERRI

Title: DEDISPERSOR: MEMORY CHANNEL BIT 2  
 Size: Document Number: REV  
 B FILE NAME: CHB.SCH  
 Date: March 22, 1995 Sheet 8 of 11





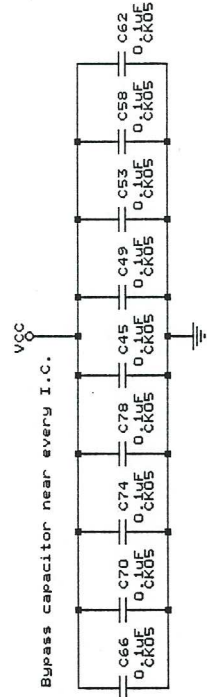
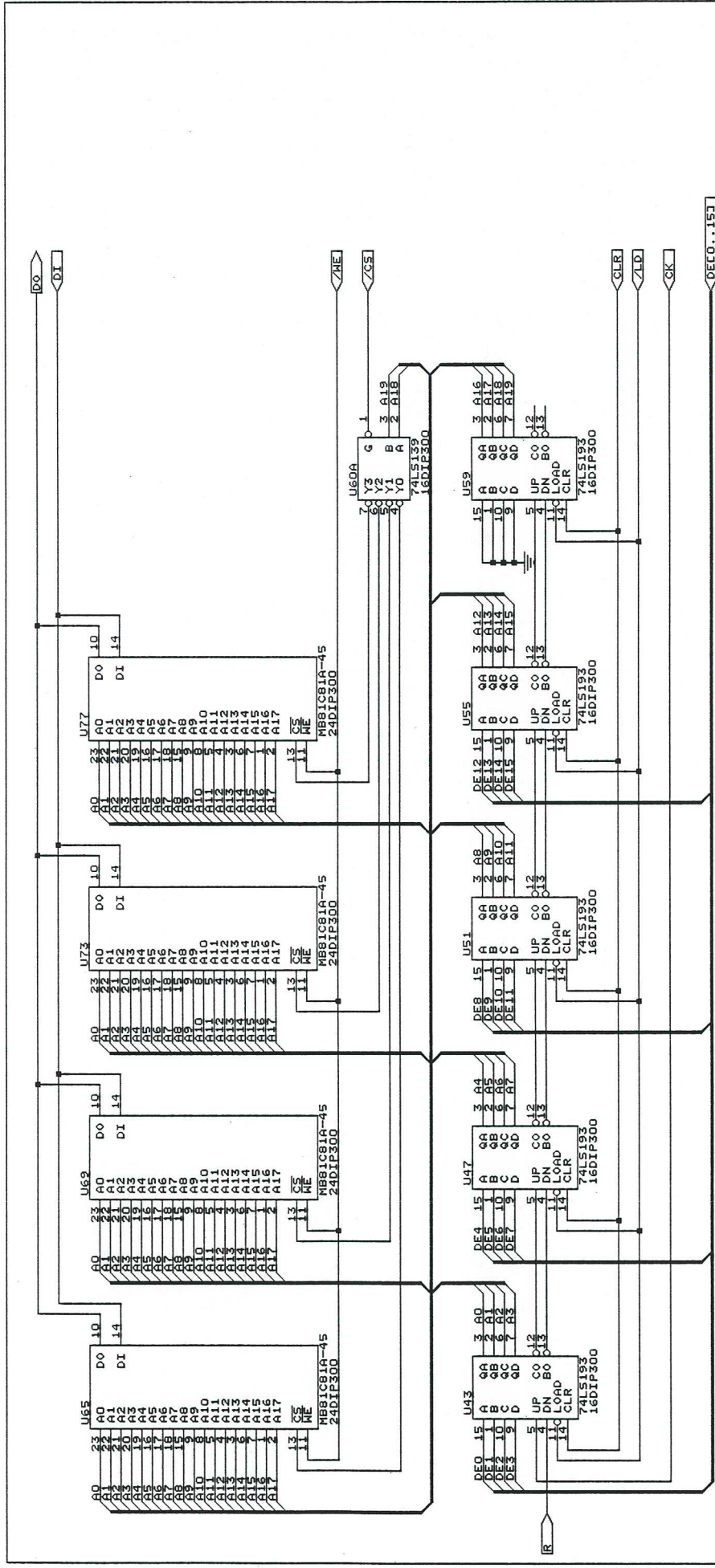


C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO GROCE DEL NORD  
 BOLOGNA, ITALY  
 A.MACCAFERRI

Title  
 DEDISPERSOR: MEMORY CHANNEL 3

Size  
 B  
 Document Number  
 FILE NAME: CHC.SCH

Date: March 22, 1995  
 Sheet 7 of 11

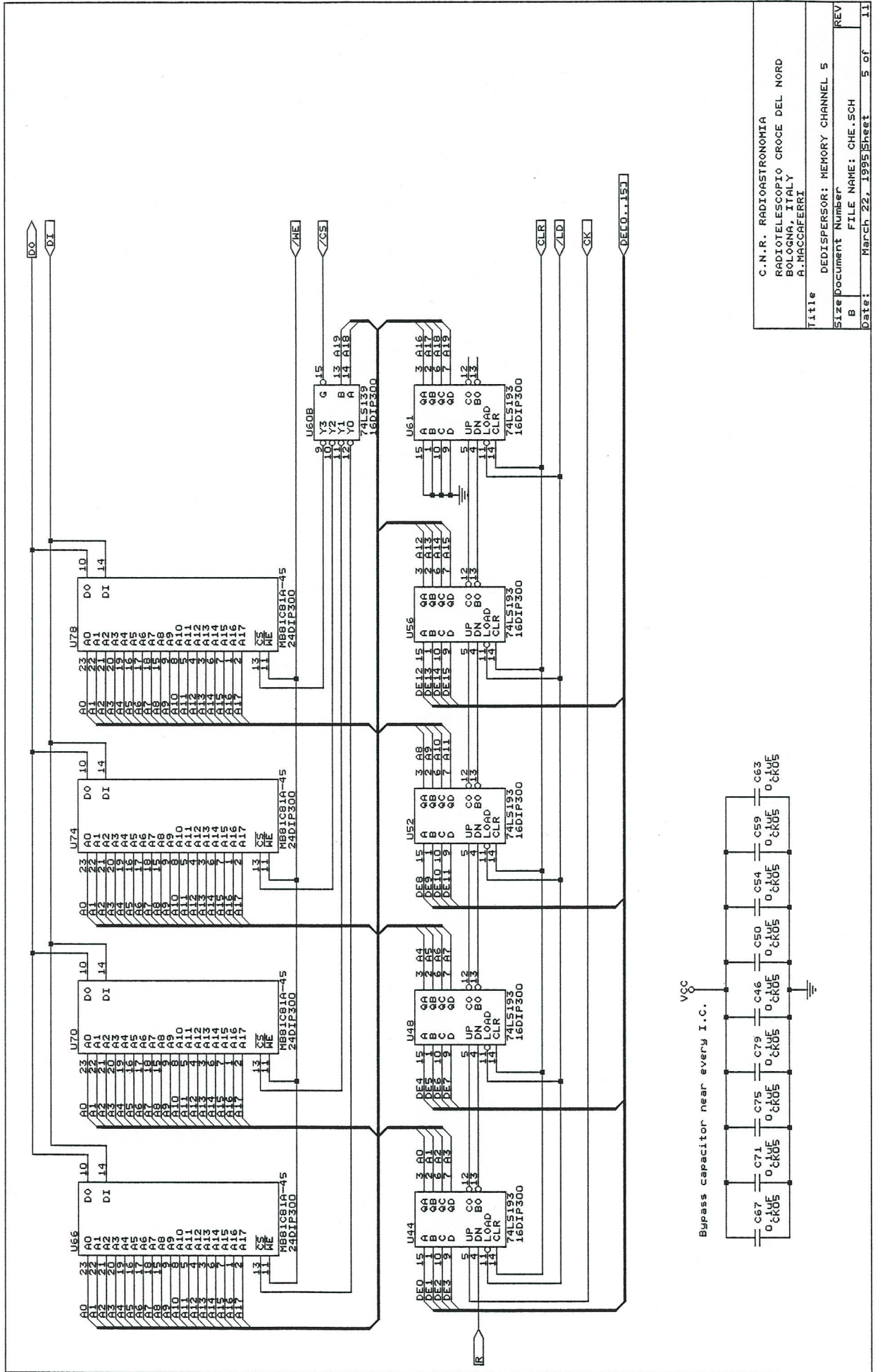


C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO CROCE DEL NORD  
 BOLOGNA, ITALY  
 A.MACCAFERRI

Title  
 DEDISPERSOR: MEMORY CHANNEL 4

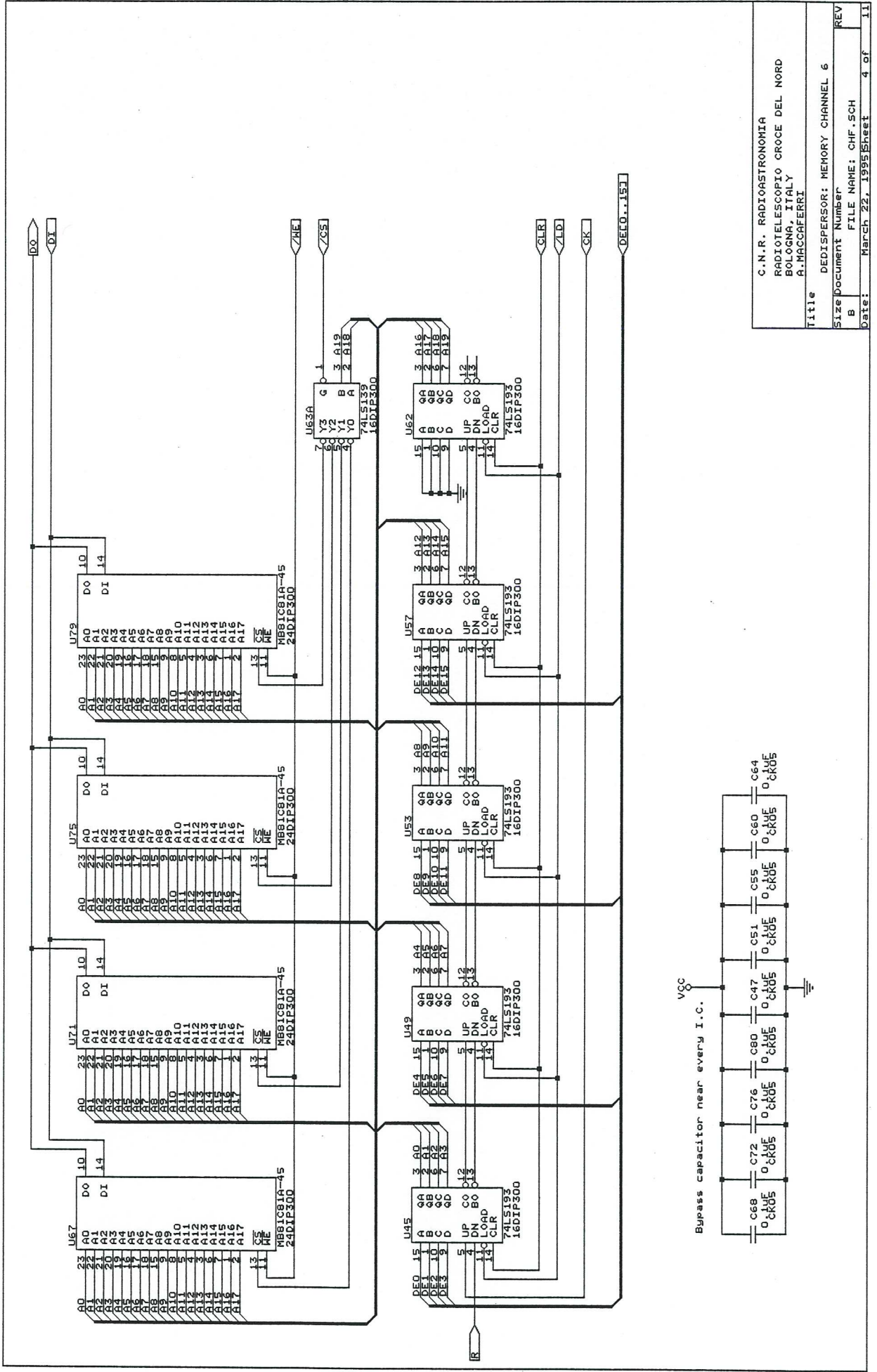
Size Document Number  
 B FILE NAME: CHD.SCH REV

