Il dedispersore digitale del sistema "Pulsar" di Medicina

A.Maccaferri, N.D'Amico

Rapporto interno IRA 198/95

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:

$$\Delta t = \frac{DM}{1.2\ 10^{-4}} \frac{\Delta \nu}{\nu^3}$$

dove DM=pc cm3 Δν=MHz
ν=MHz Δt=sec

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 dedisperdere 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 spacchettamento 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 spacchettamento dei dati al momento dell'analisi. La lettura, lo spacchettamento e la somma "software" dei dati risulta alquanto inefficente, 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 risolta su 128 canali di frequenza. La lettura dei singoli campionamenti avviene senza spacchettamento (dato che la memoria é ad 1 bit), ed é quindi molto efficente. 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.

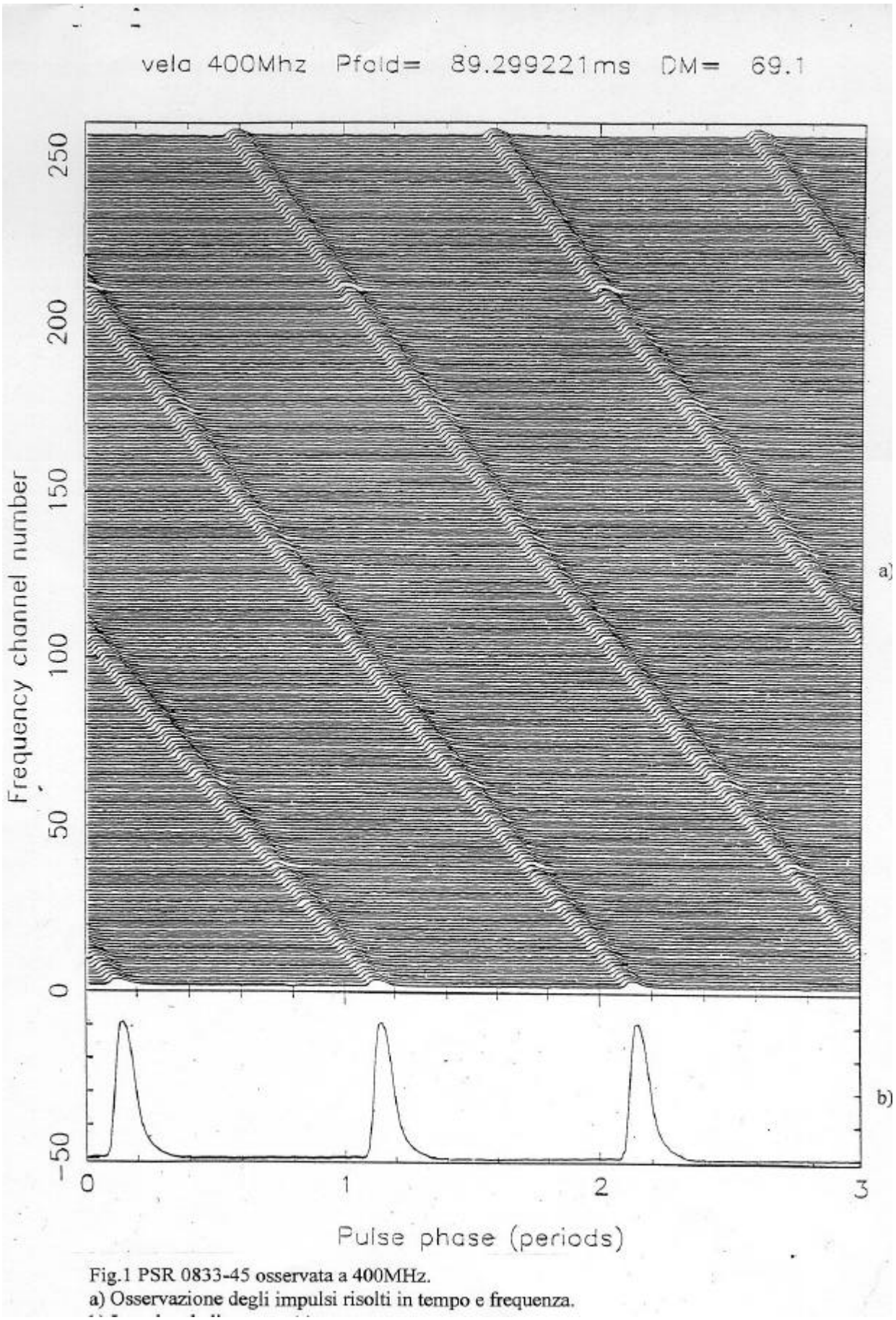


Fig.1 PSR 0833-45 osservata a 400MHz.
a) Osservazione degli impulsi risolti in tempo e frequenza.
b) Impulso dedisperso ed integrato

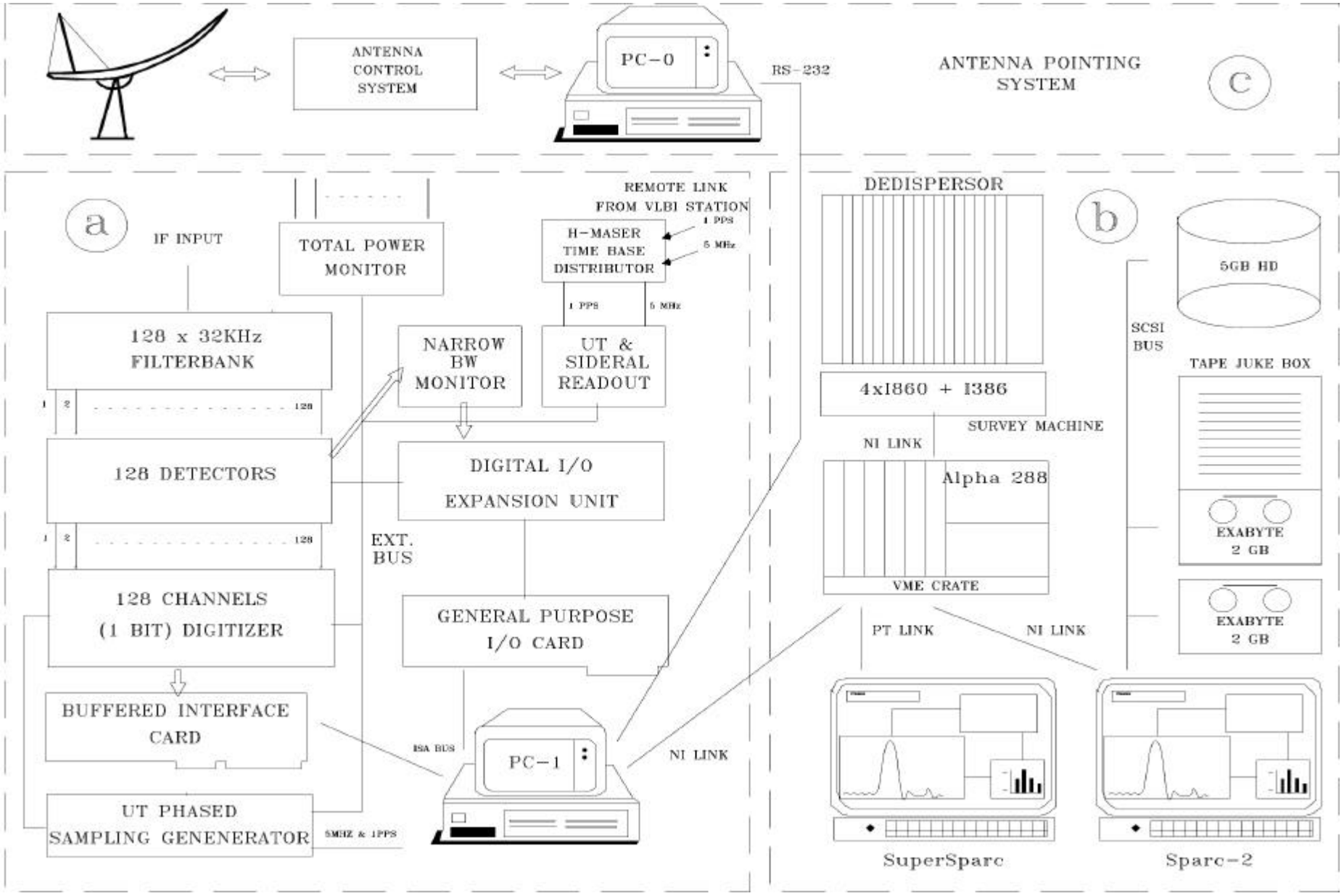


Fig 2 Sistema Pulsar di Medicina. a) Sottosistema di acquisizione dati b) Sottosistema di analisi e memorizzacione c) Sistema di puntamento


DIGITAL
DEDISPERSOR
4 X I860 VECTOR
PROCESSOR
& Dedispersor
controller
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 giá predisposto per alloggiare un secondo controller che potrá consentire l'accesso al dedispersore anche a schede poste nel VME o ad altri dispositivi. A questo scopo é giá 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, ció permette di sfruttare la massima velocitá 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 ció avrebbe richiesto una singola memoria con campo di indirizzamento pari a 2^{128} Byte, (i piú 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 velocitá 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 sommatori di 15 parole da 1 bit, ciascuna di peso 2^0, il secondo ed il terzo stadio sono composti da sommatori 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.

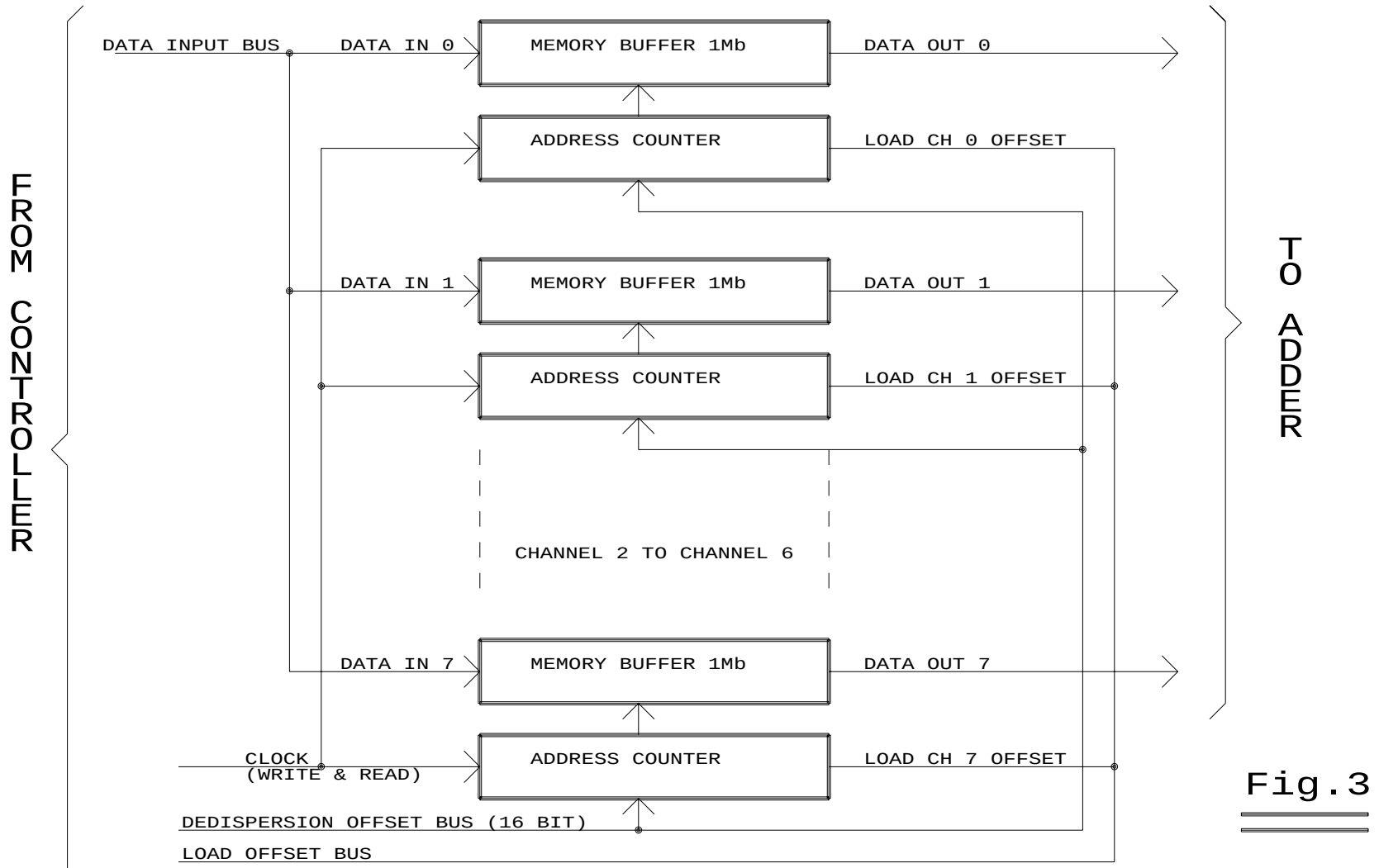


Fig.3

C.N.R. IST. di RADIOASTRONOMIA RADIOTELESCOPIO CROCE DEL NORD Bologna, Italy		
Title DEDISPERSOR: 8 CHANNEL MEMORY BUFFER		
Size A	Document Number file name: blockmem.sch	REV
Date: October 13, 1998	Sheet of	