Date: March 22, 1995 Sheet 6 of 11



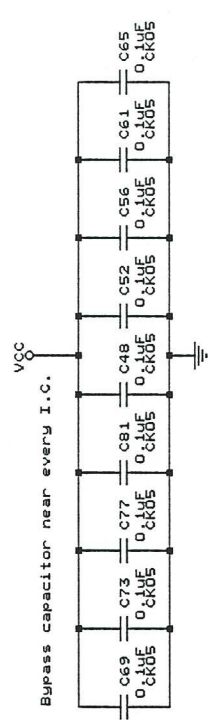
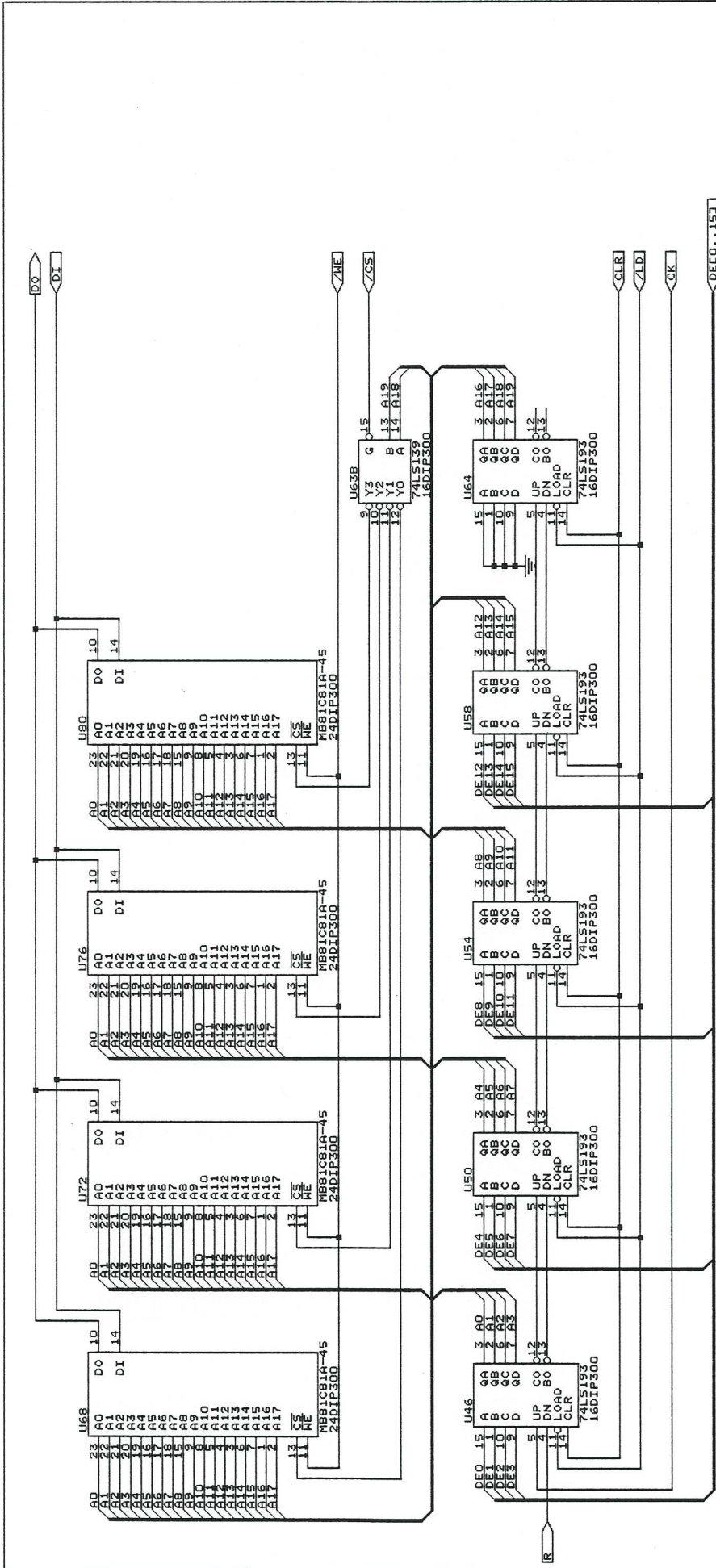
C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO GROCE DEL NORD  
 BOLOGNA, ITALY  
 A.MACCAFERRI

Title: DEDISPERSOR: MEMORY CHANNEL 5  
 Size: Document Number: REV  
 B FILE NAME: CHE.SCH  
 Date: March 22, 1995 Sheet 5 of 11



C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO CROCE DEL NORD  
 BOLOGNA, ITALY  
 A. MACCAFERRI

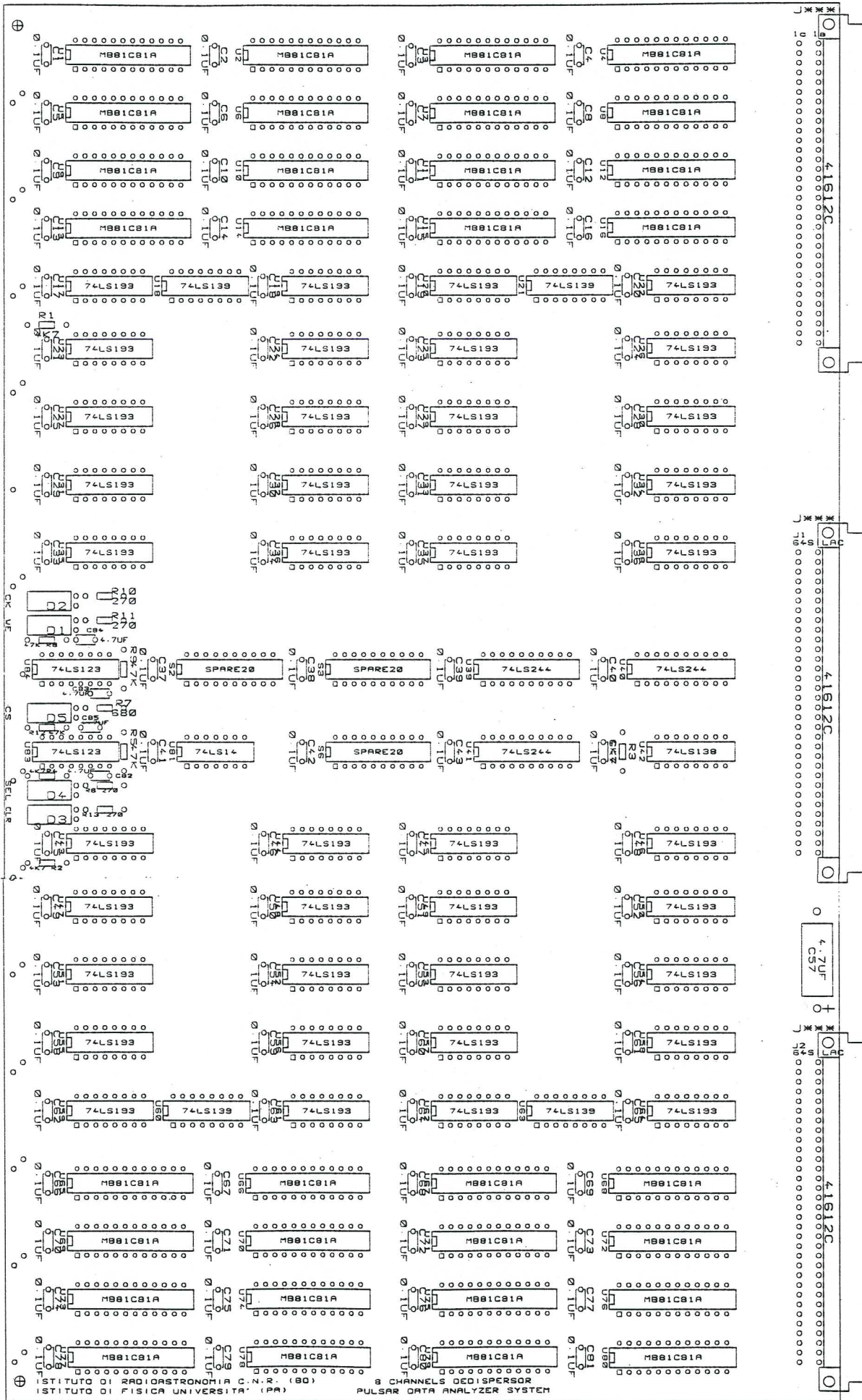
Title: DEDISPERSOR: MEMORY CHANNEL 6  
 Size: Document Number: \_\_\_\_\_ REV  
 B: FILE NAME: CHF.SCH  
 Date: March 22, 1995 Sheet 4 of 11



C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO GROCE DEL NORD  
 BOLOGNA, ITALY  
 A.MACCAFERRI

Title: DEDISPERSORE: MEMORY CHANNEL 7

Size: Document Number: B  
 FILE NAME: CHG.SCH  
 Date: March 22, 1995 Sheet 3 of 11



8 channel memory buffer printed circuit board layout

## - *Sommatore*

Nella scheda sommatore possiamo individuare 3 sezioni che compongono il sommatore vero e proprio (1° stadio, 2° stadio e 3° stadio), vi sono poi tutta una serie di buffer che permettono di leggere i vari bit in ingresso dai buffer di memoria o i risultati parziali dei vari stadi.

Il 1° stadio del sommatore é composto dalle Eprom U1..U8, il 2° stadio é composto da U9..U11, mentre U12 costituisce il 3° stadio. I file per la programmazione delle Eprom sono stati generati utilizzando il semplice programma in TurboBasic qui riportato. Ogni elemento n (n=0..14) del vettore Peso%(n) rappresenta il peso binario del segnale collegato al corrispondente pin A(n) di indirizzo della memoria.

```
REM Generazione file di programmazione per
REM eprom CY7C274 dell'ADDER
REM ingressi con peso diverso.
REM file dati per U1..U8
REM
CLS
DIM PESO%(15)
PESO%(0)=1
PESO%(1)=1
PESO%(2)=1
PESO%(3)=1
PESO%(4)=1
PESO%(5)=1
PESO%(6)=1
PESO%(7)=1
PESO%(8)=1
PESO%(9)=1
PESO%(10)=1
PESO%(11)=1
PESO%(12)=1
PESO%(13)=1
PESO%(14)=1
OPEN "stage1.DAT" FOR BINARY AS #1
FOR J%=0 TO 32767
  BINADDRESS$=BIN$(J%)
  REM PRINT J%,BINADDRESS$
  ADDENDO%=0
  FOR L%=0 TO (LEN(BINADDRESS$)-1)
    PUNTA%=LEN(BINADDRESS$)-L%
    REM PRINT J%,L%,BINADDRESS$, (VAL(MID$(BINADDRESS$,PUNTA%,1))),PESO%(L%)
    ADDENDO%=ADDENDO%+(VAL(MID$(BINADDRESS$,PUNTA%,1)))*PESO%(L%)
  NEXT L%
  ADDENDO$=CHR$(ADDENDO%)
  PRINT J%,BINADDRESS$,ADDENDO%
  SEEK #1,J%
  PUT$ #1,ADDENDO$
NEXT J%
CLOSE #1
PRINT "FILE WRITED, PROGRAM TERMINATED"
END
```

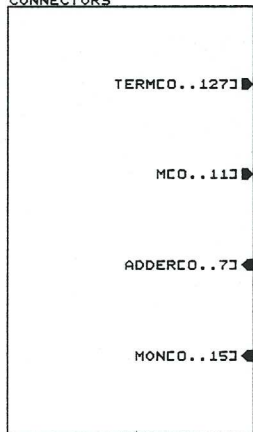
Per gli altri stadi il programma é praticamente lo stesso, cambia la definizione del vettore PESO%(N). Nella seguente tabella abbiamo riportato il suddetto vettore per ogni diverso chip di memoria del sommatore.

U1..U8	U9	U10	U11	U12
PESO%(0)=1	PESO%(0)=1	PESO%(0)=8	PESO%(0)=2	PESO%(0)=1
PESO%(1)=1	PESO%(1)=1	PESO%(1)=8	PESO%(1)=2	PESO%(1)=2
PESO%(2)=1	PESO%(2)=1	PESO%(2)=8	PESO%(2)=2	PESO%(2)=4
PESO%(3)=1	PESO%(3)=1	PESO%(3)=8	PESO%(3)=2	PESO%(3)=8
PESO%(4)=1	PESO%(4)=1	PESO%(4)=8	PESO%(4)=2	PESO%(4)=16
PESO%(5)=1	PESO%(5)=1	PESO%(5)=8	PESO%(5)=2	PESO%(5)=32
PESO%(6)=1	PESO%(6)=1	PESO%(6)=8	PESO%(6)=2	PESO%(6)=64
PESO%(7)=1	PESO%(7)=1	PESO%(7)=8	PESO%(7)=2	PESO%(7)=1
PESO%(8)=1	PESO%(8)=1*8	PESO%(8)=4	PESO%(8)=2	PESO%(8)=2
PESO%(9)=1	PESO%(9)=1*8	PESO%(9)=4	PESO%(9)=4	PESO%(9)=4
PESO%(10)=1	PESO%(10)=1*8	PESO%(10)=4	PESO%(10)=4	PESO%(10)=8
PESO%(11)=1	PESO%(11)=1*8	PESO%(11)=4	PESO%(11)=1	PESO%(11)=16
PESO%(12)=1	PESO%(12)=1*8	PESO%(12)=4	PESO%(12)=1	PESO%(12)=1
PESO%(13)=1	PESO%(13)=0	PESO%(13)=4	PESO%(13)=1	PESO%(13)=2
PESO%(14)=1	PESO%(14)=0	PESO%(14)=4	PESO%(14)=1	PESO%(14)=4

Essendo necessario un unico esemplare, questa scheda é stata prodotta utilizzando la tecnologia wire-wrap.

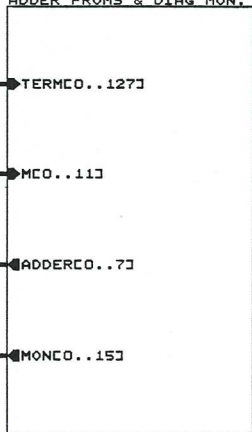


CONNECTORS



EUROCON.SCH

ADDER PROMS & DIAG MON. PORT BUFFERING

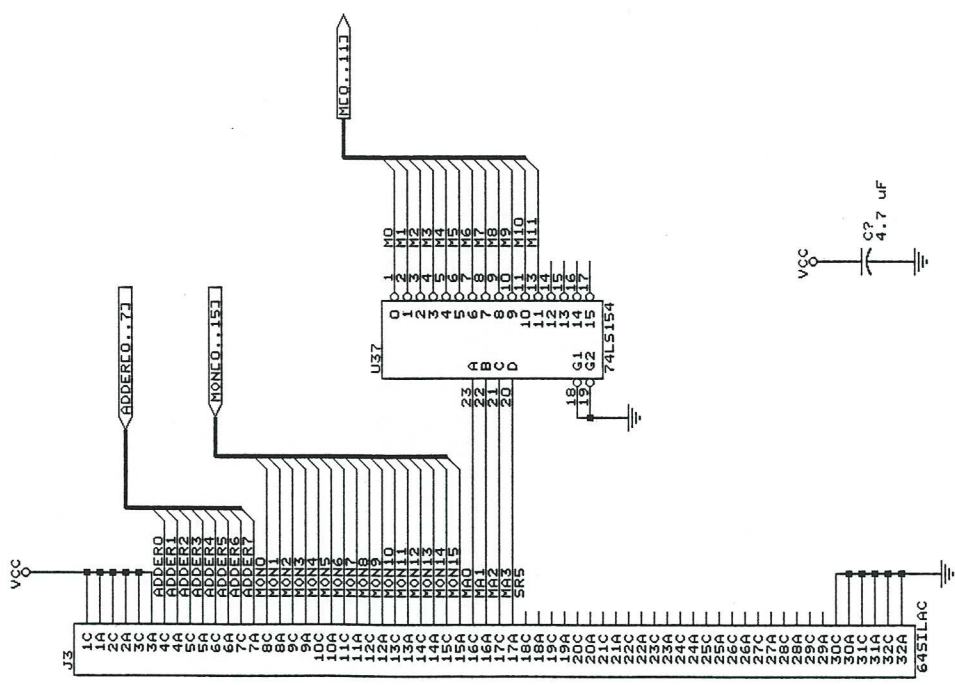
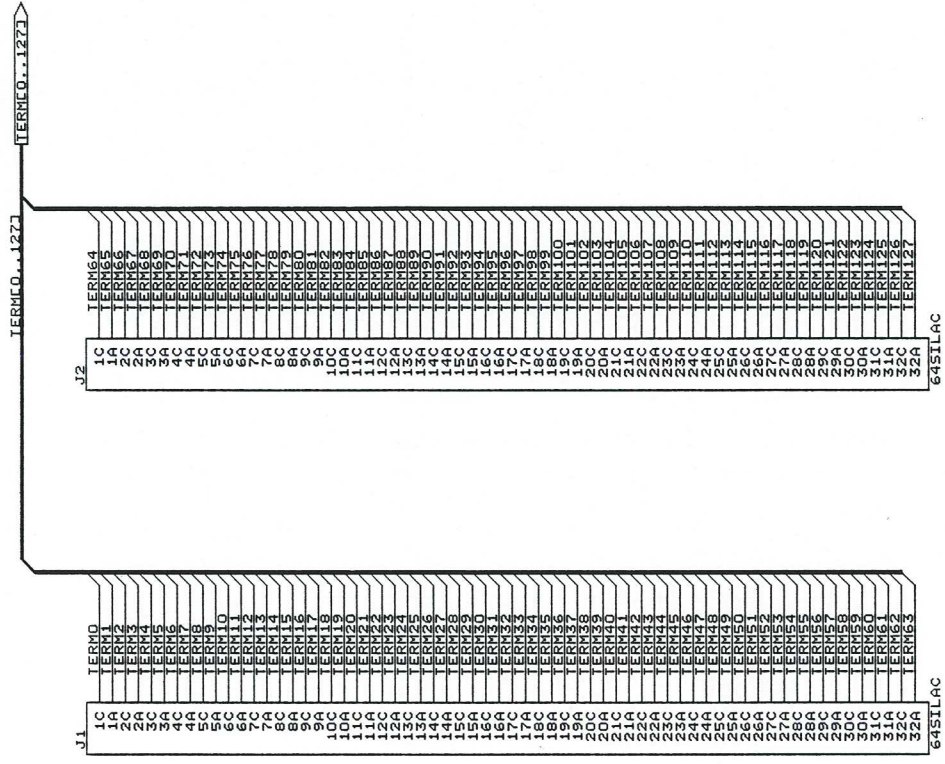


PROMS.SCH



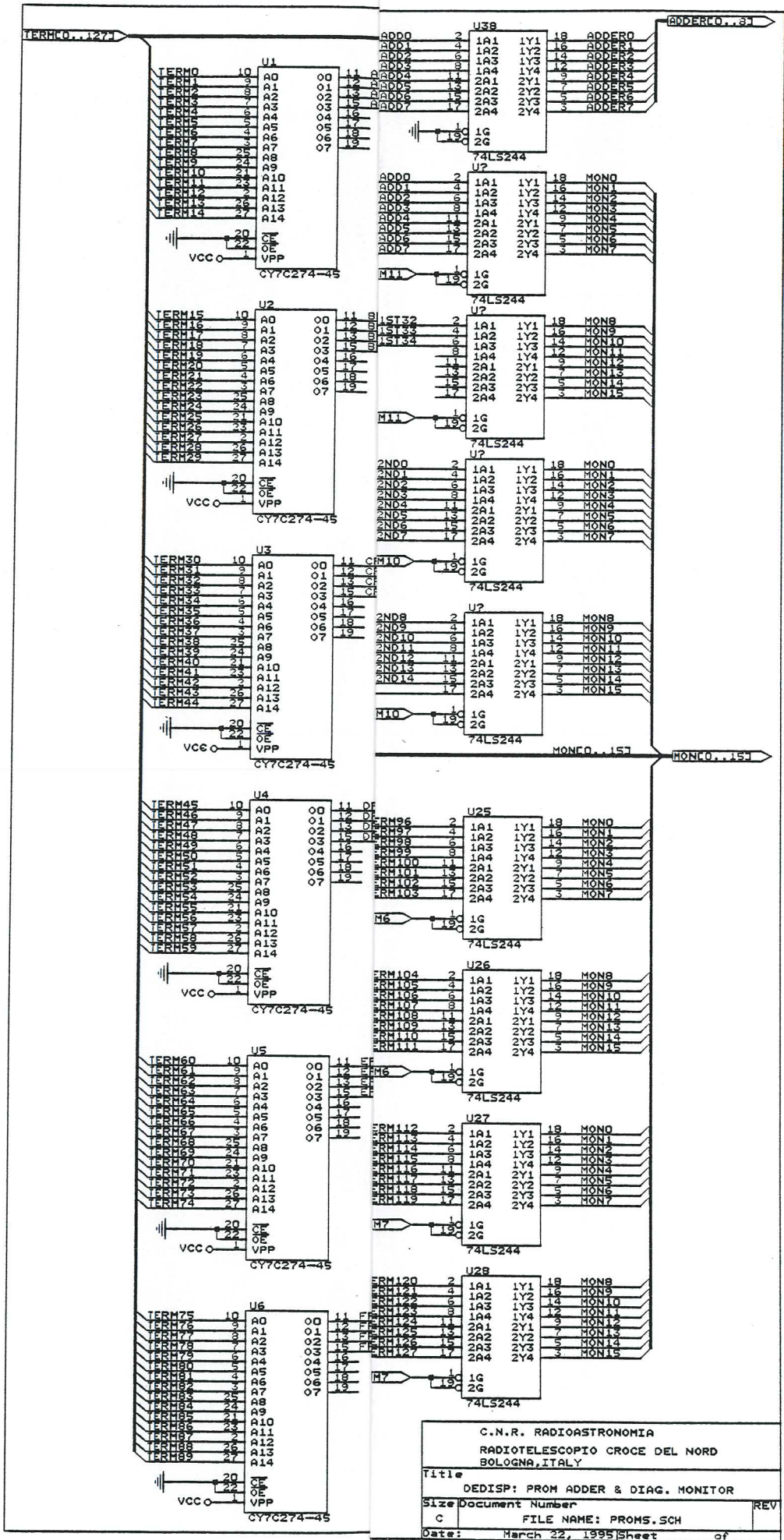
C.N.R. RADIOASTRONOMIA		
RADIOTELESCOPIO CROCE DEL NORD		
BOLOGNA, ITALY		
Title		
DEDISPERSOR: ADDER CARD MAIN BLOCK		
Size	Document Number	REV
A	FILE NAME: ADDER.SCH	
Date:	March 22, 1995	Sheet of

on every TERM there is a pull-down resistor of 1kOhm

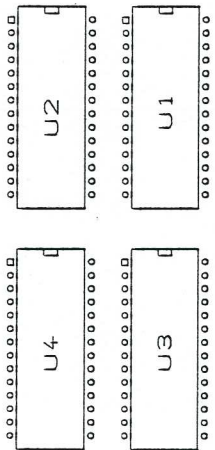
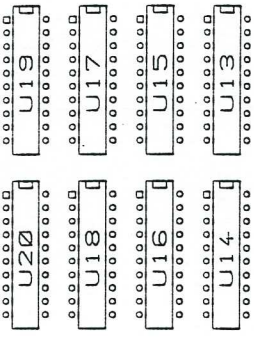
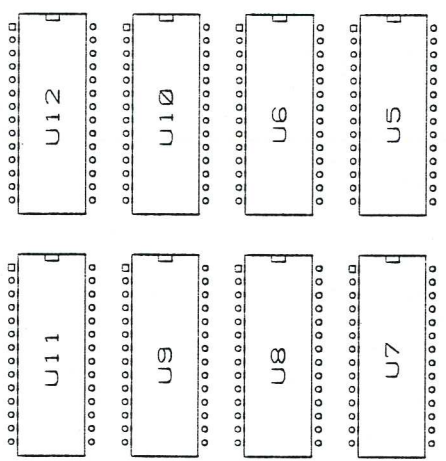
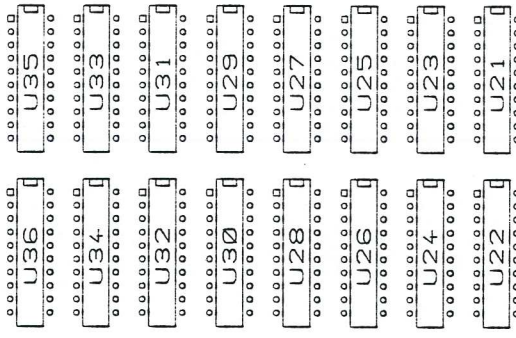
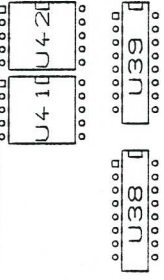


C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO CROCE DEL NORD  
 BOLOGNA, ITALY  
 A. MACCAFERRI

Title  
 DEDISP: ADDER CONN. & DIAG. MONITOR DEC.  
 Size Document Number  
 B FILE NAME: EUROCON.SCH  
 Date: March 22, 1995 Sheet of

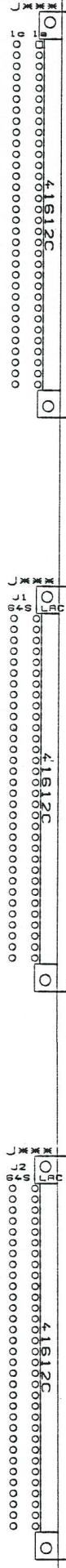


DISPLAY



DEDISPERSOR ADDER LAYOUT

DEDISPERSOR ADDER LAYOUT



## **- Controller**

Il controller ha tre funzioni principali:

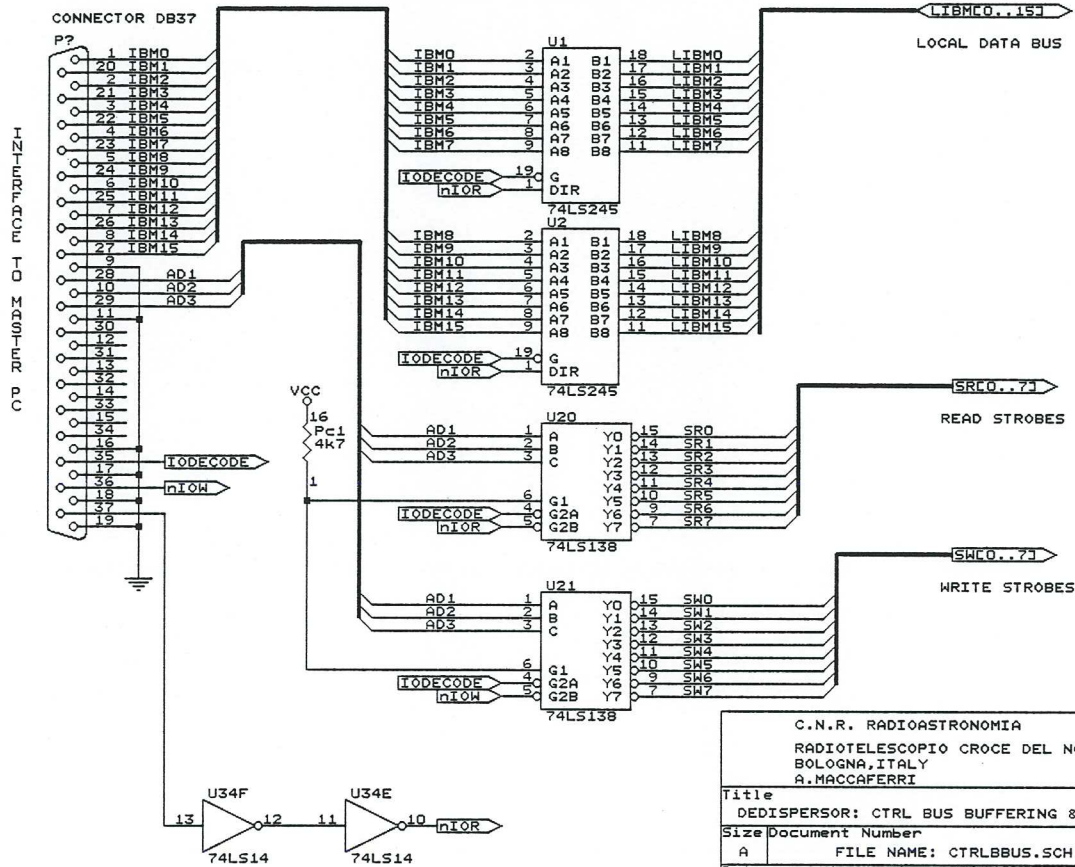
- interfacciamento verso un host (nel nostro caso il PC i386) per il controllo del dedispersore stesso e per la scrittura dei dati da dedispersere e la lettura dei dati dedispersi attraverso il link principale.
- interfacciamento verso un secondo utilizzatore che accede in sola lettura ai dati dedispersi (nella configurazione attuale la scheda multiplexer inserita sempre nel PC i386 che smista poi i dati verso le 4 schede i860).
- gestione dell'accesso al dedispersore in una configurazione multi controllore implementando la necessaria logica di arbitraggio (semaforo).