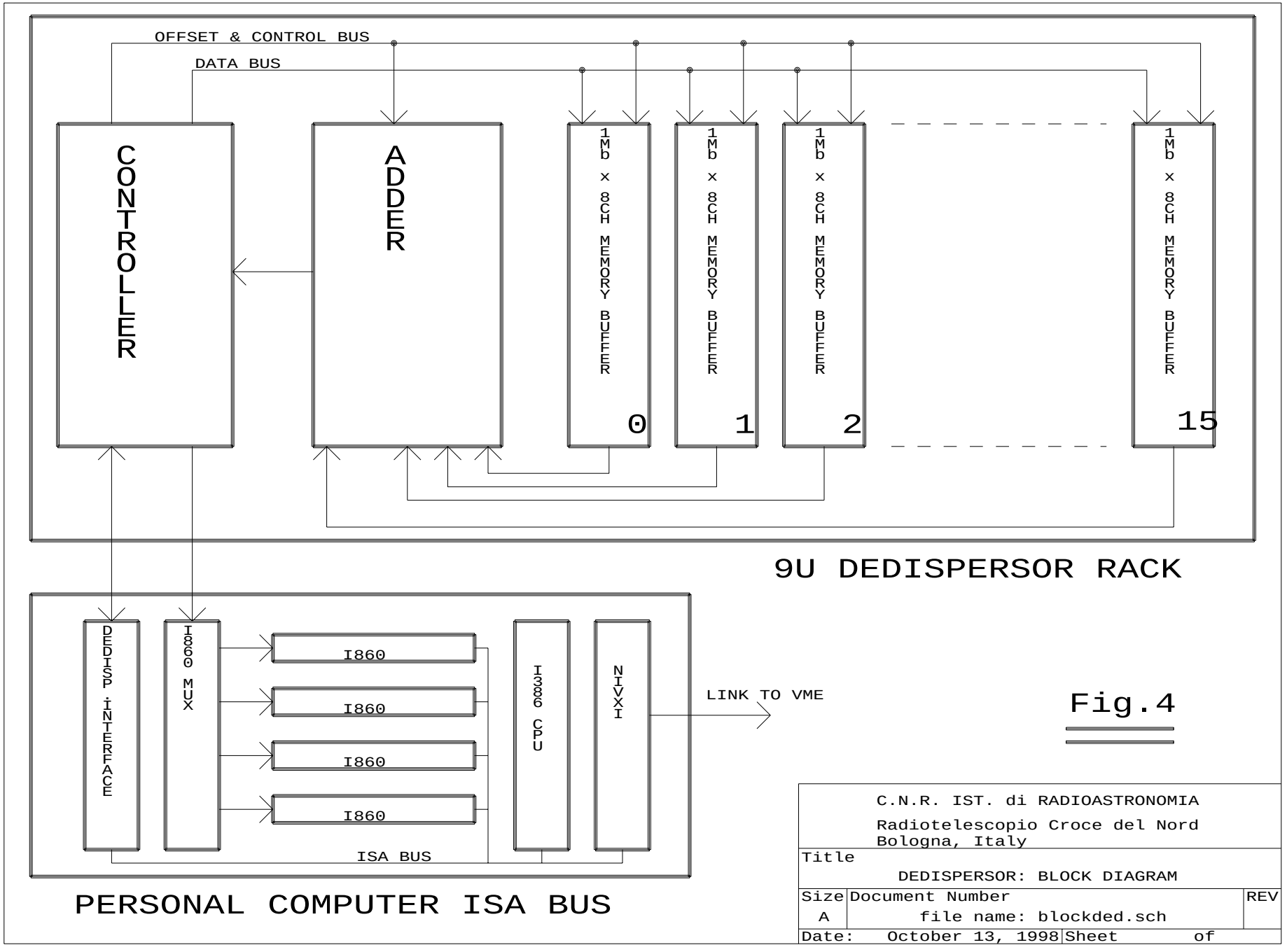


Fig.4

C.N.R. IST. di RADIOASTRONOMIA Radiotelescopio Croce del Nord Bologna, Italy		
Title DEDISPERSOR: BLOCK DIAGRAM		
Size A	Document Number file name: blockded.sch	REV
Date: October 13, 1998	Sheet of	

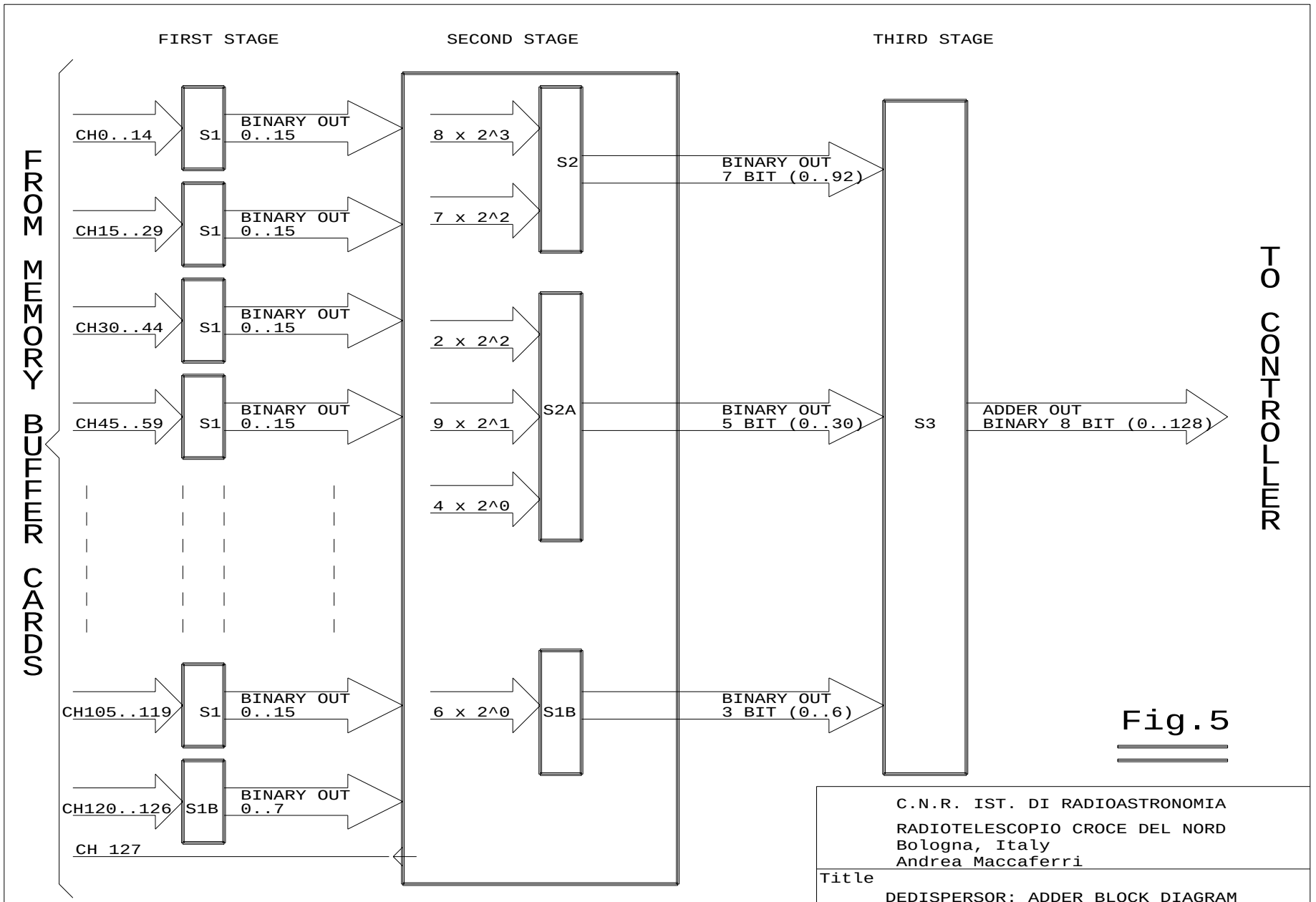


Fig.5

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

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

Title
DEDISPERSOR: ADDER BLOCK DIAGRAM

Size	Document Number	REV
A	FILE NAME: BLOCKADD.SCH	
Date: April 3, 1995	Sheet of	

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 giá dedispersi, configurare il tutto in base al numero di canali ed al fattore di dedispersione desiderato, eseguire inoltre 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 (2 byte impacchettati in una parola a 16 bit, byte L dato pari, H dispari).
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. Per aumentare il *transfer rate* vengono trasferiti contemporaneamente i dati relativi a due campionamenti, nella parte bassa della parola vi è il risultato relativo al campionamento pari (0, 2 etc..) nella parte alta i campionamenti dispari (1, 3 etc..). Precedentemente alla prima lettura di una serie, occorre inizializzare il registro (vedi registro Base+6).

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

Base+2 :

E' il registro dove scrivere l'offset di ciascun canale (in unitá di campionamenti) per la dedispersione. Il registro é a 16 bit, si puó 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 puó 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 Porta Base+6	Descrizione	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	**P**rimo stadio	P15	P14	P13	...	...	...	...	...	...	...	...	...	...	...	...	P0
9	**P**rimo stadio	P31	...	...	...	...	...	...	...	...	...	...	...	...	...	...	P16
10	**S**econdo stadio	x	S14	...	...	...	...	...	...	...	...	...	...	...	...	...	S0
11	**R**isultato e **P**rino 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	120 127	104 111	88 95	72 79	57 63	40 47	24 31	8 15	112 119	96 103	80 87	64 71	48 55	32 39	16 23	0 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;
impostare durante la fase di riempimento del buffer, in blocchi di 16, il numero di canali utilizzati, tramite le linee WSEL0..WSEL2
inizializzare il dedispersore per la lettura dati dedispersi, effettuando una scrittura prima a 0, poi a 1 del bit 15 (IN_RD).

Bit	15	14	13	12	11	10	9	8	7	6..1	0
	IN_RD	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 più 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

Lettura 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
- Inizializzare il registro per la lettura dati dedispersi, effettuando un impulso 1,0,1 sul segnale IN_RD, bit 15 Base+6.
- 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

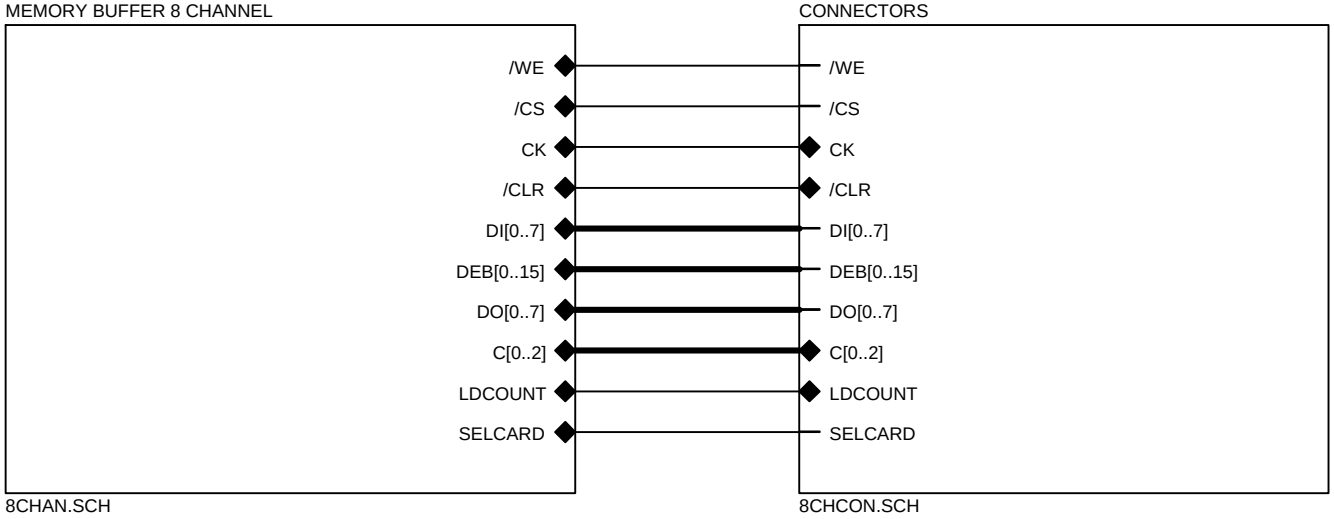
Diagnostica

- 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 presettare 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, presettabile 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 selezionare 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 presettare 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.