Facendo riferimento allo schema a blocchi del controller possiamo individuare quindi vari blocchi funzionali:

- Buffer e decodifica dei segnali del link con l'host e generazione di un bus locale.
- Registri interni di configurazione, controllo e logica di arbitraggio per l'accesso al dedispersore.
- Generazione dei vari segnali del backplane ed interfacciamento al bus locale:
  - Buffer del bus dati in scrittura nei buffer di memoria (MEMI[0..15]).
  - Contatore, decodifica e buffer per la generazione degli strobe di selezione buffer per la scrittura postincrementata dei dati (WE[0..7]).
  - Buffer del bus offset di dedispersione per il settaggio dei contatori di indirizzo di ciascun canale, valore di dedispersione (DEB[0..15]).
  - Contatore, decodifica e buffer per la selezione del contatore di indirizzamento ove scrivere l'offset di dedispersione (C[0..2] selezione contatore all'interno degli 8 di una scheda di memoria, SEL[0..15] selezione della scheda).
  - Memorizzazione e buffer dei segnali di abilitazione delle sottobande (CS[0..15]).
  - Buffer di lettura dati dedispersi dal sommatore (ADDER[0..7]).
  - Buffer di lettura dati parziali dal sommatore per diagnostica (MON[0..15]).
  - Memorizzazione e buffer dei segnali di selezione porta dati parziali dal sommatore (MA[0..3]).
- Buffer dei dati dedispersi dal sommatore verso l'utilizzatore secondario (Mux 4xI860) e relativa logica per la lettura postincrementata.

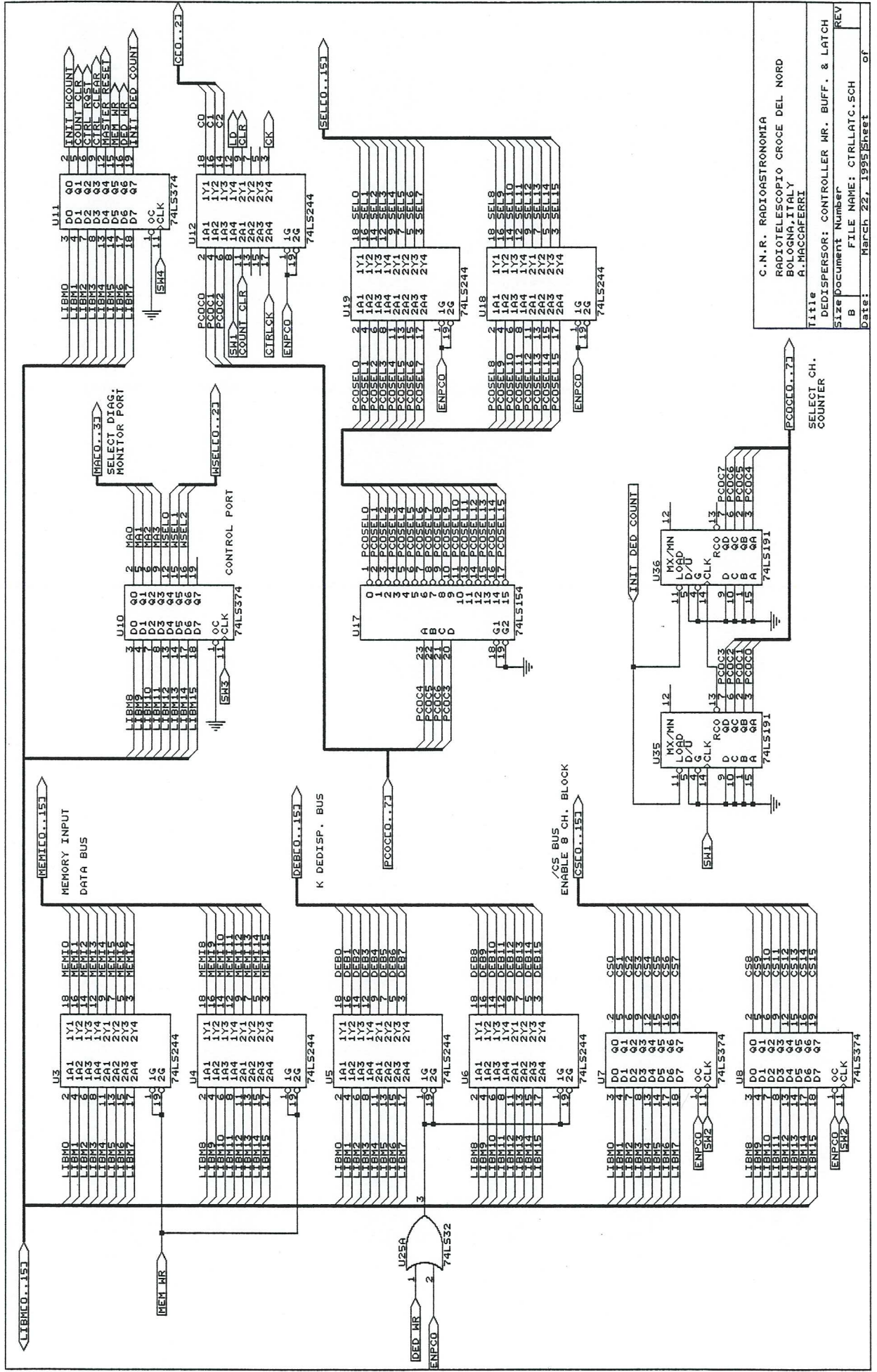
Essendo necessario un unico esemplare, anche questa scheda é stata prodotta utilizzando la tecnologia wire-wrap.

PULL-UP 820, PULL DOWN 390 on bus signal



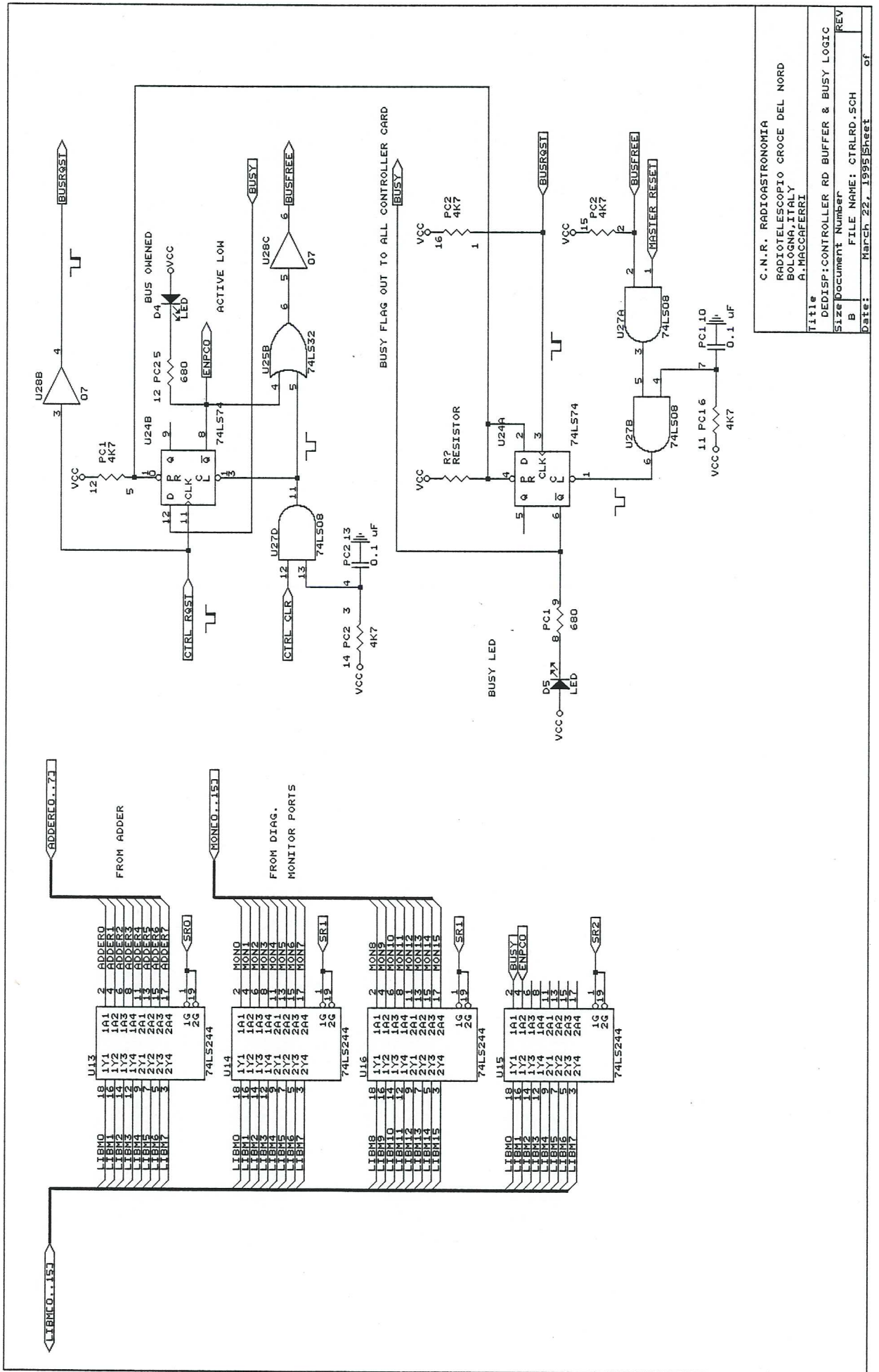
C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO CROCE DEL NORD  
 BOLOGNA, ITALY  
 A. MACCAFERRI

Title		DEDISPERSOR: CTRL BUS BUFFERING & DECODER	
Size	Document Number	REV	
A	FILE NAME: CTRLBBUS.SCH		
Date:	March 22, 1995	Sheet	of



C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO GROCE DEL NORD  
 BOLOGNA, ITALY  
 A. MACCAFERRI

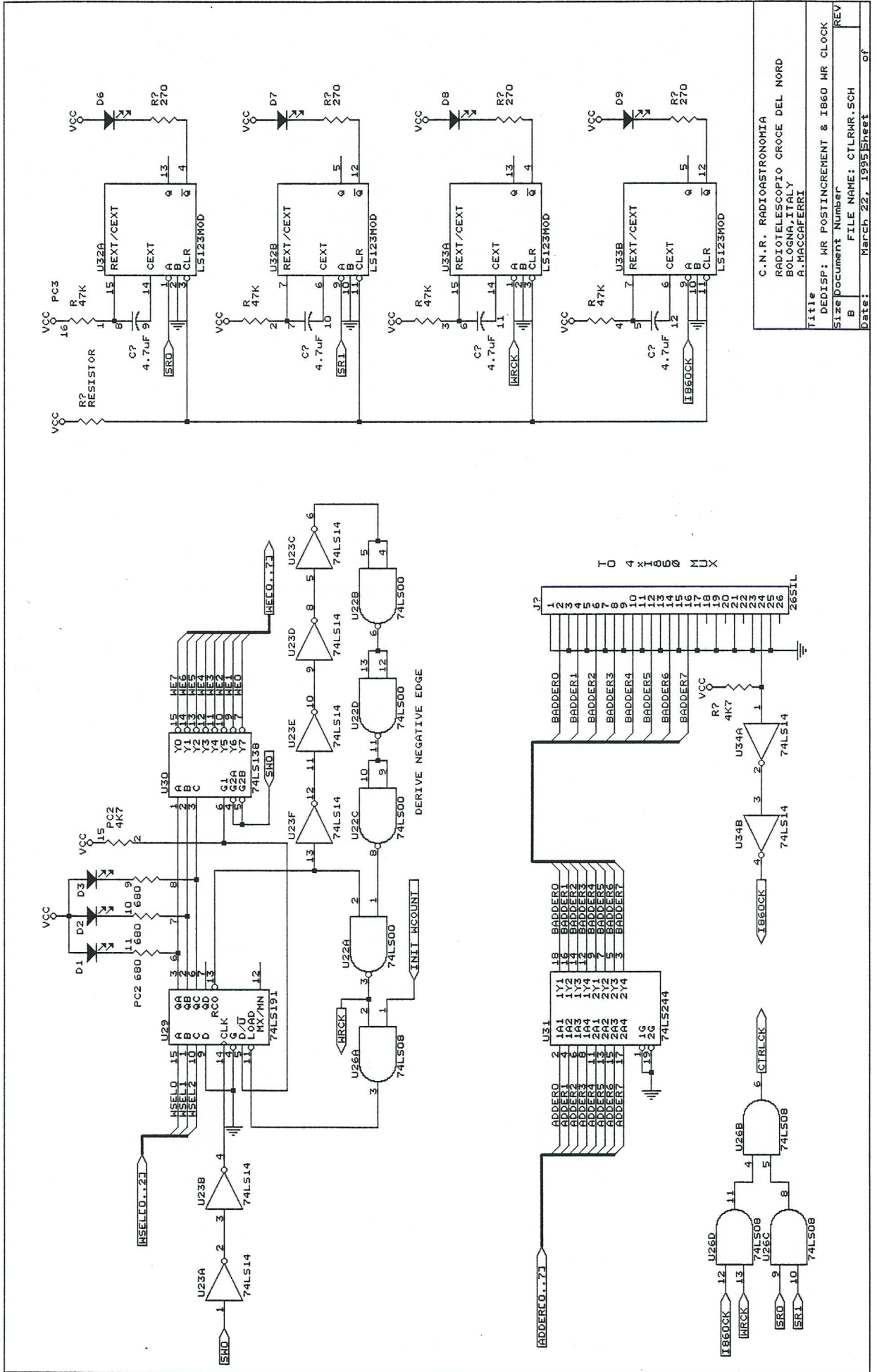
Title  
 DISPENSOR: CONTROLLER MR. BUFF. & LATCH  
 Size: Document Number: B FILE Name: CTRLLATC.SCH REV  
 Date: March 22, 1995 Sheet of



C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO GROCE DEL NORD  
 BOLOGNA, ITALY  
 A. MACCAFERRI

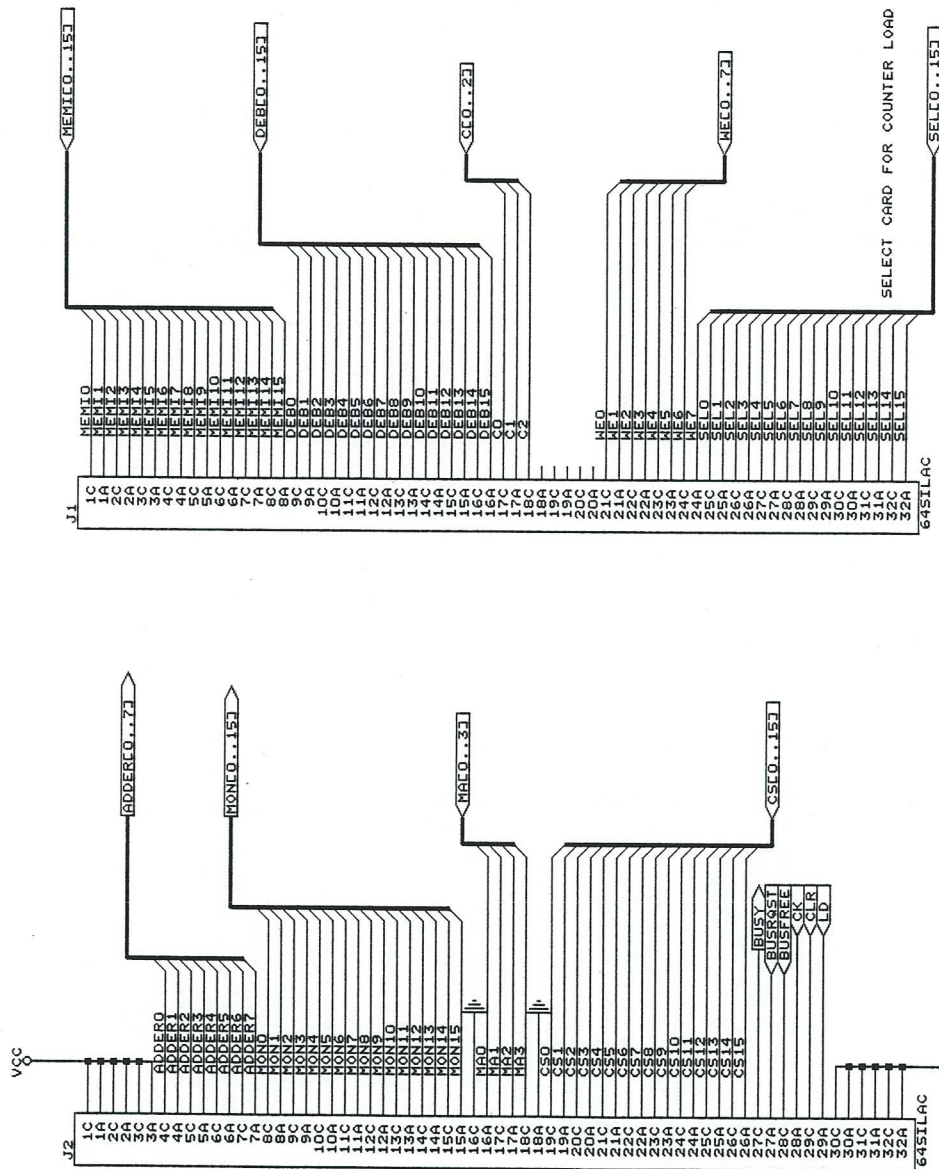
Title	DEDISP: CONTROLLER RD BUFFER & BUSY LOGIC
Size	Document Number B
REV	FILE NAME: CTRLRD.SCH
Date:	March 22, 1995
Sheet	of





C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO GROCE DEL NORD  
 BOLOGNA, ITALY  
 A. MACCAFERRI

Title: DEISP: MR POSTINCREMENT & I860 MR CLOCK  
 Size: Document Number: \_\_\_\_\_ REV \_\_\_\_\_  
 B FILE NAME: CTRLMR.SCH  
 Date: March 22, 1995 Sheet \_\_\_\_\_ of \_\_\_\_\_

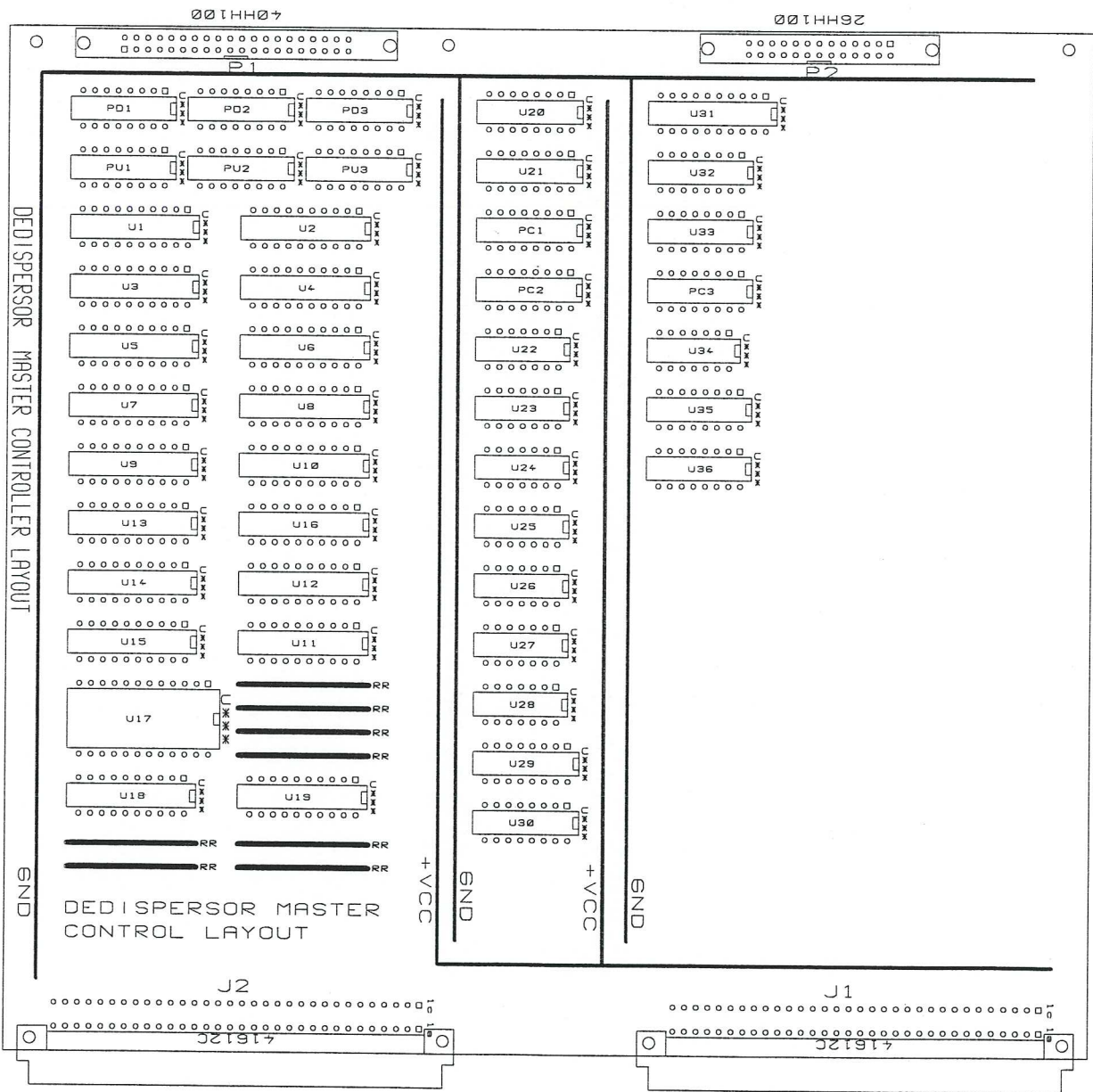


C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO GROCE DEL NORD  
 BOLOGNA, ITALY  
 A. MACCAFERRI

Title  
 DEDISP: MASTER CTRL. BACKPLANE CONNECTORS

Size Document Number  
 B FILE NAME: CTRLLEURO.SCH REV

Date: March 22, 1995 Sheet of

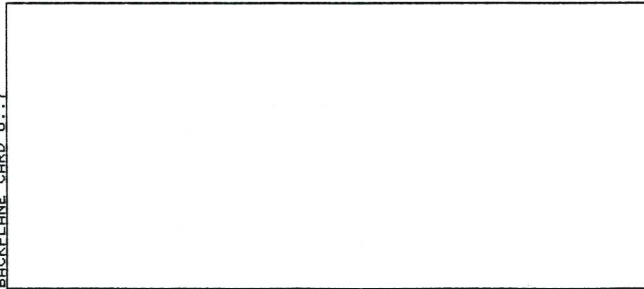


## ***- Backplane***

Il backplane posteriore ha la funzione di interconnettere le varie schede ed é costruito in tecnologia mista wire-wrap e flat cable, questa tecnologia permette di costruire agevolmente prototipi ed esemplari unici come é il caso del dedispersore in modo estremamente economico, chiaramente non consente trasferimenti dati ad alta velocità a causa del non adattamento delle impedenze, il transfer rate consentito é comunque adeguato alle richieste di questo circuito. Il collegamento in parallelo dei segnali MEMI[0..15], DEB[0..15] e C[0..2] dal controller a tutte le schede di memoria e stato effettuato tramite flat cable, mentre per il collegamento dei segnali di controllo e selezione delle singole schede sono stati fatti dei collegamenti punto-punto dal controller a ciascuna scheda in tecnologia wire-wrap. Per le uscite di ogni scheda di memoria verso il sommatore sono stati utilizzati dei collegamenti flat cable punto-punto. I connettori del controller principale e di quello opzionale, sono collegati in parallelo tramite flat cable. Per chiarezza di schematizzazione, i collegamenti fisici non sono stati indicati, le linee con lo stesso nome sono collegate assieme.

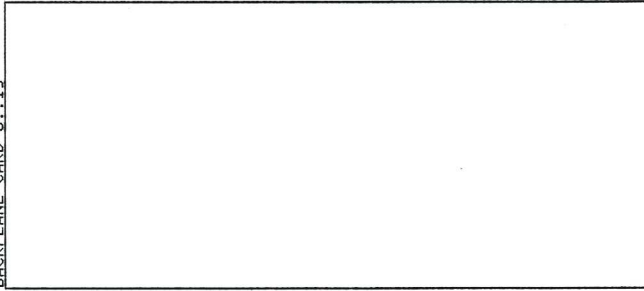
Per agevolare eventuali controlli e riparazioni, é stata riportata sia la vista frontale che quella dal retro. A causa delle dimensioni fisiche dei fogli, ogni vista é suddivisa in 3 parti A, B e C rispettivamente da sinistra a destra.

BACKPLANE\_CARD\_0..7



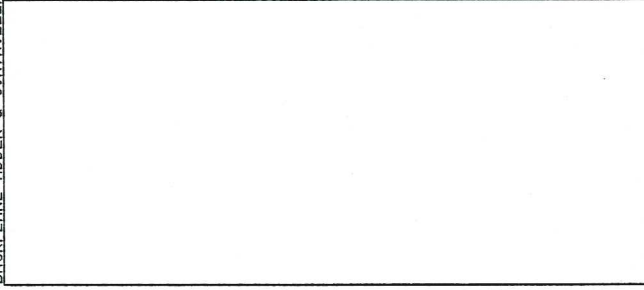
BACKPA.SCH

BACKPLANE\_CARD\_0..15



BACKPB.SCH

BACKPLANE\_ADDER & CONTROLLERS



BACKPC.SCH

C.N.R. RADIOASTRONOMIA  
RADIOtelescopio Croce del Nord  
BOLOGNA, ITALY  
A. MACCAFERRI

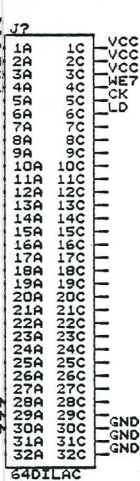
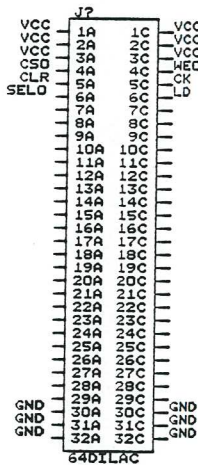
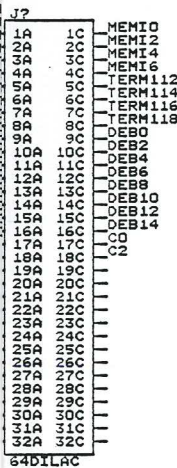
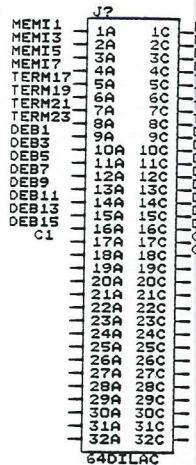
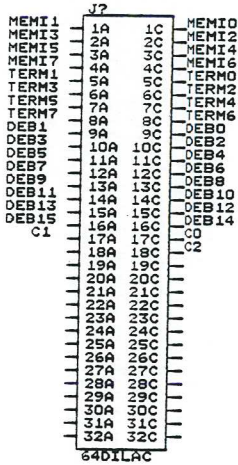
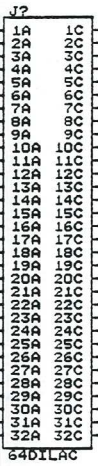
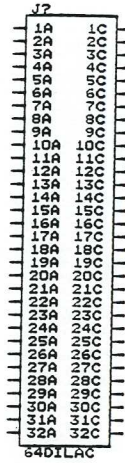
Title

DEDISP: BACKPLANE WIRING DIAGRAM FRONT VIEW

Size Document Number

B FILE NAME: BACKPLANE.SCH REV

Date: March 22, 1995 Sheet of



CARD 0  
CH. 0..7

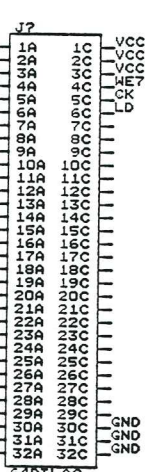
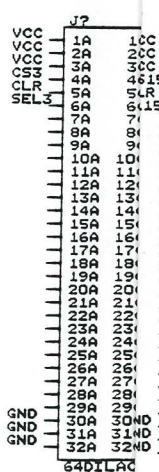
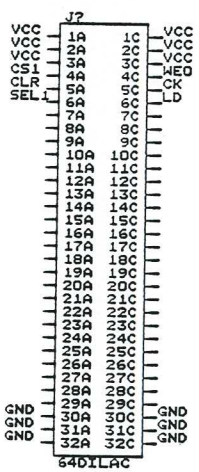
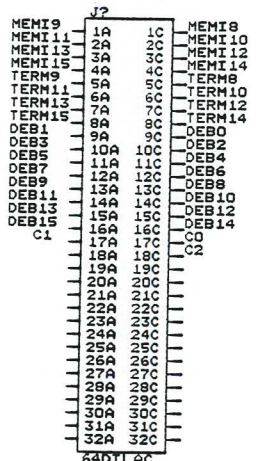
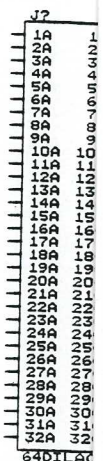
CARD 1  
CH. 16..23