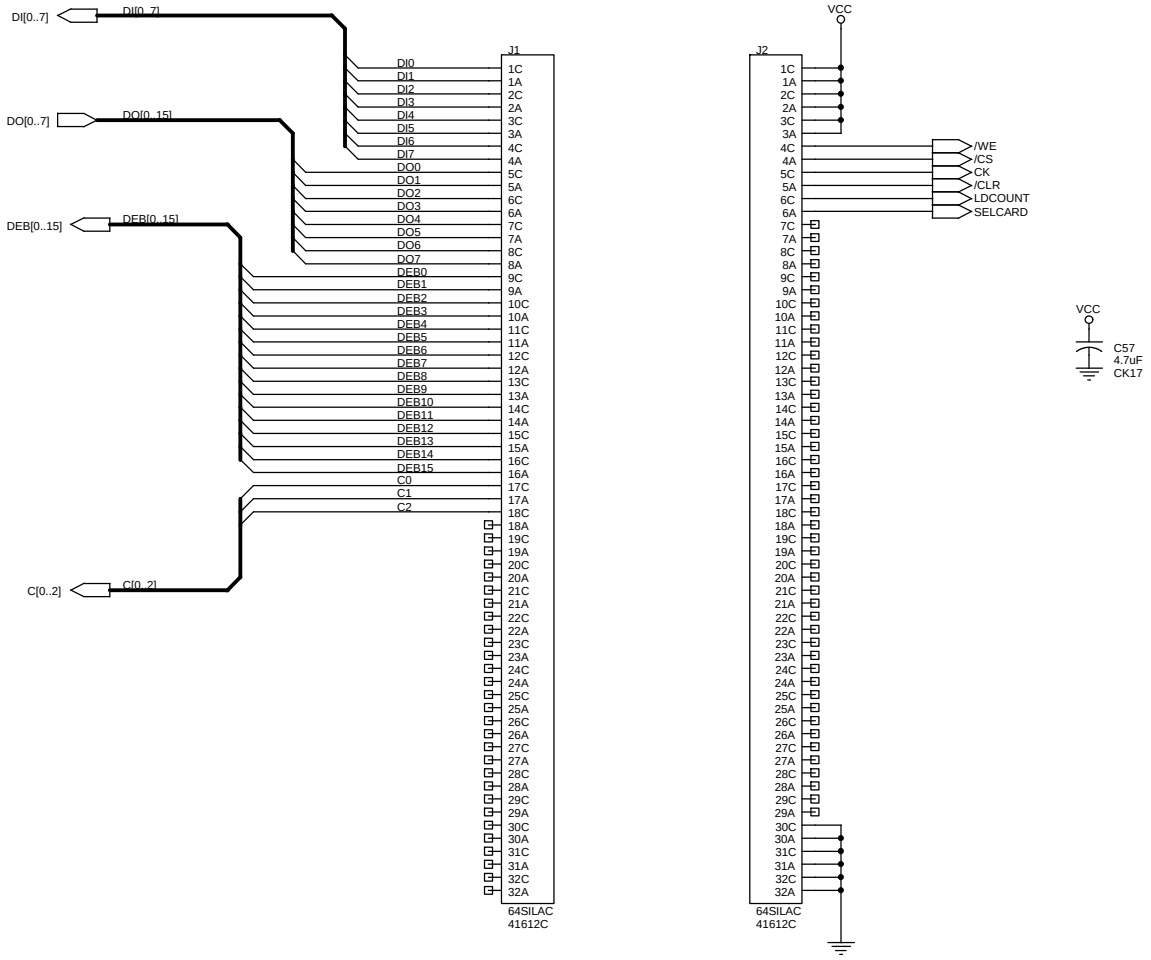
C.N.R. RADIOASTRONOMIA

RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
A.MACCAFERRI

Title
DEDISPERSORE: 8 CHANNEL CARD

Size	Document Number	Rev
A	FILE NAME: DED8.SCH	2

Date: Thursday, May 04, 2000 Sheet 1 of 11



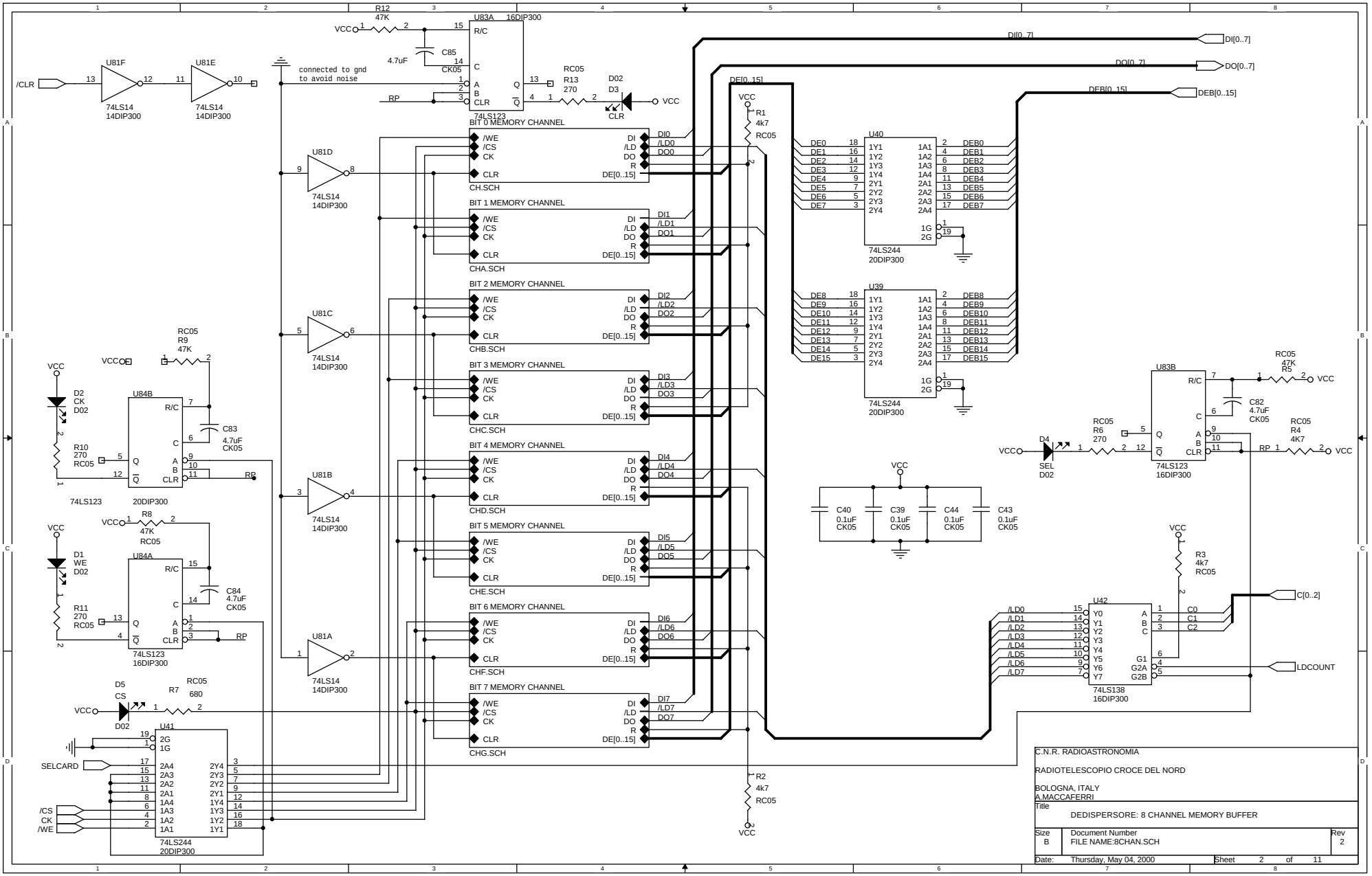
C.N.R. RADIOASTRONOMIA

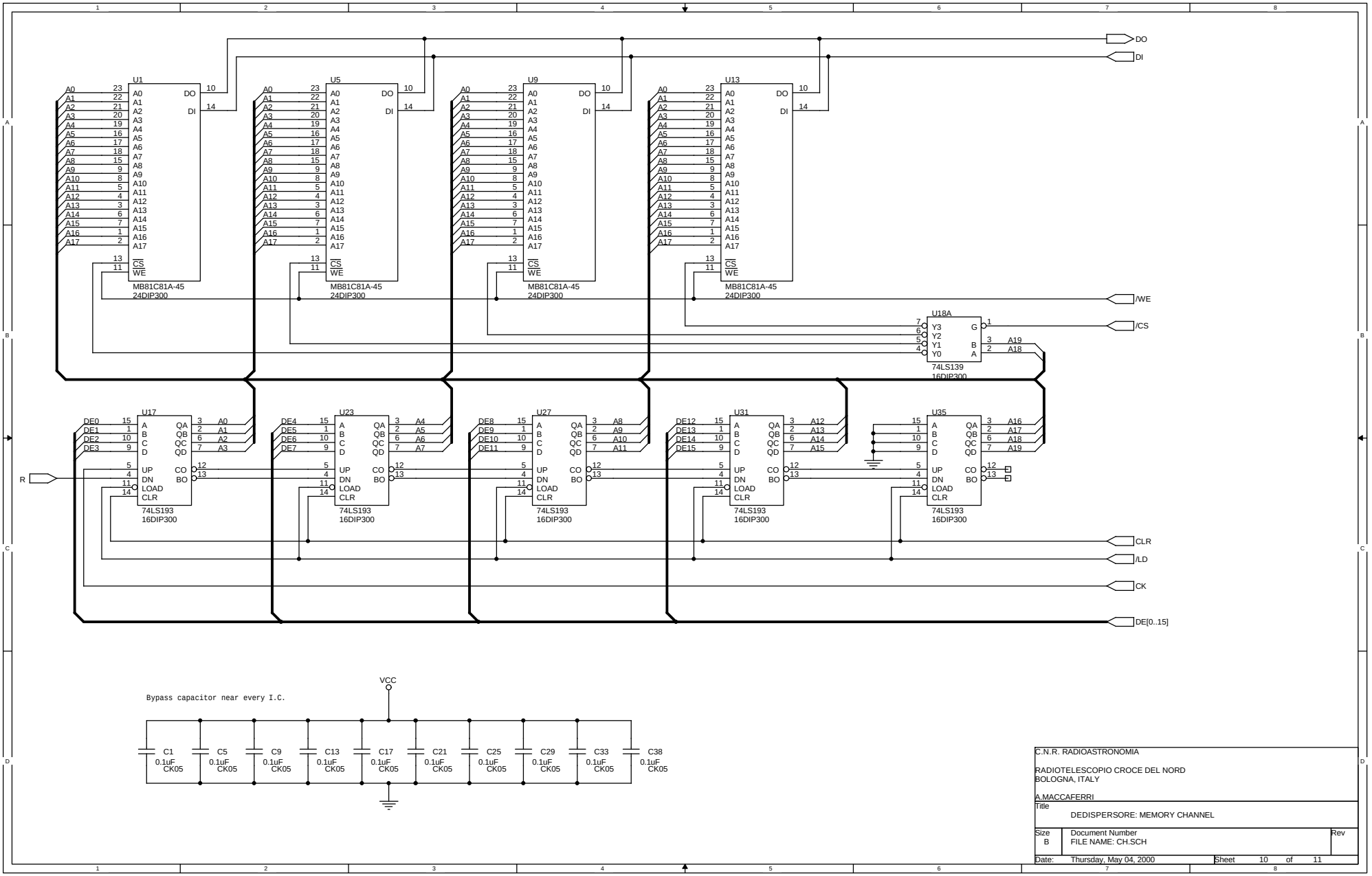
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
A.MACCAFERRI

Title: DEDISPERSORE: CONNETTORI SCHEDA 8 CANALI

Size	Document Number	Rev
B	FILE NAME: 8CHCON.SCH	2

Date: Thursday, May 04, 2000 Sheet 11 of 11


/CLR
U81F
U81E
74LS14
14DIP300
connected to gnd
to avoid noise
R12
47K
VCC
U83A
16DIP300
R/C
C
A
B
CLR
Q
C85
4.7uF
CK05
RP
RC05
R13
270
D02
D3
74LS123
BIT 0 MEMORY CHANNEL
BIT 1 MEMORY CHANNEL
BIT 2 MEMORY CHANNEL
BIT 3 MEMORY CHANNEL
BIT 4 MEMORY CHANNEL
BIT 5 MEMORY CHANNEL
BIT 6 MEMORY CHANNEL
BIT 7 MEMORY CHANNEL
/WE
/CS
CK
CLR
DI
/LD
DO
R
DE[0..15]
CH.SCH
CHA.SCH
CHB.SCH
CHC.SCH
CHD.SCH
CHE.SCH
CHF.SCH
CHG.SCH
DI0
/LD0
DO0
DI1
/LD1
DO1
DI2
/LD2
DO2
DI3
/LD3
DO3
DI4
/LD4
DO4
DI5
/LD5
DO5
DI6
/LD6
DO6
DI7
/LD7
DO7
U81D
U81C
U81B
U81A
DI[0..7]
DO[0..7]
DE[0..15]
DEB[0..15]
R1
4k7
RC05
R2
U40
U39
74LS244
20DIP300
DE0
DE1
DE2
DE3
DE4
DE5
DE6
DE7
DE8
DE9
DE10
DE11
DE12
DE13
DE14
DE15
DEB0
DEB1
DEB2
DEB3
DEB4
DEB5
DEB6
DEB7
DEB8
DEB9
DEB10
DEB11
DEB12
DEB13
DEB14
DEB15
1G
2G
C40
0.1uF
CK05
C39
C44
C43
U83B
R5
C82
4.7uF
R6
R4
D4
SEL
D02
R3
U42
74LS138
C[0..2]
LDCOUNT
U84B
U84A
U41
R9
R8
R7
680
R10
R11
C83
C84
D1
WE
D2
CS
D5
SELCARD
/CS
CK
/WE
20DIP300
16DIP300
C.N.R. RADIOASTRONOMIA
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
A.MACCAFERRI
Title
DEDISPERSORE: 8 CHANNEL MEMORY BUFFER
Size
B
Document Number
FILE NAME:8CHAN.SCH
Rev
2
Date:
Thursday, May 04, 2000
Sheet
2
of
11



C.N.R. RADIOASTRONOMIA

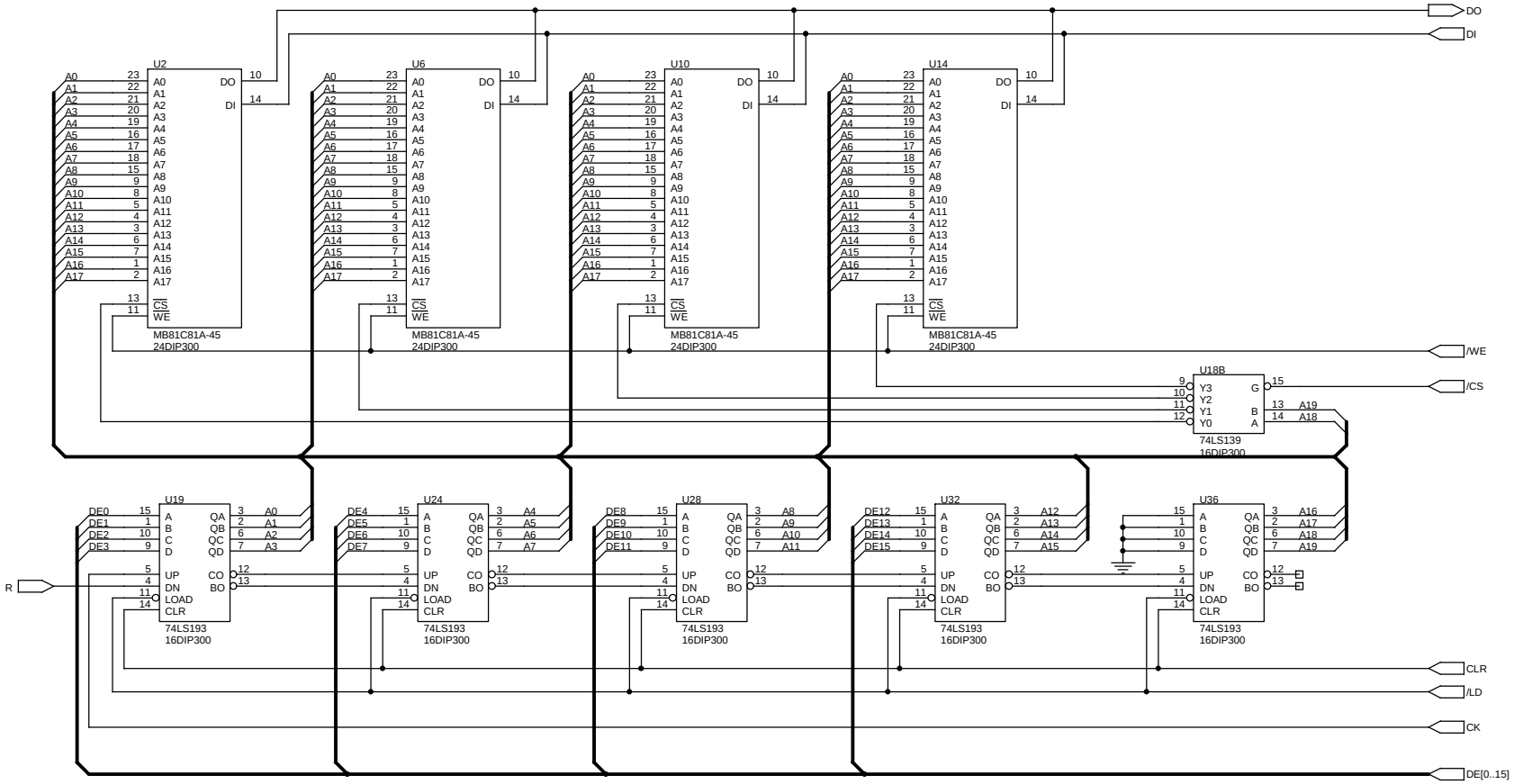
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY

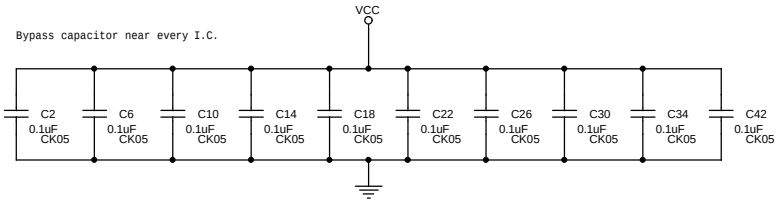
A.MACCAFERRI

Title: DEDISPERSORE: MEMORY CHANNEL

Size	Document Number	Rev
B	FILE NAME: CH.SCH	

Date: Thursday, May 04, 2000 Sheet 10 of 11





C.N.R. RADIOASTRONOMIA

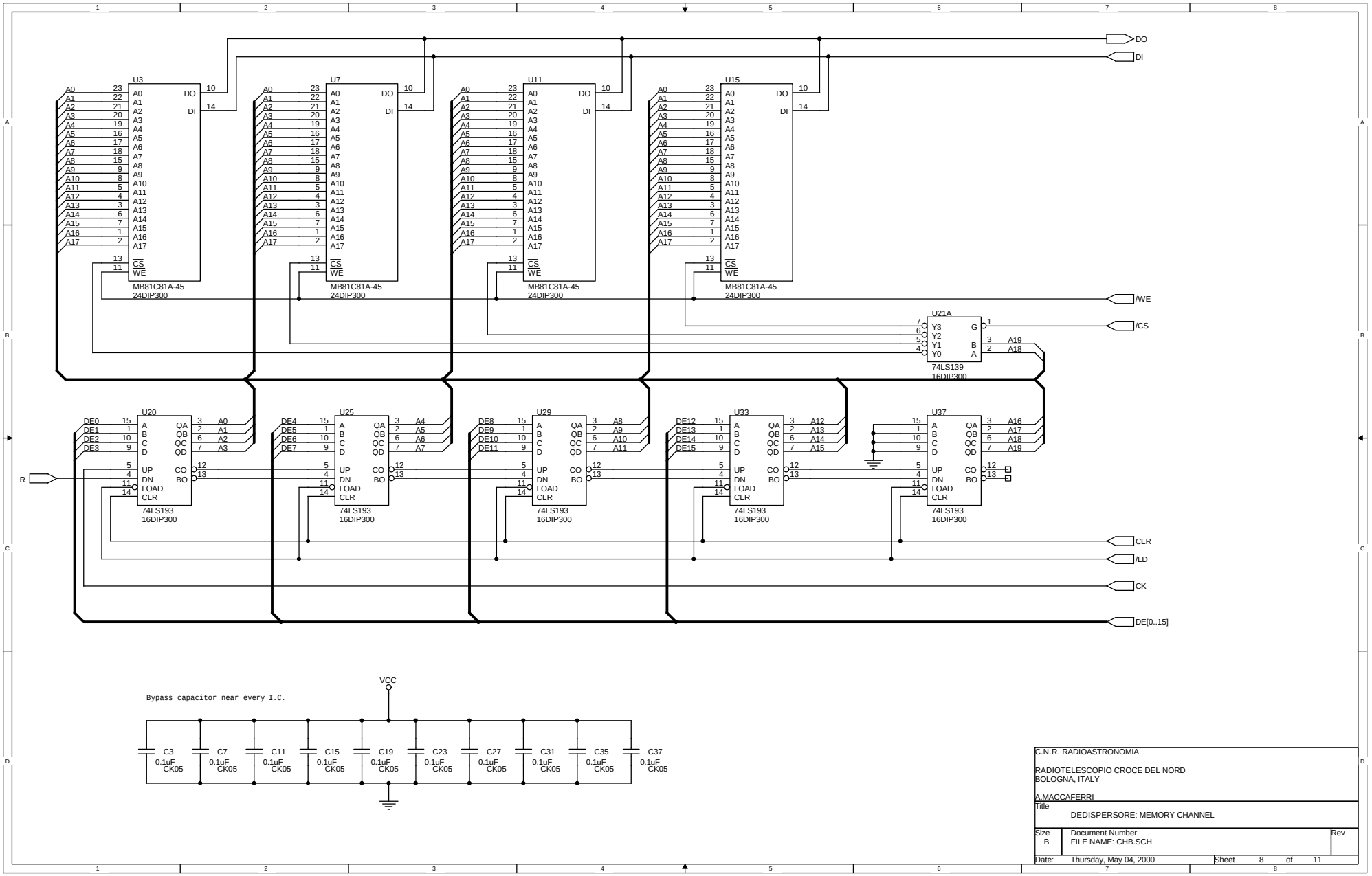
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY

A.MACCAFERRI

Title: DEDISPERSORE: MEMORY CHANNEL

Size: B | Document Number: FILE NAME: CHA.SCH | Rev:

Date: Thursday, May 04, 2000 | Sheet 9 of 11



C.N.R. RADIOASTRONOMIA

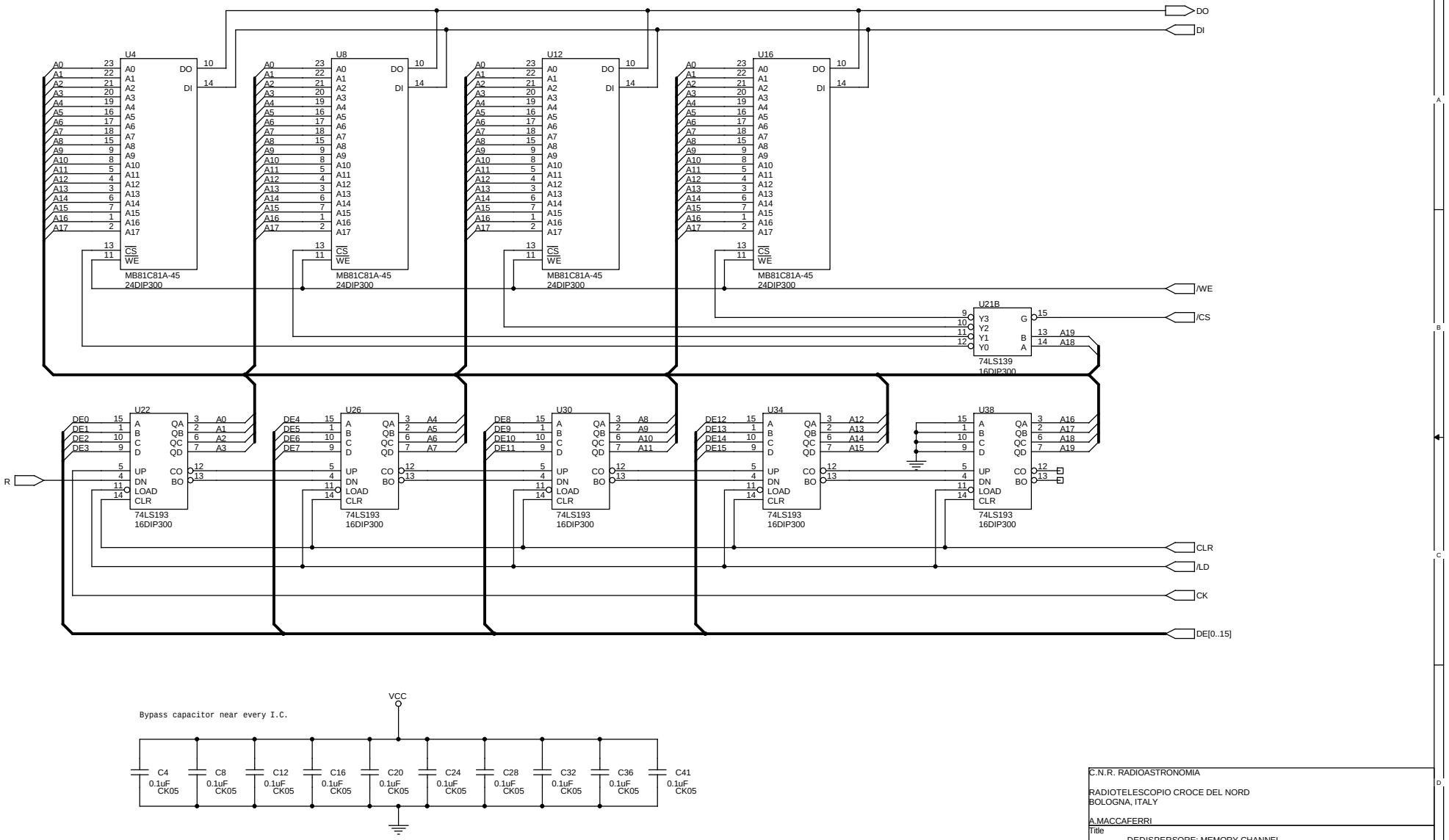
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY

A.MACCAFERRI

Title: DEDISPERSORE: MEMORY CHANNEL

Size	Document Number	Rev
B	FILE NAME: CHB.SCH	

Date: Thursday, May 04, 2000 Sheet 8 of 11



C.N.R. RADIOASTRONOMIA

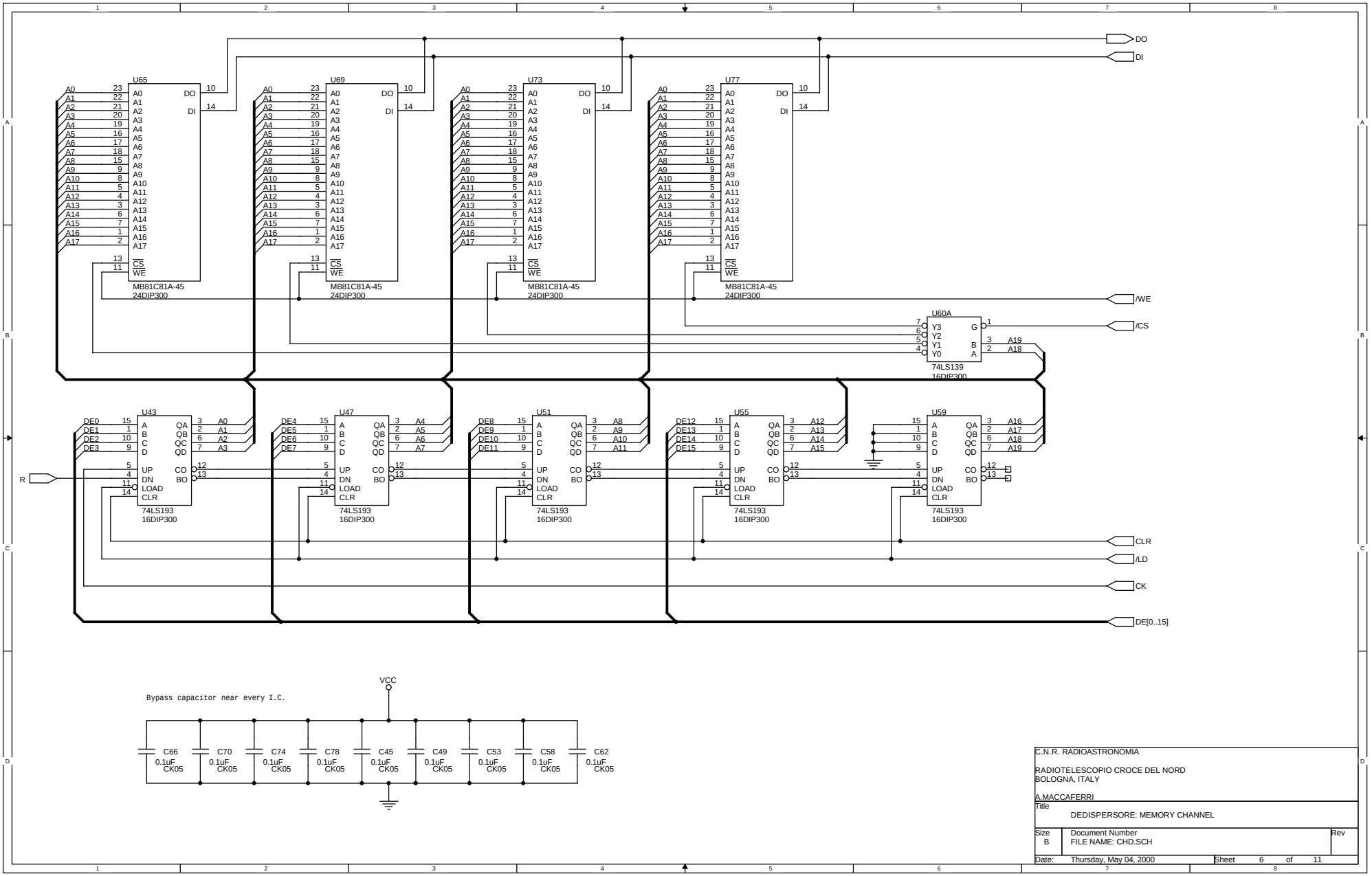
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY

A.MACCAFERRI

Title: DEDISPERSORE: MEMORY CHANNEL

Size: B | Document Number: FILE NAME: CHC.SCH | Rev:

Date: Thursday, May 04, 2000 | Sheet 7 of 11



C.N.R. RADIOASTRONOMIA

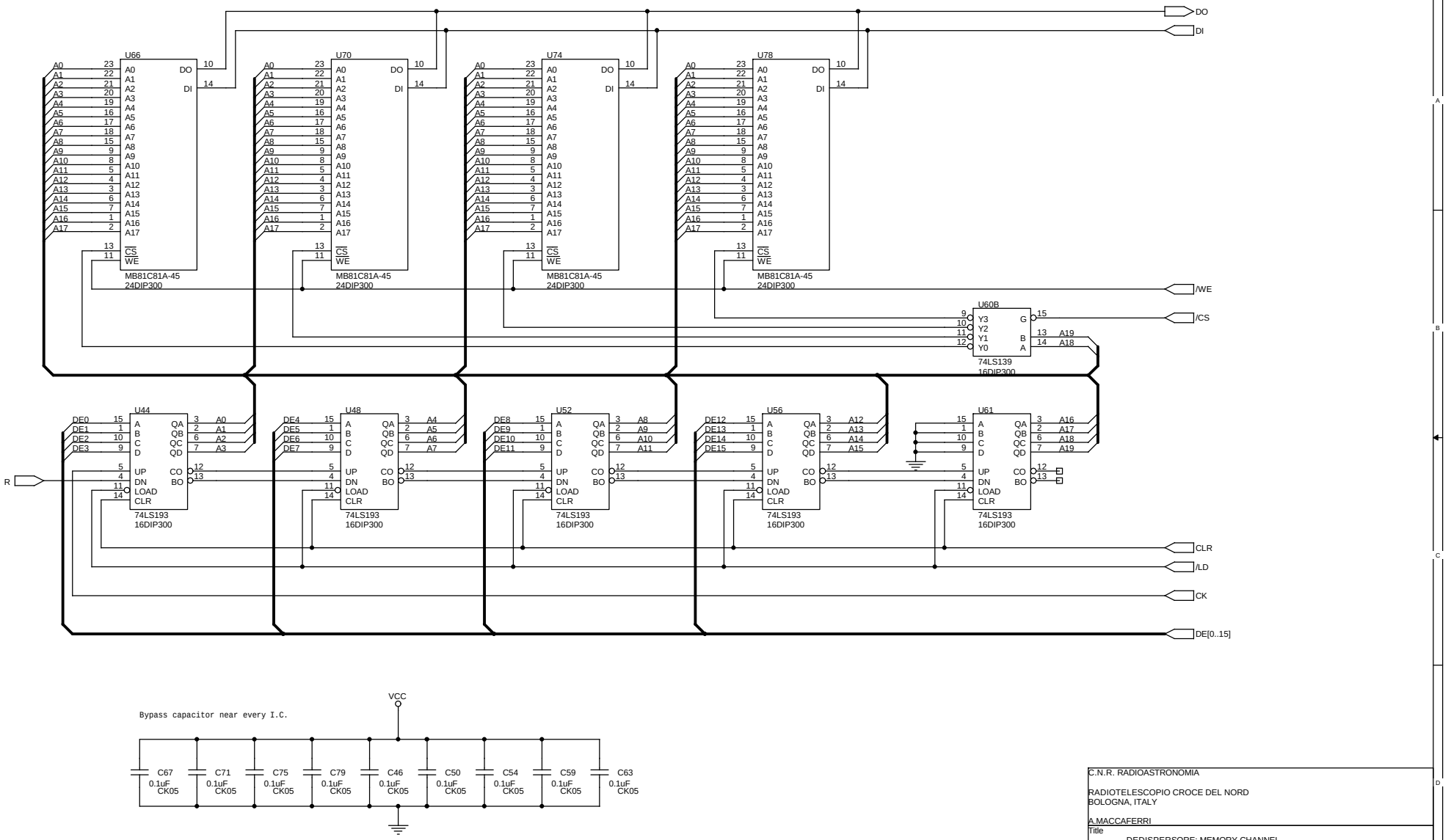
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY

A.MACCAFERRI

Title: DEDISPERSORE: MEMORY CHANNEL

Size	Document Number	Rev
B	FILE NAME: CHD.SCH	

Date: Thursday, May 04, 2000 Sheet 6 of 11



C.N.R. RADIOASTRONOMIA

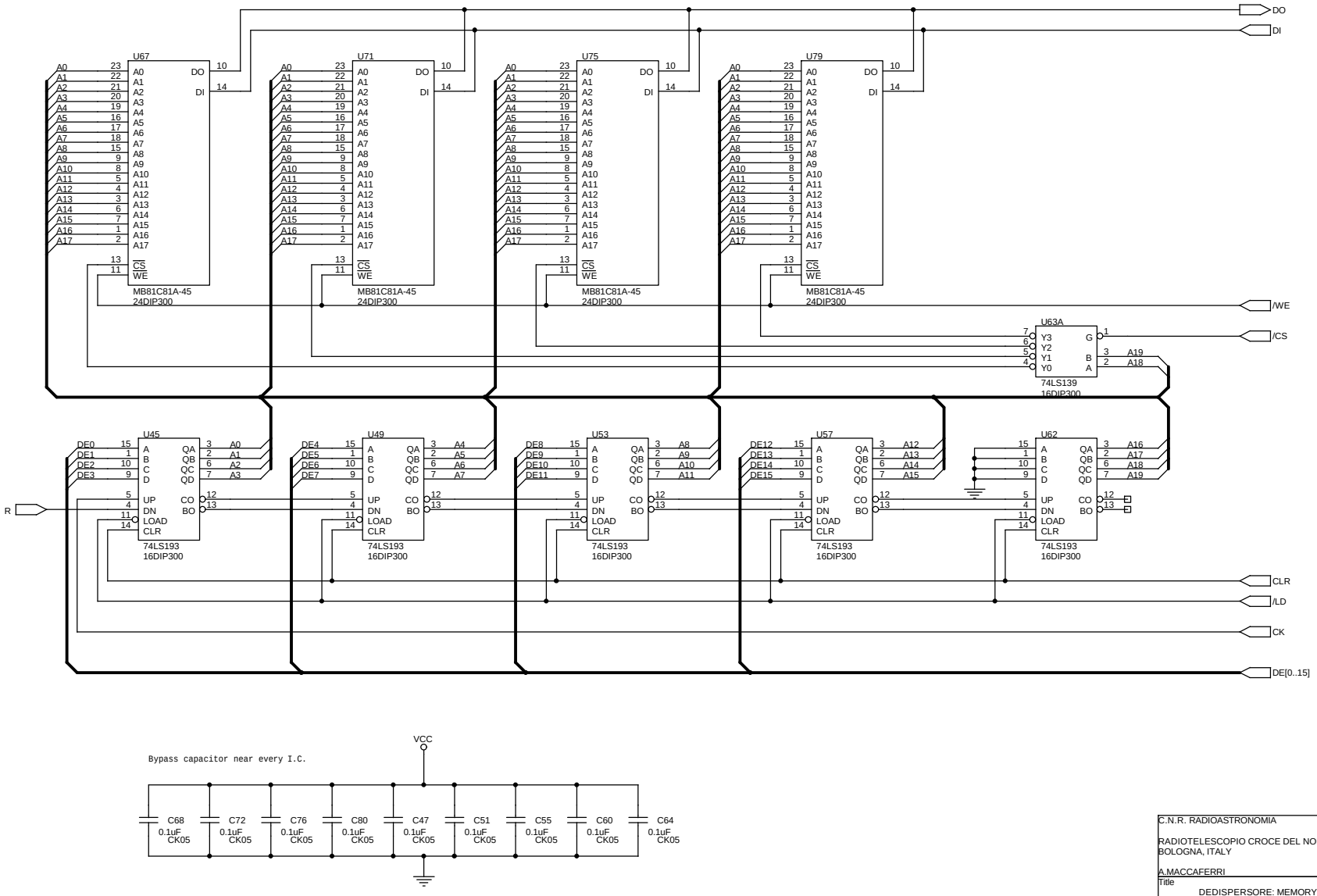
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY

A.MACCAFERRI

Title: DEDISPERSORE: MEMORY CHANNEL

Size	Document Number	Rev
B	FILE NAME: CHE.SCH	

Date: Thursday, May 04, 2000 Sheet 5 of 11



C.N.R. RADIOASTRONOMIA

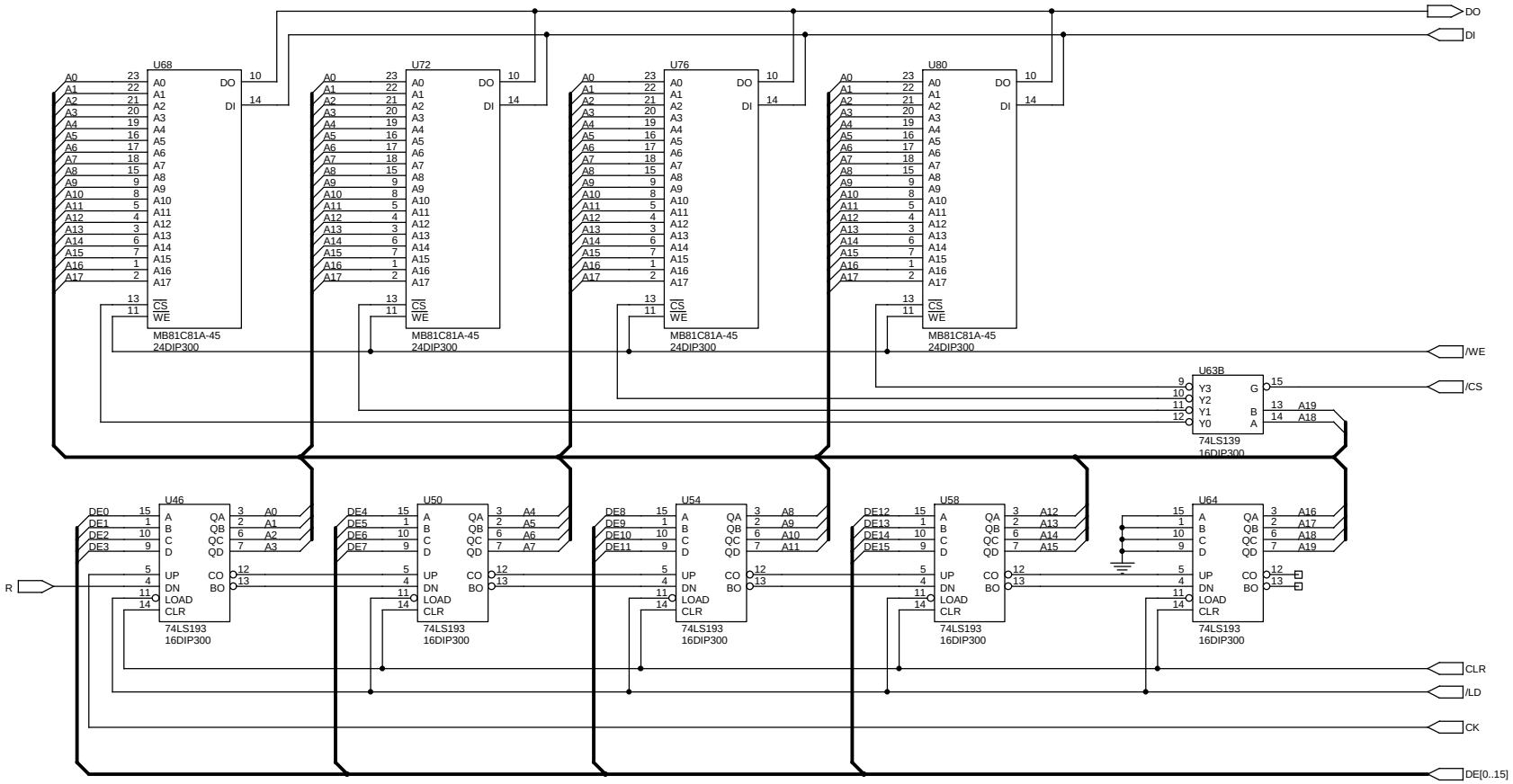
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY

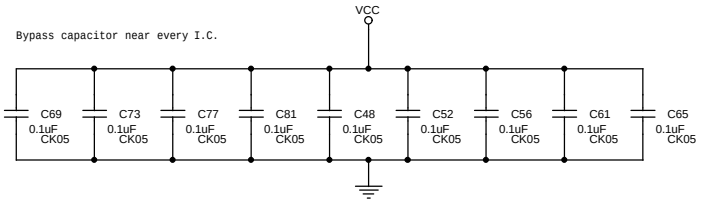
A.MACCAFERRI

Title: DEDISPERSORE: MEMORY CHANNEL

Size: B | Document Number: FILE NAME: CHF.SCH | Rev:

Date: Thursday, May 04, 2000 | Sheet 4 of 11





C.N.R. RADIOASTRONOMIA

RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY

A.MACCAFERRI

Title: DEDISPERSORE: MEMORY CHANNEL

Size	Document Number	Rev
B	FILE NAME: CHG.SCH	

Date: Thursday, May 04, 2000 Sheet 3 of 11

ISTITUTO DI RADIOASTRONOMIA C.N.R. (BO)
ISTITUTO DI FISICA UNIVERSITA' (PA)