CARD 7  
CH. 112..119

VIEW FROM CARD SIDE

C.N.R. RADIOASTRONOMIA  
RADIOTELESCOPIO CROCE DEL NORD  
BOLOGNA, ITALY  
A. MACCAFERRI

Title: OEDISP:BACKPLANE SEZ.A WIRING ASSEMBLY FRONT  
Size: Document Number: REV  
C FILE NAME: BACKPA.SCH  
Date: March 22, 1995 Sheet of



CARD 8  
CH. 8..15

CARD 10  
CH. 24..127

VIEW FROM CARD SIDE

C.N.R. RADIOASTRONOMIA	
RADIOTELESCOPIO CROCE DEL NORD	
BOLOGNA, ITALY	
A. MACCAFERRI	
Title	
DEDISP:BACKPLANE SEZ.8 WIRING ASSEMBLY FRONT	
Size Document Number	
C	FILE NAME: BACKPB.SCH
Date:	March 22, 1995 Sheet of

J7		
TERM1	1A	1C
TERM3	2A	2C
TERM5	3A	3C
TERM7	4A	4C
TERM9	5A	5C
TERM11	6A	6C
TERM13	7A	7C
TERM15	8A	8C
TERM17	9A	9C
TERM19	10A	10C
TERM21	11A	11C
TERM23	12A	12C
TERM25	13A	13C
TERM27	14A	14C
TERM29	15A	15C
TERM31	16A	16C
TERM33	17A	17C
TERM35	18A	18C
TERM37	19A	19C
TERM39	20A	20C
TERM41	21A	21C
TERM43	22A	22C
TERM45	23A	23C
TERM47	24A	24C
TERM49	25A	25C
TERM51	26A	26C
TERM53	27A	27C
TERM55	28A	28C
TERM57	29A	29C
TERM59	30A	30C
TERM61	31A	31C
TERM63	32A	32C

64DILAC

J7		
1A	1C	
2A	2C	
3A	3C	
4A	4C	
5A	5C	
6A	6C	
7A	7C	
8A	8C	
9A	9C	
10A	10C	
11A	11C	
12A	12C	
13A	13C	
14A	14C	
15A	15C	
16A	16C	
17A	17C	
18A	18C	
19A	19C	
20A	20C	
21A	21C	
22A	22C	
23A	23C	
24A	24C	
25A	25C	
26A	26C	
27A	27C	
28A	28C	
29A	29C	
30A	30C	
31A	31C	
32A	32C	

64DILAC

J7		
TERM65	1A	1C
TERM67	2A	2C
TERM69	3A	3C
TERM71	4A	4C
TERM73	5A	5C
TERM75	6A	6C
TERM77	7A	7C
TERM79	8A	8C
TERM81	9A	9C
TERM83	10A	10C
TERM85	11A	11C
TERM87	12A	12C
TERM89	13A	13C
TERM91	14A	14C
TERM93	15A	15C
TERM95	16A	16C
TERM97	17A	17C
TERM99	18A	18C
TERM101	19A	19C
TERM103	20A	20C
TERM105	21A	21C
TERM107	22A	22C
TERM109	23A	23C
TERM111	24A	24C
TERM113	25A	25C
TERM115	26A	26C
TERM117	27A	27C
TERM119	28A	28C
TERM121	29A	29C
TERM123	30A	30C
TERM125	31A	31C
TERM127	32A	32C

64DILAC

J7		
MEMI1	1A	1C
MEMI3	2A	2C
MEMI5	3A	3C
MEMI7	4A	4C
MEMI9	5A	5C
MEMI11	6A	6C
MEMI13	7A	7C
MEMI15	8A	8C
DEB1	9A	9C
DEB3	10A	10C
DEB5	11A	11C
DEB7	12A	12C
DEB9	13A	13C
DEB11	14A	14C
DEB13	15A	15C
DEB15	16A	16C
C1	17A	17C
	18A	18C
	19A	19C
	20A	20C
WE1	21A	21C
WE3	22A	22C
WE5	23A	23C
WE7	24A	24C
SEL1	25A	25C
SEL3	26A	26C
SEL5	27A	27C
SEL7	28A	28C
SEL9	29A	29C
SEL11	30A	30C
SEL13	31A	31C
SEL15	32A	32C

64DILAC

J7		
VCC	1A	1C
VCC	2A	2C
VCC	3A	3C
ADDER1	4A	4C
ADDER3	5A	5C
ADDER5	6A	6C
ADDER7	7A	7C
MON1	8A	8C
MON3	9A	9C
MON5	10A	10C
MON7	11A	11C
MON9	12A	12C
MON11	13A	13C
MON13	14A	14C
MON15	15A	15C
MA0	16A	16C
MA2	17A	17C
GND	18A	18C
	19A	19C
	20A	20C
	21A	21C
	22A	22C
	23A	23C
	24A	24C
	25A	25C
	26A	26C
	27A	27C
	28A	28C
	29A	29C
GND	30A	30C
GND	31A	31C
GND	32A	32C

64DILAC

J7		
VCC	1A	1C
VCC	2A	2C
VCC	3A	3C
ADDER1	4A	4C
ADDER3	5A	5C
ADDER5	6A	6C
ADDER7	7A	7C
MON1	8A	8C
MON3	9A	9C
MON5	10A	10C
MON7	11A	11C
MON9	12A	12C
MON11	13A	13C
MON13	14A	14C
MON15	15A	15C
MA0	16A	16C
MA2	17A	17C
GND	18A	18C
CS1	19A	19C
CS3	20A	20C
CS5	21A	21C
CS7	22A	22C
CS9	23A	23C
CS11	24A	24C
CS13	25A	25C
CS15	26A	26C
BUSRQST	27A	27C
PLCK	28A	28C
LD	29A	29C
GND	30A	30C
GND	31A	31C
GND	32A	32C

64DILAC

ADDER

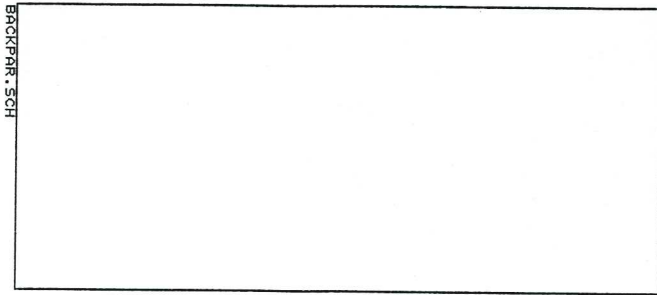
PC MASTER  
CONTROLLER  
INTERFACE

VIEW FROM CARD SIDE

C.N.R. RADIOASTRONOMIA		
RADIOTELESCOPIO CROCE DEL NORD		
BOLOGNA, ITALY		
A. MACCAFERRI		
Title	DEDISP:BACKPLANE SEZ.C WIRING ASSEMBLY FRONT	
Size	Document Number	REV
C	BACKPC.SCH	
Date:	March 22, 1995	Sheet of

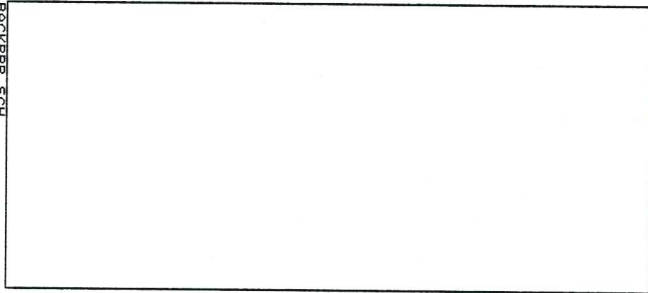


BACKPLANE\_CARD\_0..7



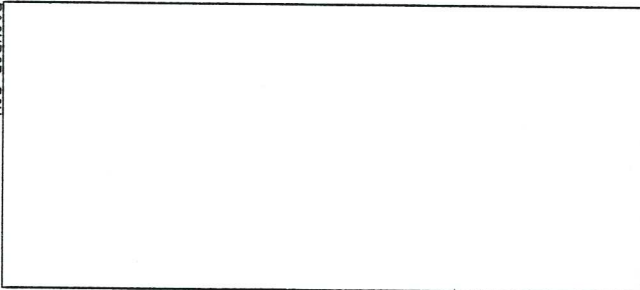
BACKPAR.SCH

BACKPLANE\_CARD\_8..15



BACKPBR.SCH

BACKPLANE\_ADDER & CONTROLLERS



BACKFOR.SCH

C.N.R. RADIOASTRONOMIA

RADIOTELESCOPIO GROCE DEL NORD

BOLZONA, ITALY

A. MASCHERERI

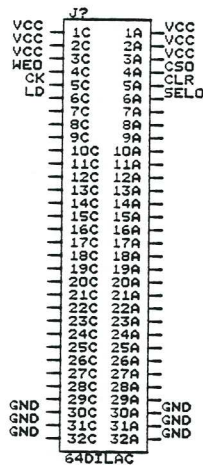
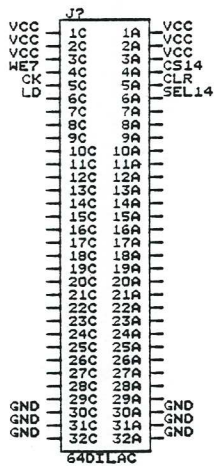
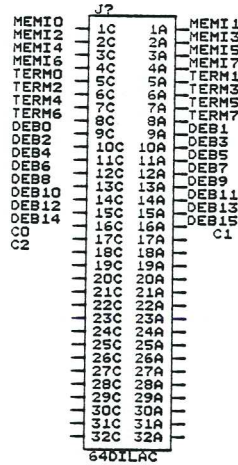
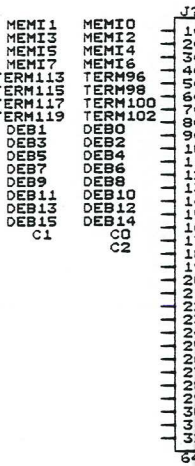
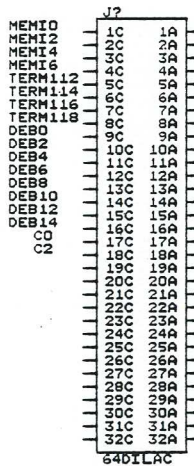
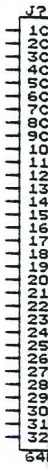
TRAV VLV

DEDISP:BACKPLANE WIRING DIAGRAM BACK

Size Document Number REV

B FILE NAME: BACKMIRE.SCH

Date: March 22, 1995 Sheet of



CARD 7

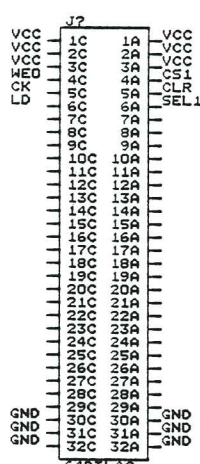
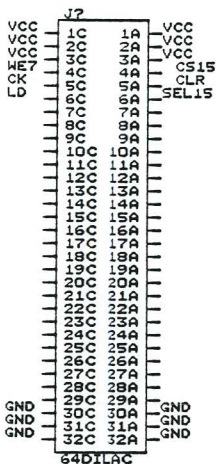
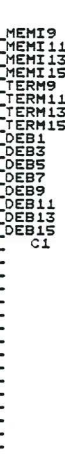
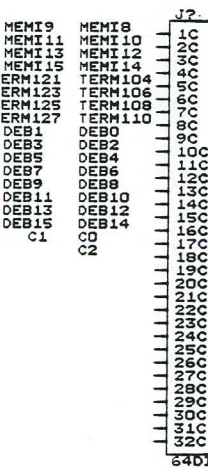
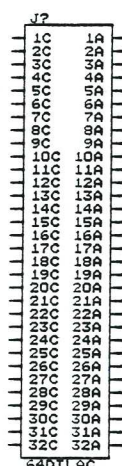
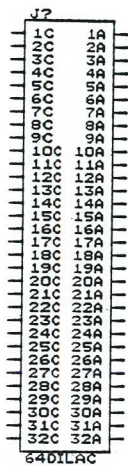
CH. 112..119

CARD 0

CH. 0..7

VIEW FROM REAR SIDE

C.N.R. RADIOASTRONOMIA		
RADIOTELESCOPIO CROCE DEL NORD		
BOLOGNA, ITALY		
A. MACCAFERRI		
Title		
DEDISP: BACKPLANE SEZ. A REAR VIEW		
Size Document Number		
C	FILE NAME BACKPAR.SCH	REV
Date: March 22, 1995 Sheet of		