8 CHANNELS DEDISPERSOR
PULSAR DATA ANALYZER SYSTEM

4.7UF
CS7

41612C

41612C

- Sommatore

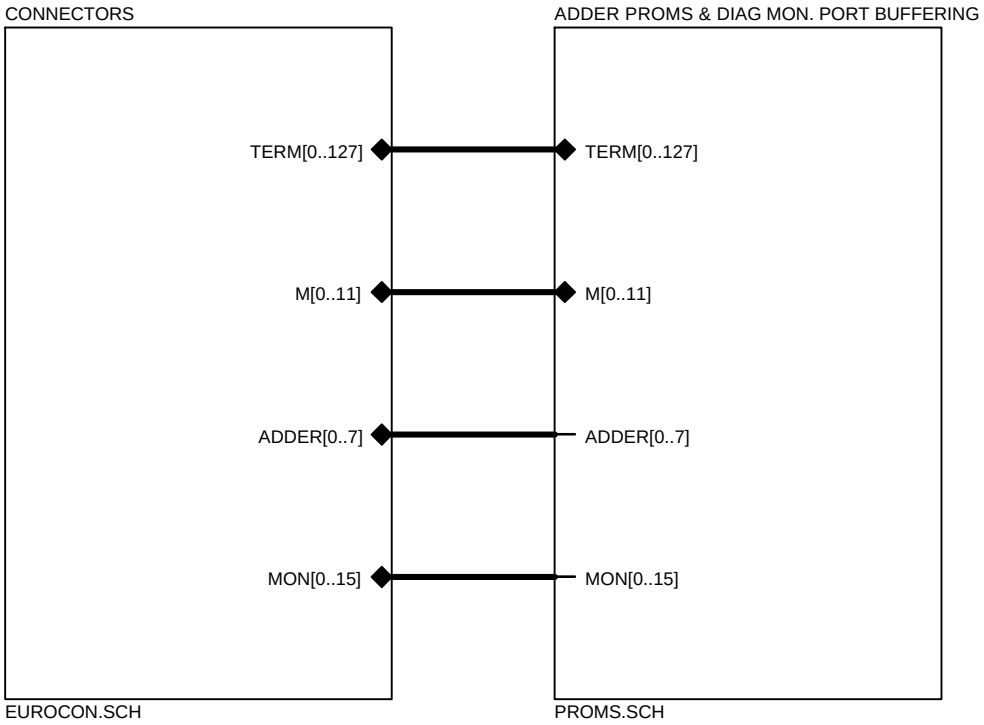
Nella scheda sommatore possiamo individuare 3 sezioni che compongono il sommatore vero e proprio (1° stadio, 2° stadio e 3° stadio), vi sono poi tutta una serie di buffer che permettono di leggere i vari bit in ingresso dai buffer di memoria o i risultati parziali dei vari stadi.
Il 1° stadio del sommatore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



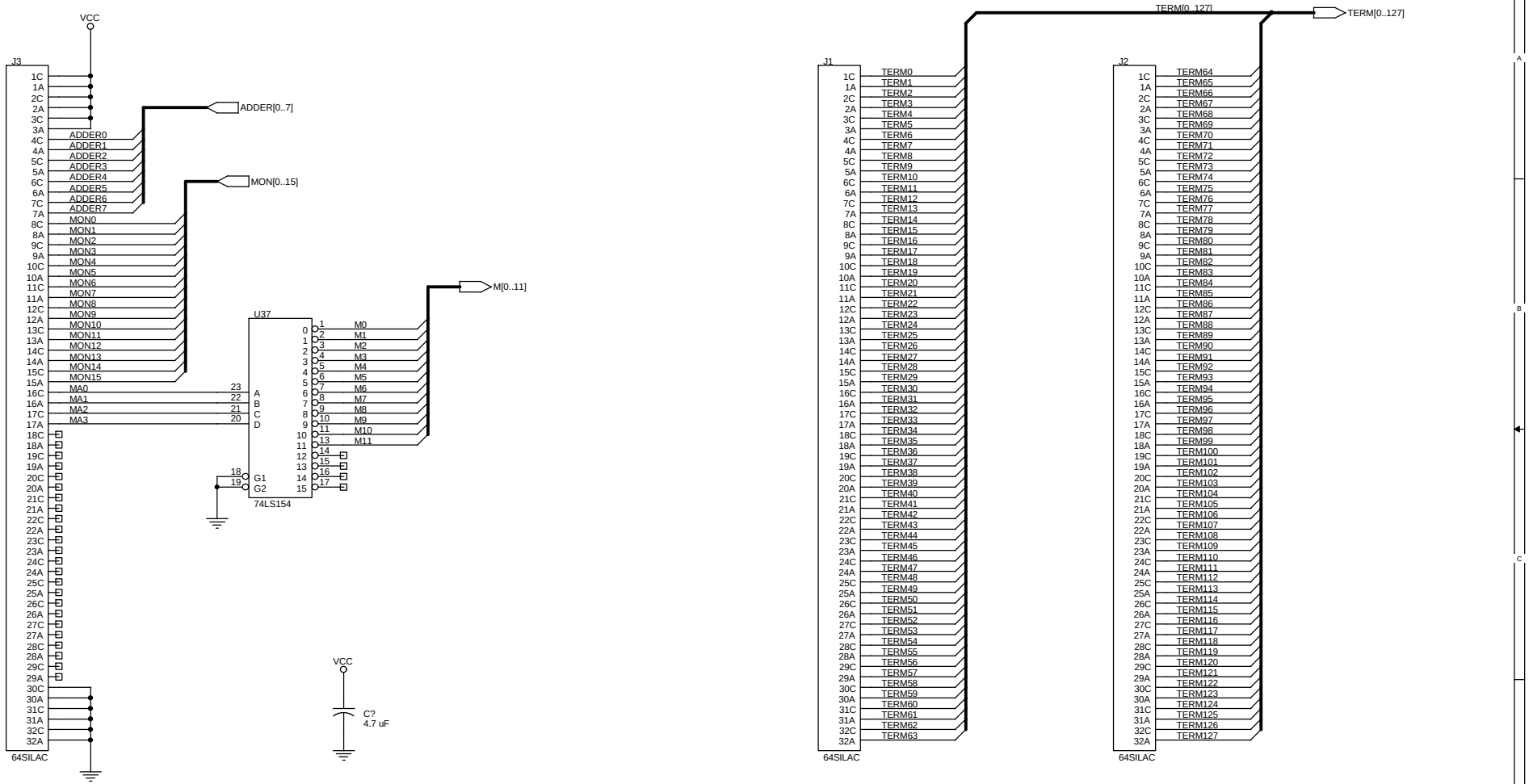
C.N.R. RADIOASTRONOMIA

RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY

Title
DEDISPERSOR: ADDER CARD MAIN BLOCK

Size	Document Number	Rev
A	FILE NAME: ADDER.SCH	

Date: Thursday, May 04, 2000 Sheet 0 of 0



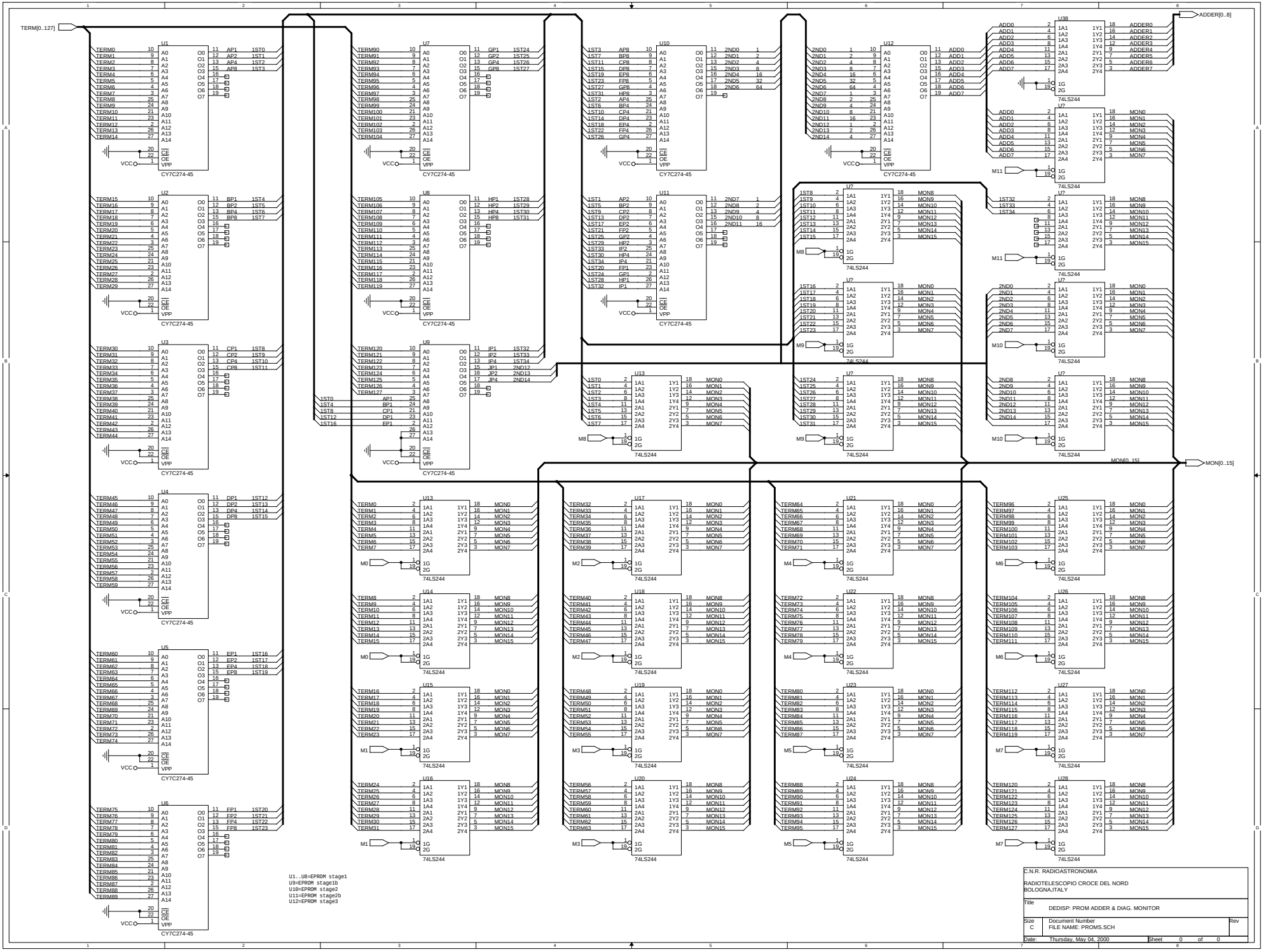
C.N.R. RADIOASTRONOMIA

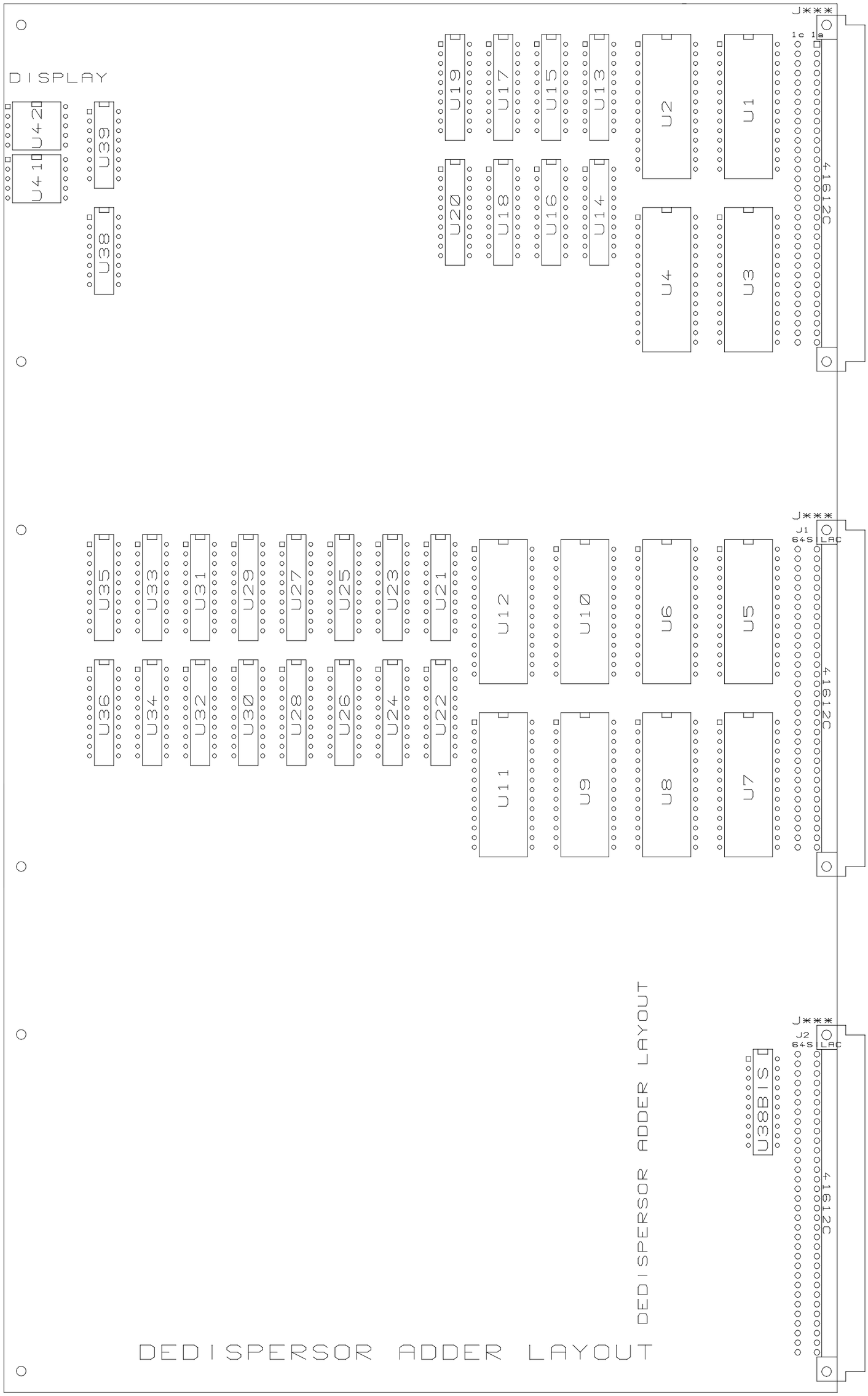
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA,ITALY
A.MACCAFERRI

Title: DEDISP: ADDER CONN. & DIAG.MONITOR DEC.

Size	Document Number	Rev
B	FILE NAME: EUROCON.SCH	

Date: Thursday, May 04, 2000 Sheet 0 of 0


TERM[0..127]
ADDER[0..8]
MON[0..15]
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
U16
U17
U18
U19
U20
U21
U22
U23
U24
U25
U26
U27
U28
CY7C274-45
74LS244
U1..U8=EPROM stage1
U9=EPROM stage1b
U10=EPROM stage2
U11=EPROM stage2b
U12=EPROM stage3
C.N.R. RADIOASTRONOMIA
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA,ITALY
Title
DEDISP: PROM ADDER & DIAG. MONITOR
Size C
Document Number
FILE NAME: PROMS.SCH
Rev
Date: Thursday, May 04, 2000
Sheet 0 of 0



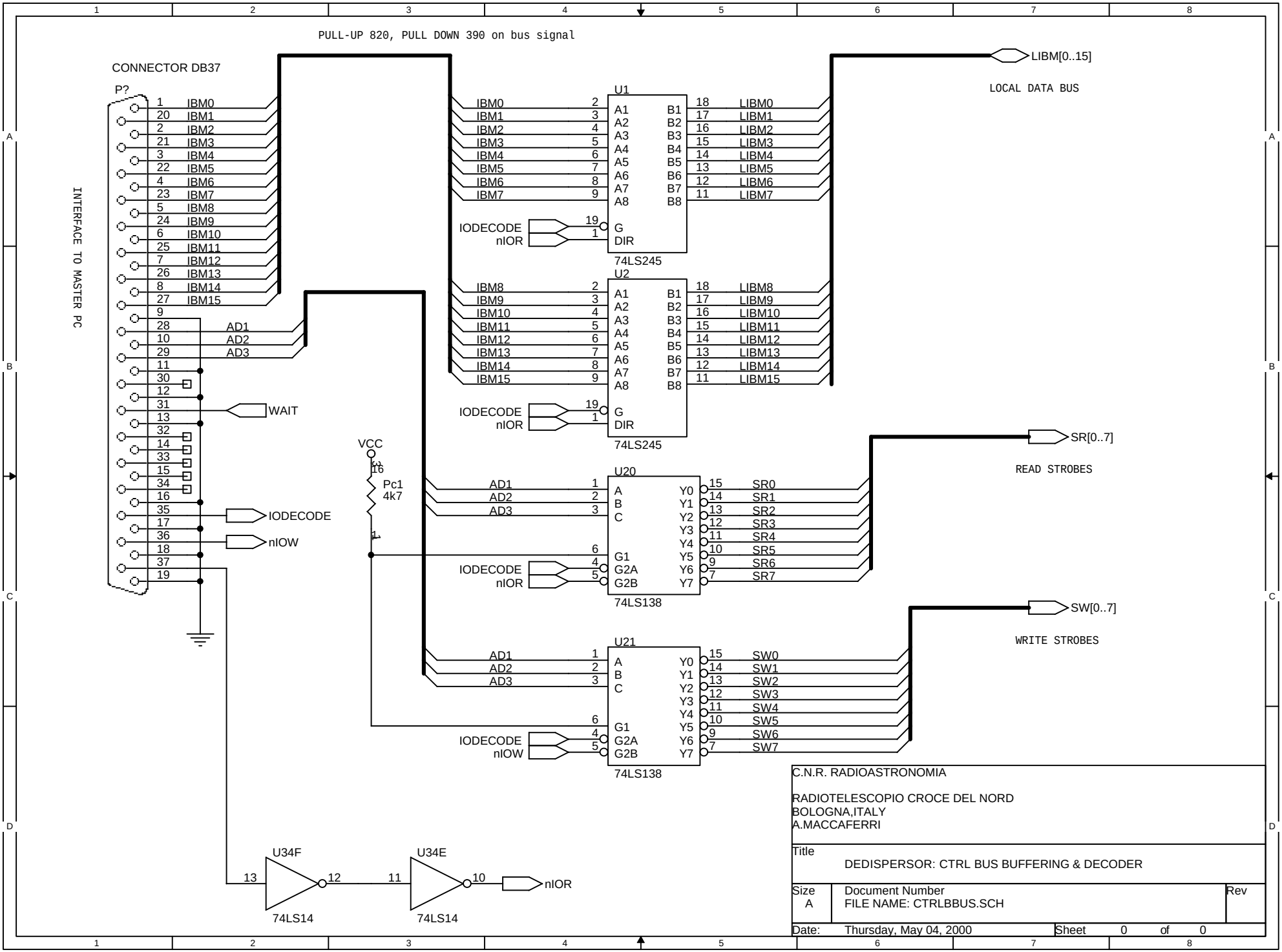
DEDISPERSOR ADDER LAYOUT

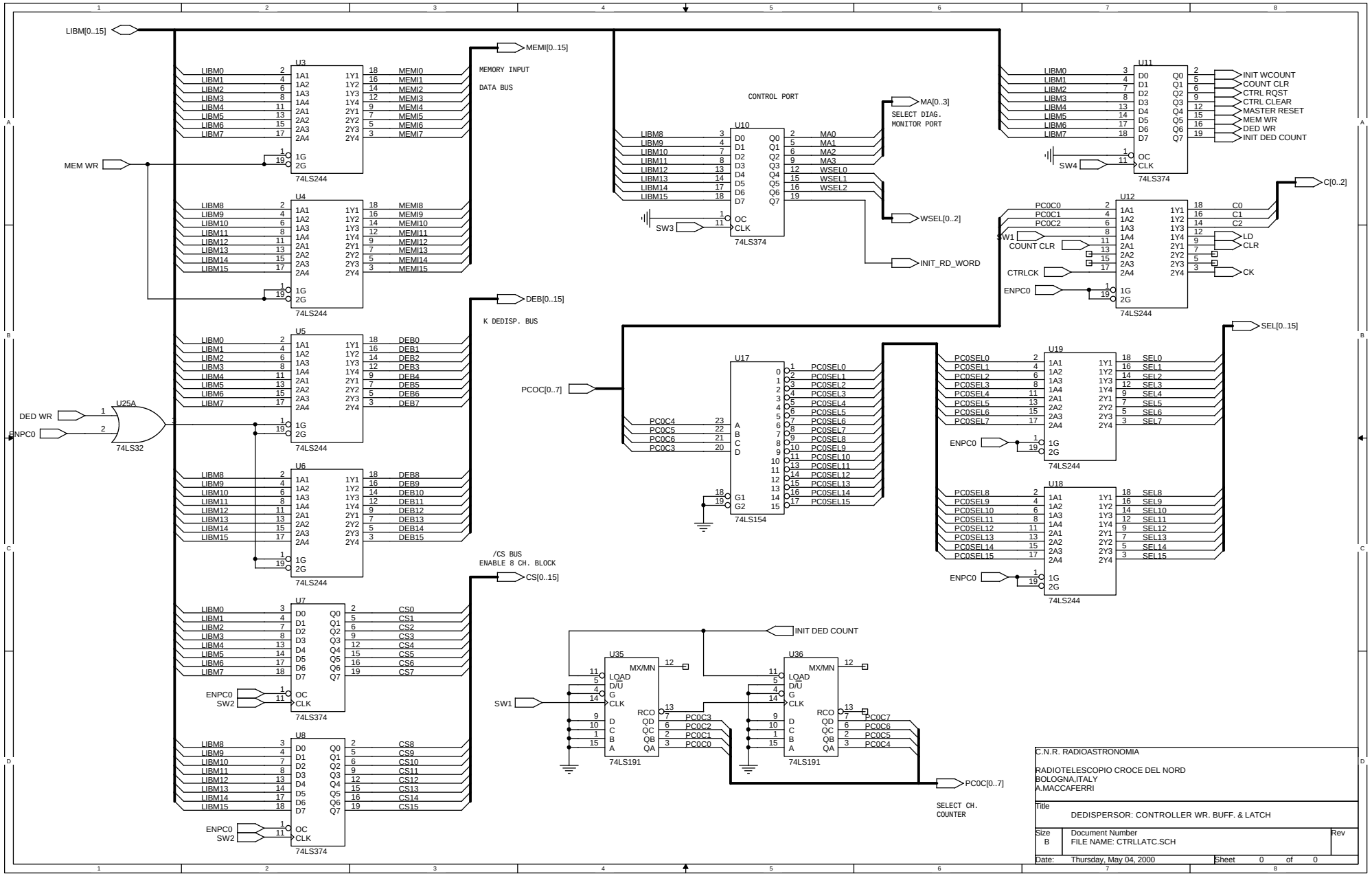
- Controller

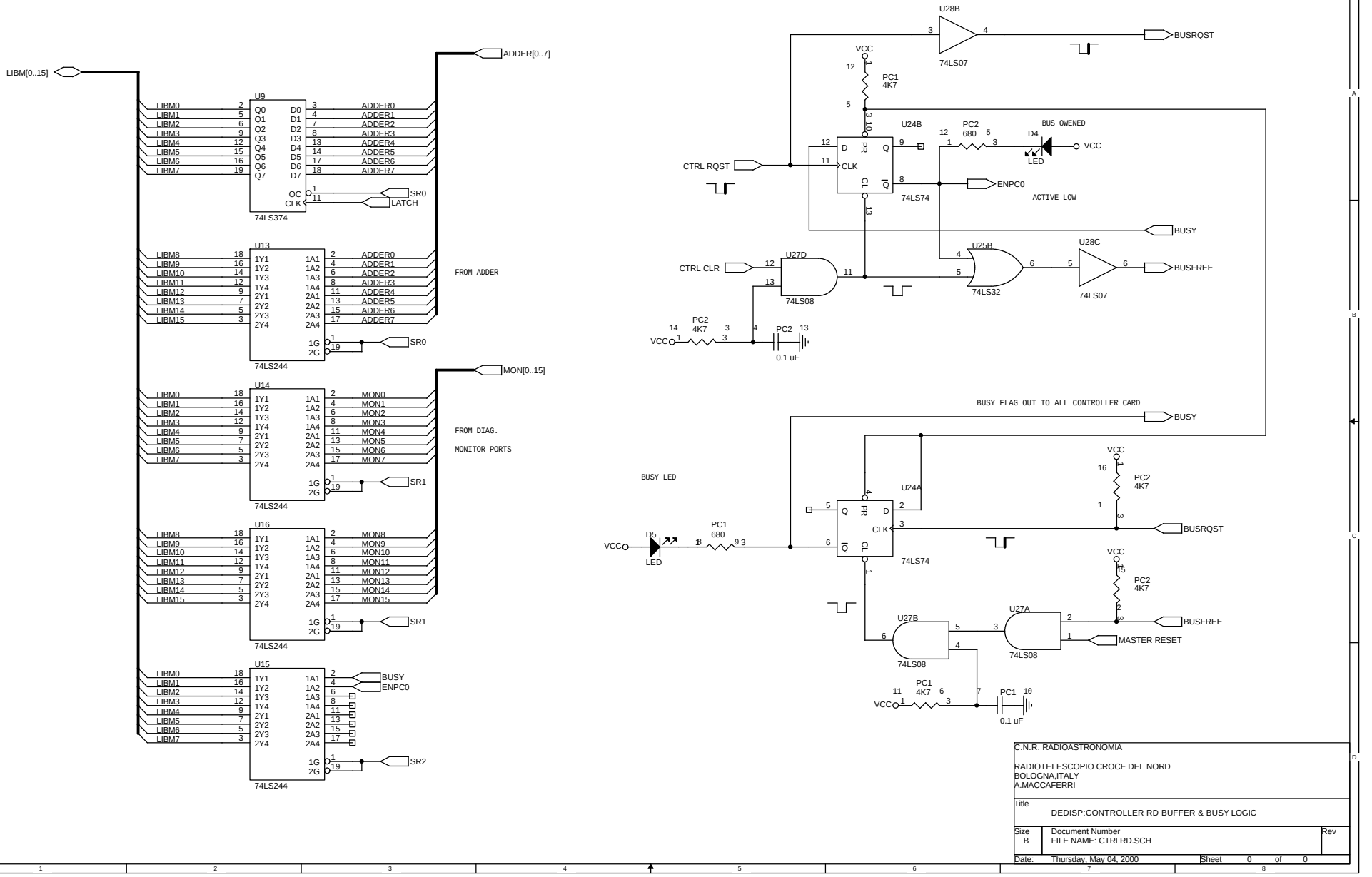
Il controller ha tre funzioni principali:
- interfacciamento verso un host (nel nostro caso il PC i386) per il controllo del dedispersore stesso e per la scrittura dei dati da dedisperdere e la lettura dei dati dedispersi attraverso il link principale.
- interfacciamento verso un secondo utilizzatore 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 postincrementa.

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
CONNECTOR DB37
P?
INTERFACE TO MASTER PC
IBM0
IBM1
IBM2
IBM3
IBM4
IBM5
IBM6
IBM7
IBM8
IBM9
IBM10
IBM11
IBM12
IBM13
IBM14
IBM15
AD1
AD2
AD3
WAIT
IODECODE
nIOW
nIOR
VCC
Pc1
4k7
U1
U2
74LS245
LIBM[0..15]
LOCAL DATA BUS
U20
U21
74LS138
SR[0..7]
READ STROBES
SW[0..7]
WRITE STROBES
U34F
U34E
74LS14
C.N.R. RADIOASTRONOMIA
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA,ITALY
A.MACCAFERRI
Title
DEDISPERSOR: CTRL BUS BUFFERING & DECODER
Size
A
Document Number
FILE NAME: CTRLBBUS.SCH
Rev
Date:
Thursday, May 04, 2000
Sheet
0
of
0


LIBM[0..15]
MEMI[0..15]
MEMORY INPUT
DATA BUS
MEM WR
U3
U4
74LS244
DEB[0..15]
K DEDISP. BUS
DED WR
ENPC0
U25A
74LS32
U5
U6
/CS BUS
ENABLE 8 CH. BLOCK
CS[0..15]
U7
U8
74LS374
CONTROL PORT
U10
MA[0..3]
SELECT DIAG.
MONITOR PORT
WSEL[0..2]
INIT_RD_WORD
SW3
74LS374
U11
INIT WCOUNT
COUNT CLR
CTRL RQST
CTRL CLEAR
MASTER RESET
MEM WR
DED WR
INIT DED COUNT
SW4
C[0..2]
U12
SW1
COUNT CLR
CTRLCK
ENPC0
LD
CLR
CK
PCOC[0..7]
U17
74LS154
SEL[0..15]
U19
U18
INIT DED COUNT
U35
U36
74LS191
SW1
PC0C[0..7]
SELECT CH.
COUNTER
C.N.R. RADIOASTRONOMIA
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA,ITALY
A.MACCAFERRI
Title
DEDISPERSOR: CONTROLLER WR. BUFF. & LATCH
Size
B
Document Number
FILE NAME: CTRLLATC.SCH
Rev
Date:
Thursday, May 04, 2000
Sheet
0
of
0