CARD 15

CARD 15

CARD 8

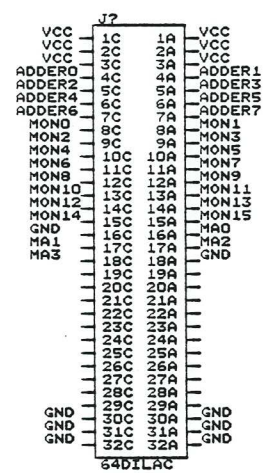
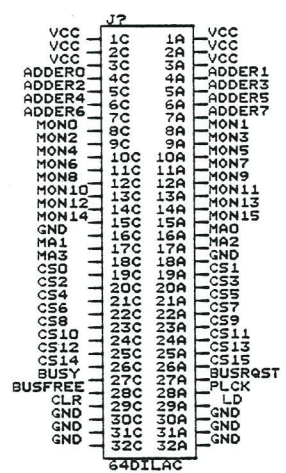
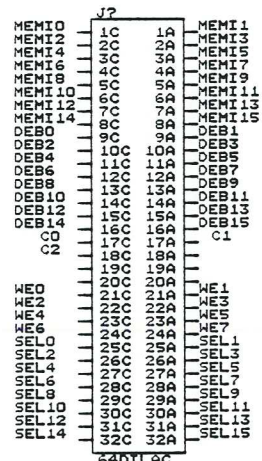
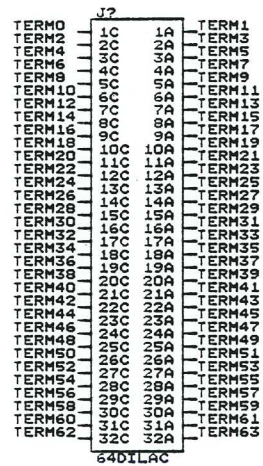
CH. 120..127

CH. 120..127

CH. 8..15

VIEW FROM REAR SIDE

C.N.R. RADIOASTRONOMIA		
RADIOTELESCOPIO CROCE DEL NORD		
BOLOGNA, ITALY		
A. MACCAFERRI		
Title	DEDISP: BACKPLANE SEZ. B REAR VIEW	
Size	Document Number	REV
C	FILE NAME BACKPBR.SCH	
Date:	March 22, 1995	Sheet of



PC MASTER  
CONTROLLER  
INTERFACE

ADDER

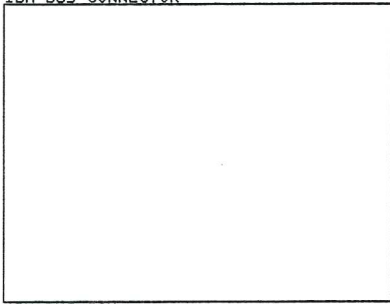
VIEW FROM REAR SIDE

C.N.R. RADIOASTRONOMIA	
RADIOTELESCOPIO GROCE DEL NORD	
BOLOGNA, ITALY	
A. MACCAFERRI	
Title	
DEDISP: BACKPLANE SEZ.C REAR VIEW	
Size	Document Number
C	FILE NAME BACKPCR.SCH
Date:	March 22, 1995 Sheet of

## ***- La scheda Dedisp-Interface***

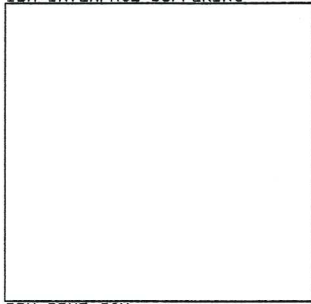
Questa scheda posta nel bus ISA del Personal Computer i386 host, collegandosi con la scheda Controller del dedispersore tramite un flat cable, permette ad un programma che opera sul PC di controllare tutto il funzionamento del dedispersore. In pratica i registri del dedispersore vengono visti come se fossero porte di I/O collegate direttamente al bus del PC stesso. A questo scopo il bus dati del PC viene bufferato (IBM[0..15]) e trasmesso sul cavo, assieme ad alcune linee di controllo (AD[1..3], nIOW e nIOR) generate dall'apposita logica di decodifica degli indirizzi di I/O del PC, che permettono di indirizzare i vari registri su cui operare e discriminare una operazione di scrittura da una di lettura.

IBM BUS CONNECTOR



IBM\_BUS.SCH

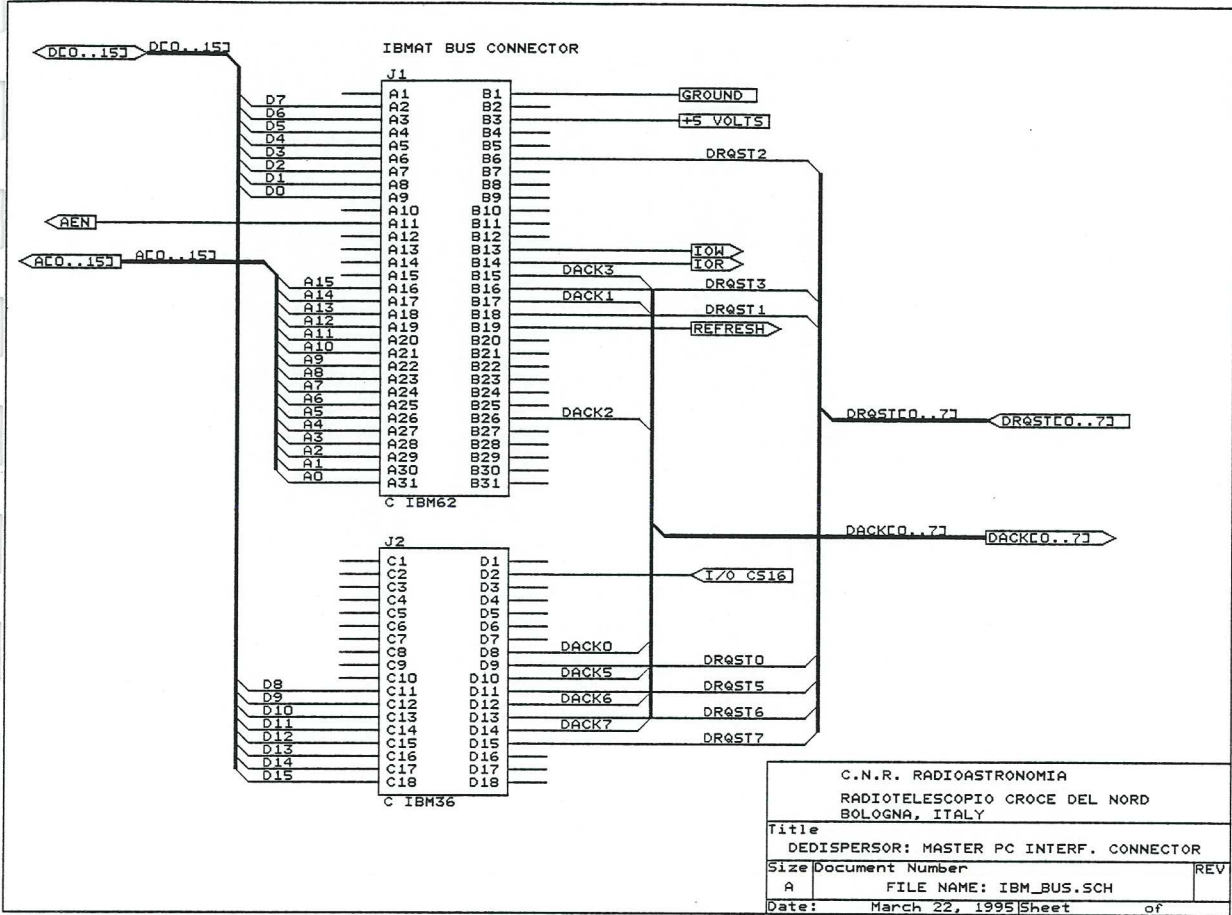
IBM INTERFACE BUFFERING



IBM\_BBUF.SCH

C.N.R. RADIOASTRONOMIA  
RADIOTELESCOPIO CROCE DEL NORD  
BOLOGNA, ITALY

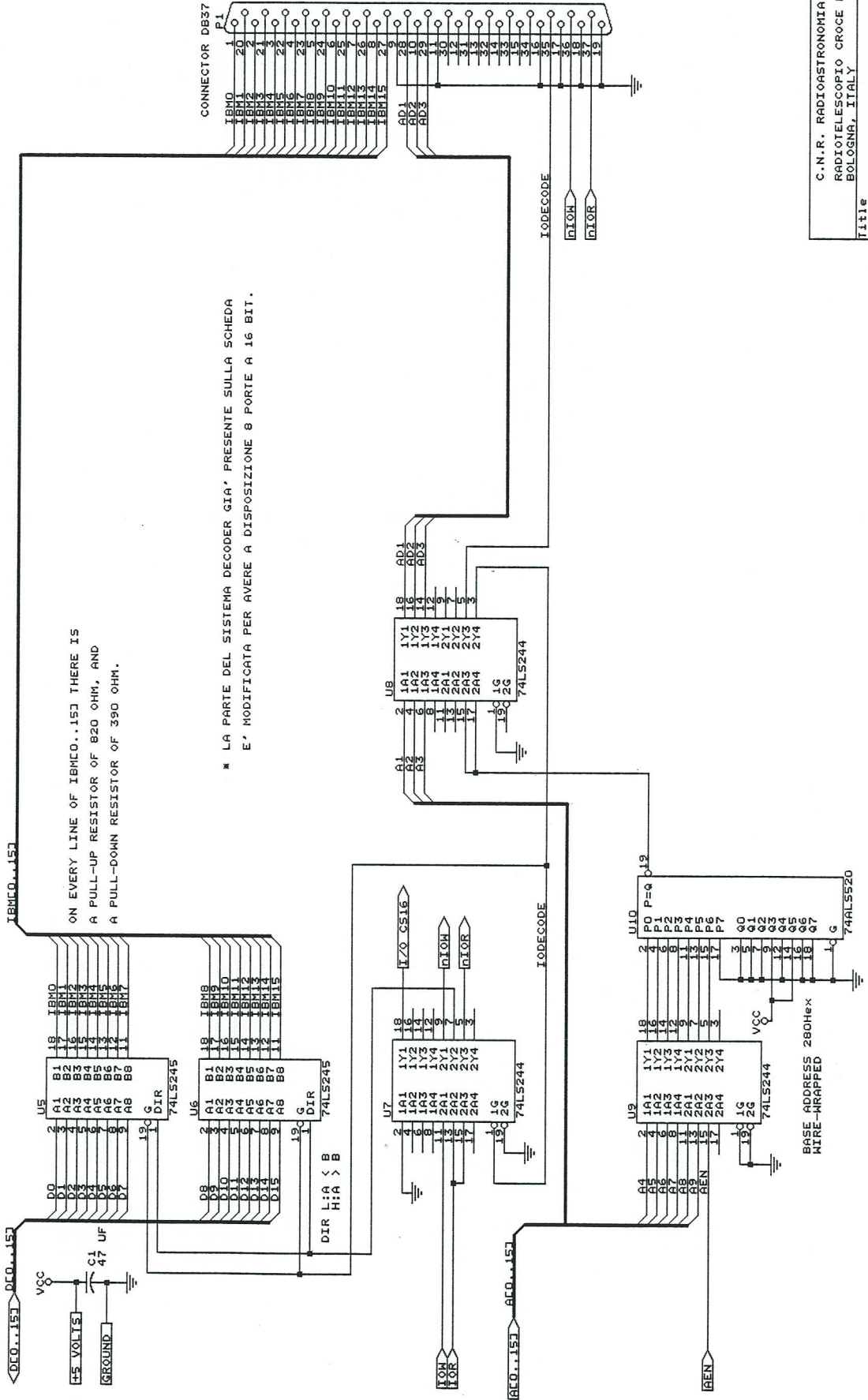
Title		
DEDISPERSOR: MASTER PC INTERFACE MAIN BLOCK		
Size	Document Number	REV
A	FILE NAME: PCD.SCH	
Date: March 22, 1995		Sheet of



C.N.R. RADIOASTRONOMIA  
 RADIOTELESCOPIO CROCE DEL NORD  
 BOLOGNA, ITALY

Title		REV
DEDISPERSOR: MASTER PC INTERF. CONNECTOR		
Size	Document Number	REV
A	FILE NAME: IBM_BUS.SCH	
Date:	March 22, 1995	Sheet of

# DEHSPERSOR CONTROLLER LINK



C.N.R. RADIOASTRONOMIA RADIOTELESCOPIO CROCE DEL NORD BOLOGNA, ITALY	
Title	DEHSPERSOR: MASTER PC INTERF. BUFFER LOGIC
Size	B
Document Number	FILE NAME: IBM_BBUF.SCH
Date:	March 22, 1985
Sheet	of



## Software di test

Per effettuare il test del dedispersore é stato scritto un programma in FORTRAN che viene eseguito sul PC host (i386), é stato incluso in questa nota tecnica come esempio per la configurazione e l'utilizzo del dedispersore. Il programma può essere suddiviso in alcune sezioni principali:

- 1) Inizializzazione del dedispersore.
- 2) Preparazione di un vettore di 8000 dati casuali con cui riempire interamente il dedispersore.
- 3) Riempimento del dedispersore.
- 4) Calcolo di 128 valori casuali di offset di dedispersione.
- 5) Configurazione del dedispersore con gli offset ottenuti.
- 6) Calcolo del vettore dedisperso di riferimento in funzione dei dati scritti e dell'offset.
- 7) Confronto fra i dati in uscita dal dedispersore ed il vettore di riferimento calcolato e visualizzazione degli eventuali errori riscontrati.
- 8) Ripetizione dal punto 2) se non vi sono errori.

Sono state inoltre riportate alcune routine scritte in assembler richiamabili da FORTRAN, utilizzate per la lettura e la scrittura sia "single word" che "block mode" verso le porte di I/O.

Program pctst

```
c      programma di test del dedispersore digitale per PULSAR
c
c      scrittura e lettura tramite ioblock (256 word)by test adder
c      test con varie dedispersioni per varie schede con calcolo veloce.
c      scrittura di valori random per riempire il buffer di memoria
```

```
$notruncate
```

```
$declare
```

```
implicit none
call init
end
```

```
Subroutine init
```

```
$notruncate
```

```
$declare
```

```
implicit none
character*32 stringi,stringj
integer*4 val4,lenght,control_a,longloop
integer*2 value,address,card,monitor_ch,offset,cardon,addertot
integer*2 maxoffset,reference,adder,nwords,ref
integer*2 intero,resto,offset_exc(0:7,0:15),card_exc(0:15)
integer*4 i,j,jj,k,nloop,mask,longtest,ihr,imin,isec,i100th
integer*4 vect(0:15)
integer*2 refbuffer(0:32767),buffer(0:32767),adderverc(0:16383)
real*4 r,ranval
integer*2 sh(128),sh2(0:127)
dimension offset(0:7,0:15)
dimension cardon(0:15)
equivalence (val4,value)
equivalence (sh(1),offset(0,0))
```

```
data vect/1,2,4,8,16,32,64,128,256,512,1024,2048,4096,8192,
&16384,32768/
```

```
data card_exc/0,2,4,6,8,10,12,14,1,3,5,7,9,11,13,15/
```

```
c      master reset
val4=239
address=648
call out$W (value,address)
val4=255
address=648
call out$w (value,address)
```

```
c      master control request
val4=251
address=648
call out$W (value,address)
val4=255
address=648
call out$w (value,address)
```

```
c      seleziona numero di canali in gruppi da 16 canali
write(*,*)' enter first card present from left (0-7) '
write(*,*)' if not all card are present, you must write '
write(*,*)' from card 7 to the card you want to test '
read(*,*)card
```

```

card=7-card
control_a=control_a+(4096*card)
val4=control_a
address=646
call out$W (value,address)

c      (init wcount) carica il valore di canali nel counter
val4=254
address=648
call out$W (value,address)
val4=255
address=648
call out$w (value,address)

c      (count clr) faccio il clear dei counter delle memorie
val4=253
address=648
call out$w (value,address)
val4=255
call out$w (value,address)

c      azzero vettore schede sotto test
do card=0,15
cardon(card)=0
enddo

99     continue
write(*,*)'select memory card to check (0-15,>15=all,-1=end) '
read(*,*)card
if (card.gt.15)then
do card=0,15
    cardon(card)=1
enddo
else
if (card.ne.-1)then
    cardon(card)=1
    goto 99
endif
endif

c      scrivo nelle memorie il valore dell'address
write(*,*)'enter memory lenght in decimal (0-1048575) '
read(*,*)lenght
4      format(a)

c      start test and write new data in memory every loop
c      if you would like to test only new dedispersion
c      with the same data comment next line (51) and uncomment
c      (510) befor dedispersion load block.
51     do longloop=0,10000
c      (init wcount) carica il valore di canali nel counter
c      per il load autoincrementato
val4=254
address=648
call out$W (value,address)
val4=255
address=648
call out$w (value,address)

```

```

c    abilito bus dati mem input e dedispersor
    val4=31
    address=648
    call out$w (value,address)
    val4=159
    call out$w (value,address)

c    (count clr) faccio il clear dei counter delle memorie
c    preparo il vettore
    do j=0,127
sh2(j)=0
    enddo
c    scrivo fisicamente i 128 valori nei counter dedisp.
    value=128
    address=642
    call outp$b(sh2,value,address)

write(*,*)'i am writing data into the memory'
write(*,*)' '
call gettim (ihr,imin,isec,i100th)
call seed (i100th+isec+imin+ihr)
i=0
do j=0,999
3    format('+',1x,i8)
    r=j/100
    if ((j-r*100).eq.0)then
        write(*,3)j
    endif
    do k=0,7
c        scrivo in ogni scheda valori casuali
        call random (ranval)
        reference=int2(ranval*32767)
        refbuffer(i)=reference
c        refbuffer(i)=i+i*256
        i=i+1
    enddo
enddo
do j=1,2
    do k=0,7999
        if (((j*8000)+k).gt.16383)exit
        refbuffer((j*8000)+k)=refbuffer(k)
    enddo
enddo

c
c    visualizza primi 15 valori buffer di test
c    do j=0,15
c        i=refbuffer(j)
c        call convbin(i,stringi)
c        write(*,*)' buffer ',j,refbuffer(j)
c        write(*,2) stringi
c    enddo
c    pause
c    write(*,*)' writing blocks..'
    do j=0,(lenght/1000)
        address=640
        if ((j+1)*1000.gt.lenght)then
            nwords=(lenght-(j*8000))
            write(*,*)' end writing ',j,nwords

```

```

        else
            nwords=8000
        endif
        call outp$b (refbuffer,nwords,address)
        write(*,3)j,nwords
    enddo
c        write(*,*)k,j
c        pause

510 do longtest=0,100
c    (count clr) faccio il clear dei counter delle memorie
c    preparo il vettore
    do j=0,127
sh2(j)=0
    enddo
c    scrivo fisicamente i 128 valori nei counter dedisp.
    value=128
    address=642
    call outp$b(sh2,value,address)
c    rileggo le memorie per fare il check
c    testando anche l'offset di dedispersione.
    do j=0,7
        do k=0,15
            offset(j,k)=0
        enddo
    enddo

c    write dedispersion offset for 16 board, 8 channel/board
    do card=0,15
        call gettim (ihr,imin,isec,i100th)
        call seed (i100th+isec+imin+ihr)
        do j=0,7
            jj=(card*8)+j
            call random (ranval)
            offset(j,card)=int2(ranval*32767)
            sh2(jj)=offset(j,card)
            write(*,*) j,card,offset(j,card)
        enddo
    enddo

c    scrivo fisicamente i 128 valori nei counter dedisp.
    value=128
    address=642
    call outp$b(sh2,value,address)
    write(*,*) ' cards dedispersion writed ',cards
    if (card.gt.7)then
        monitor_ch=card-8
        mask=65280
    else
        monitor_ch=card
        mask=255
    endif
    val4=control_a+(monitor_ch*256)
    address=646
    call out$w(value,address)

c    calcolo il valore massimo di offset
    maxoffset=0

```

```

do k=0,7
  do j=0,15
    if (offset(k,j).gt.maxoffset)then
      maxoffset=offset(k,j)
    endif
  enddo
enddo

c   calcolo il vettore offset scambiato 0.8.1.9 etc...
do card=0,15
  do k=0,7
    offset_exc(k,card)=offset(k,card_exc(card))
  enddo
enddo

do j=0,15
  write(*,*)'cardon ',j,cardon(j)
enddo
write(*,*)' max offset = ',maxoffset

write(*,*)' computing values for compare....'
c   calcolo il vettore di 16384 valori per fare il check
do nloop=0,999
  addertot=0
  do card=0,15
    if (cardon(card).eq.1)then
      ref=0
      if (card.le.7)then
c       calcola il byte teoricamente in uscita dalla scheda
        do i=0,7
c         write(*,*)' buffer ',refbuffer(nloop+offset(i,card))
          intero=offset_exc(i,card)/1000
          resto=offset_exc(i,card)-intero*1000
c         write(*,*)' card,nloop,resto',card,nloop,resto
c         write(*,*)' refbuffer',refbuffer((nloop+resto)*8)
c         write(*,*)' vect(i)',vect(i)
          ref=ref+iand(refbuffer(((nloop+resto)*8)+card),vect(i))
        enddo
        adder=0
        do i=0,7
          if (btest(ref,i))then
            adder=adder+1
          endif
        enddo
c         write(*,*)' adder,card,ref,nloop',adder,card,ref,nloop
c         write(*,*)' adder,card ',adder,card
          addertot=addertot+adder
        else
c       calcola il byte teoricamente in uscita dalla scheda
        do i=0,7
          intero=offset_exc(i,card)/1000
          resto=offset_exc(i,card)-intero*1000
c         write(*,*)' card,nloop,resto',card,nloop,resto
c         write(*,*)' refbuffer',refbuffer(((nloop+resto)*8)+card-8)
c         write(*,*)' vect(i+8)',vect(i+8)
          ref=ref+iand(refbuffer(((nloop+resto)*8)+card-8),vect(i+8))
        enddo
        adder=0
        do i=8,15

```

```

        if (btest(ref,i)) then
        adder=adderr+1
        endif
        enddo
c        write(*,*) 'adder, card, ref, nloop', adder, card, ref, nloop
        addertot=addertot+adder
        endif
        endif
        enddo
c        write(*,*) ' addertot ', nloop, addertot
        addervec(nloop)=addertot
        enddo
do j=1,16
  do k=0,999
    if (((j*1000)+k).gt.16383) exit
    addervec((j*1000)+k)=addervec(k)
  enddo
enddo
do j=0,20
write(*,*) ' addervec ', j, addervec(j)
enddo

c        confronto il vettore di test con quello letto
do k=0, (length-maxoffset-1-8000), 8000
write(*,*) ' block, longest, longloop ', k, longest, longloop
address=640
nwords=8000
call inpsb(buffer, nwords, address)
do nloop=0, 7999
  if ((k+nloop).gt.length) exit
  value=buffer(nloop)
  if (iand(value, 255).eq.addervec(nloop)) then
c        test ok
        else
          j=iand(value, 255)
          i=ref
          write(*,*) 'k, nloop, ref, addtot, val rd mask, value read'
          write(*,*) 'error at ', k, nloop, i, addertot, j, value
          write(*,1) k
          write(*,1) i, j
1          format (1x, z6)
          call convbin(i, stringi)
          call convbin(j, stringj)
          write(*,*) '10987654321098765432109876543210'
          write(*,2) stringi, stringj
2          format (1x, a, /, 1x, a)
          pause
        endif
      enddo
    enddo
    write(*,*) 'read back ended'
    write(*,*) ' test loop n...', longest, longloop
  enddo
c  fine loop longest (new offset dedisp)
enddo
c  fine loop longloop (new data in buffer)
return
end

```

```
C*****  
C* conversione decimale binario e visualizzazione *  
C*****
```

```
subroutine convbin(val4,string)
```

```
implicit none
```

```
$notruncate
```

```
$declare
```

```
character*(*) string
```

```
integer*4 val4
```

```
integer*4 i,k
```

```
do i=31,0,-1
```

```
  k=31-i+1
```

```
  if (btest(val4,i)) then
```

```
    string(k:k)='1'
```

```
  else
```

```
    string(k:k)='0'
```

```
  endif
```

```
enddo
```

```
return
```

```
end
```



## Listato di alcune routine assembler richiamate da fortran

```
TITLE ncasm.asm
NAME ncasm
.286C
NCASM_TEXT SEGMENT BYTE PUBLIC 'CODE'
NCASM_TEXT ENDS
_DATA SEGMENT WORD PUBLIC 'DATA'
_DATA ENDS
CONST SEGMENT WORD PUBLIC 'CONST'
CONST ENDS
_BSS SEGMENT WORD PUBLIC 'BSS'
_BSS ENDS
DGROUP GROUP CONST, _BSS, _DATA
ASSUME CS: NCASM_TEXT, DS: DGROUP, SS: DGROUP, ES: DGROUP
EXTRN __acrtused:ABS
NCASM_TEXT SEGMENT
;
;=====
; Subroutine OUT$W(word,address)
;=====
PUBLIC OUT$W
OUT$W PROC FAR
push bp
mov bp,sp

push es

les bx,DWORD PTR [bp+10] ;word
mov ax,es:[bx]

les bx,DWORD PTR [bp+6] ;address
mov dx,es:[bx]

out dx,ax
jmp $+2

pop es
pop bp
ret 8
OUT$W ENDP
;=====
; Subroutine INP$W(word,address)
;=====
PUBLIC INP$W
INP$W PROC FAR
push bp
mov bp,sp

push es

les bx,DWORD PTR [bp+6] ;address
mov dx,es:[bx]

in ax,dx
jmp $+2

les bx,DWORD PTR [bp+10] ;word
mov es:[bx],ax
```

```

    pop    es
    pop    bp
    ret    8
INP$W ENDP
;=====
;    Subroutine INP$B(buffer,nwords,address)
;=====
    PUBLIC    INP$B
INP$B PROC FAR
    push    bp
    mov     bp,sp

    push    es
    push    di

    les    bx,DWORD PTR [bp+14]    ;BUFFER
    mov    di,bx

    mov    cx,es
    shr    di,4d
    add    cx,di
    and    bx,15d
    mov    di,bx
    mov    es,cx

    push    es

    les    bx,DWORD PTR [bp+10]    ;Nwords
    mov    cx,es:[bx]

    les    bx,DWORD PTR [bp+6]    ;input address
    mov    dx,es:[bx]

    pop    es
    cld
rep insw
    jmp    $+2

    pop    di
    pop    es
    pop    bp
    ret    12

INP$B ENDP
;=====
;    Subroutine OUTP$B(buffer,nwords,address)
;=====
    PUBLIC    .OUTP$B
OUTP$B PROC FAR
    push    bp
    mov     bp,sp

    push    es
    push    si
    push    ds

    les    bx,DWORD PTR [bp+14]    ;BUFFER
    mov    si,bx

```

```

        push    es
        pop     ds

        les    bx,DWORD PTR [bp+10]    ;Nwords
        mov    cx,es:[bx]

        les    bx,DWORD PTR [bp+6]    ;input address
        mov    dx,es:[bx]

        rep    outsw

        pop     ds
        pop     si
        pop     es
        pop     bp
        ret    12

```

```

OUTP$B ENDP

```

```

;=====
; Subroutine IRQ_EN
;=====

```

```

        PUBLIC    IRQ_EN
IRQ_EN    PROC FAR

```

```

        push    bp
        mov     bp,sp

```

```

        push    es
        STI
        pop     es
        pop     bp
        ret

```

```

IRQ_EN    ENDP

```

```

;=====
; Subroutine IRQ_DI
;=====

```

```

        PUBLIC    IRQ_DI
IRQ_DI    PROC FAR

```

```

        push    bp
        mov     bp,sp

```

```

        push    es
        CLI
        pop     es
        pop     bp
        ret

```

```

IRQ_DI    ENDP

```

```

NCASM_TEXT ENDS
END

```

## Bibliografia

- 1) **"Il rivelatore a 128 canali del sistema pulsar di Medicina"**  
*A.Cattani, C.Bortolotti,,A.Cattani, N.D'Amico, A.Maccafèrri, S.Montebugnoli*  
IRA 169/92
- 2) **"Il digitalizzatore ad 1 bit - 128 canali del sistema pulsar di Medicina"**  
*A.Cattani, S.Montebugnoli, N.D'Amico, A.Maccafèrri*  
IRA 170/92
- 3) **"Scheda di interfaccia per acquisizione dati ad alta velocità su bus ISA"**  
*A.Maccafèrri, N.D'Amico*  
IRA 154/92
- 4) **"Pulsar astronomy at the Northern Cross"**  
*N.D'Amico, C.Bortolotti, A.Cattani, F.Fauci, G.Grueff, A.Maccafèrri, S.Montebugnoli, M.Roma, G.Tomassetti.*  
IRA 137/90