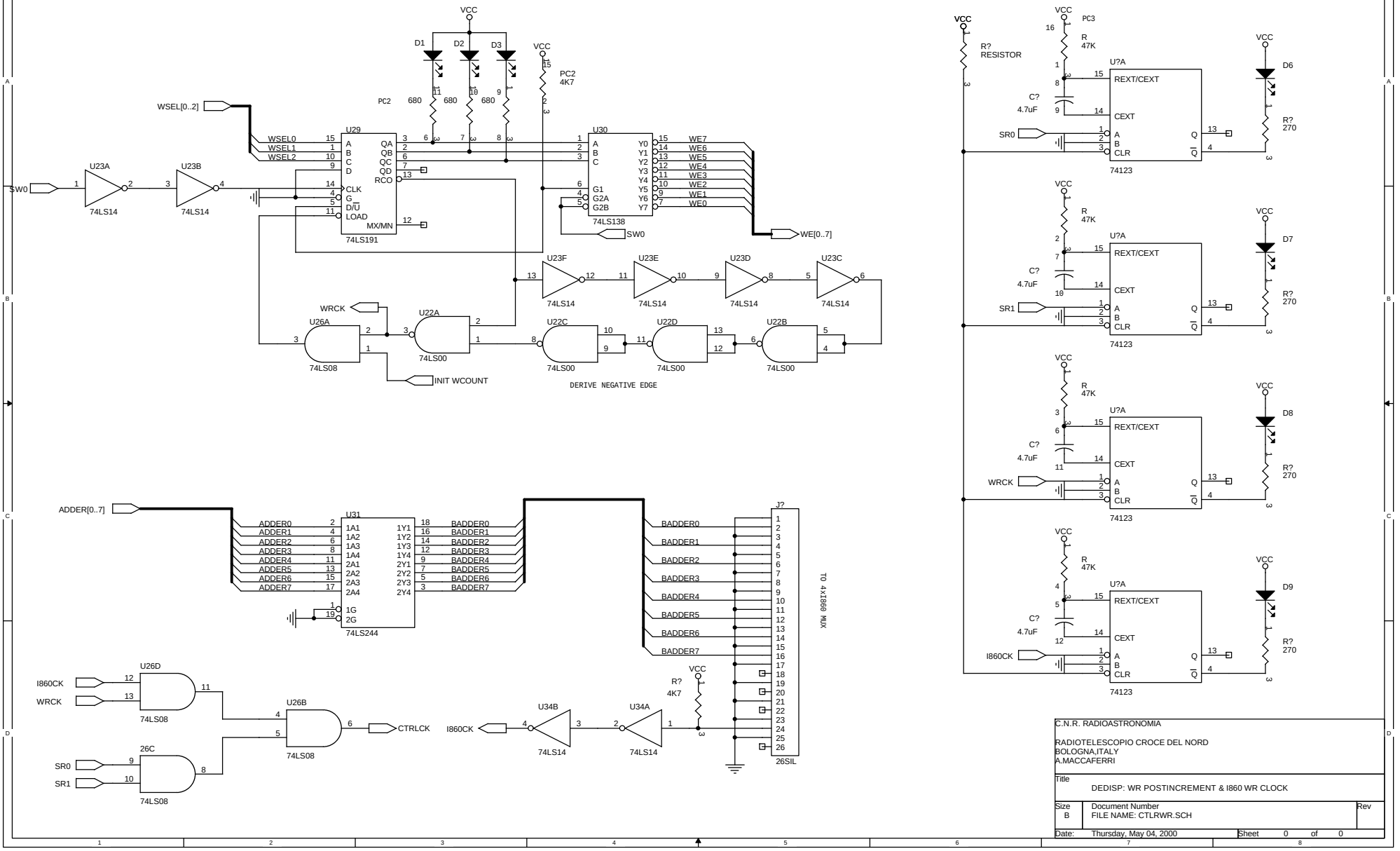
C.N.R. RADIOASTRONOMIA

RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA,ITALY
A.MACCAFERRI

Title: DEDISP:CONTROLLER RD BUFFER & BUSY LOGIC

Size	Document Number	Rev
B	FILE NAME: CTRLRD.SCH	

Date: Thursday, May 04, 2000 Sheet 0 of 0



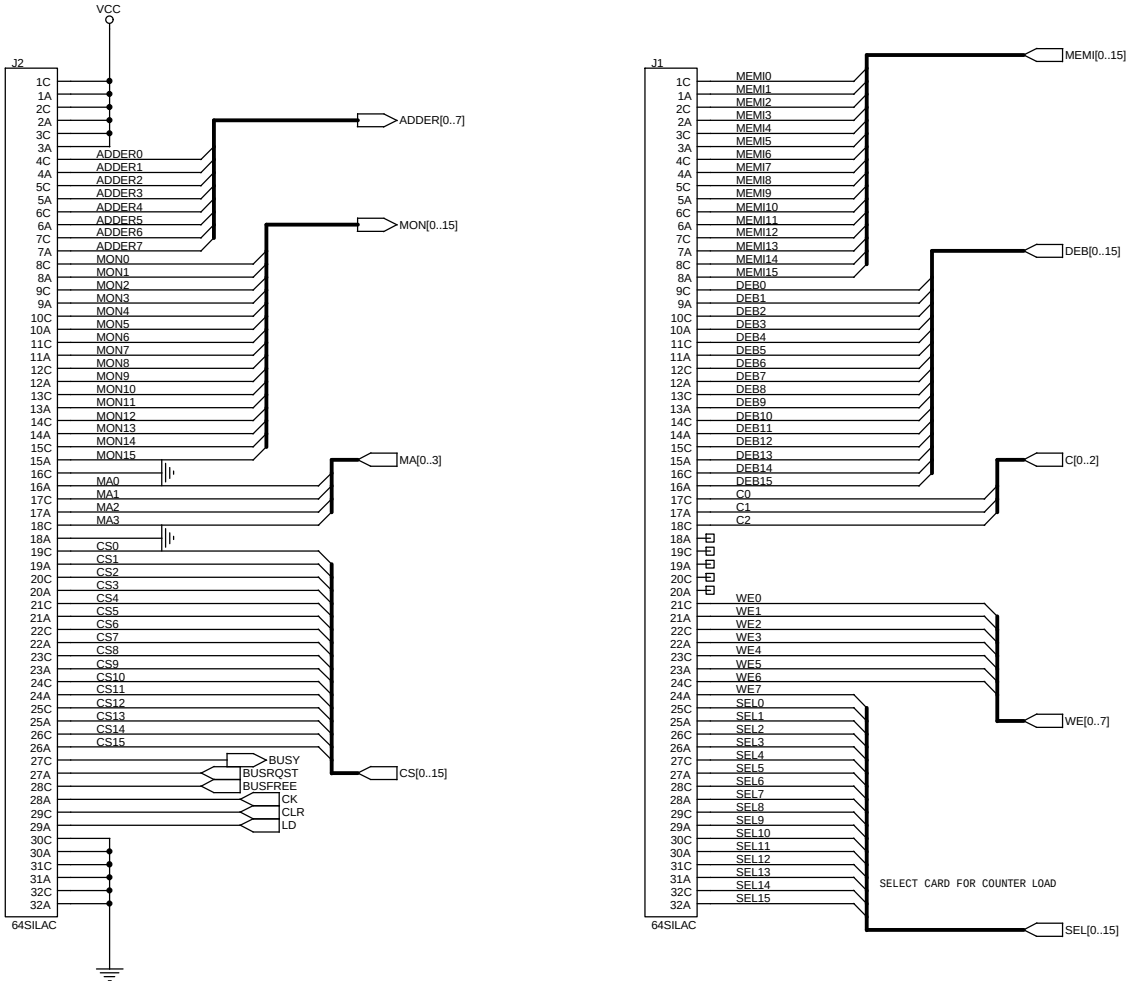
C.N.R. RADIOASTRONOMIA

RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA,ITALY
A.MACCAFERRI

Title: DEDISP: WR POSTINCREMENT & I860 WR CLOCK

Size	Document Number	Rev
B	FILE NAME: CTLRWR.SCH	

Date: Thursday, May 04, 2000 Sheet 0 of 0



C.N.R. RADIOASTRONOMIA

RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA,ITALY
A.MACCAFERRI

Title: DEDISP: MASTER CTRL. BACKPLANE CONNECTORS

Size	Document Number	Rev
B	FILE NAME: CTRLEURO.SCH	

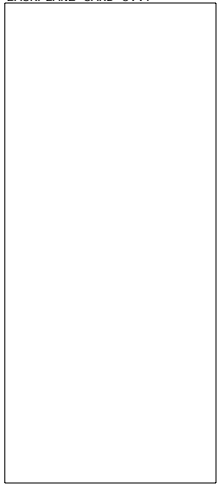
Date: Thursday, May 04, 2000 Sheet 0 of 0

DEDISPERSOR MASTER CONTROLLER LAYOUT
GND
DEDISPERSOR MASTER
CONTROL LAYOUT
+VCC
GND
+VCC
GND
P1
P2
J2
J1
PD1
PD2
PD3
PU1
PU2
PU3
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U13
U19
U14
U12
U15
U11
U17
U18
U16
U20
U21
FC1
PC2
U22
U23
U24
U25
U26
U27
U28
U29
U30
U31
U32
U33
PC3
U34
U35
U36

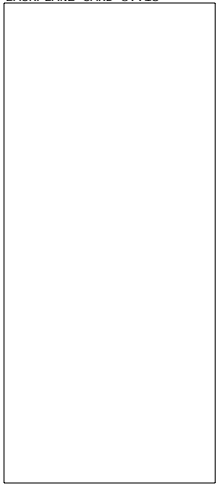
- Backplane

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.

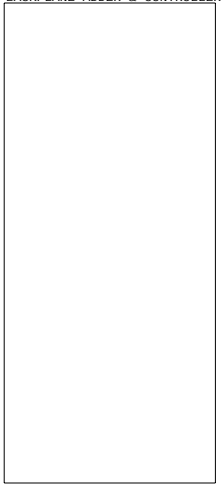
BACKPLANE CARD 0..7



BACKPLANE CARD 8..15



BACKPLANE ADDER & CONTROLLERS



C.N.R. RADIOASTRONOMIA

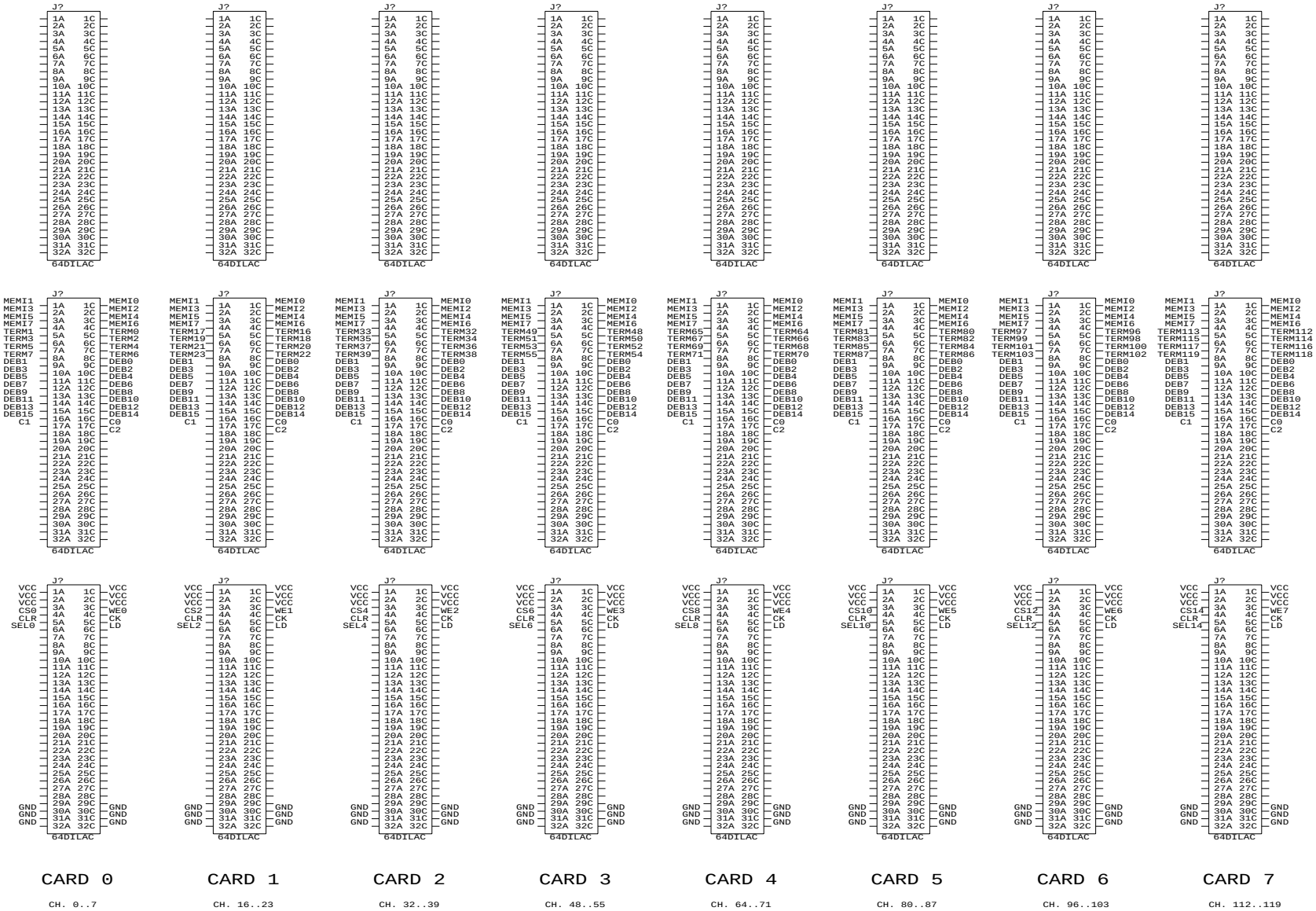
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
A.MACCAFERRI

Title

CHASSIS BACKPLANE WIRING DIAGRAM

Size	Document Number	REV
B	FILE NAME: BACKPLANE.SCH	

Date: August 27, 1992 Sheet of



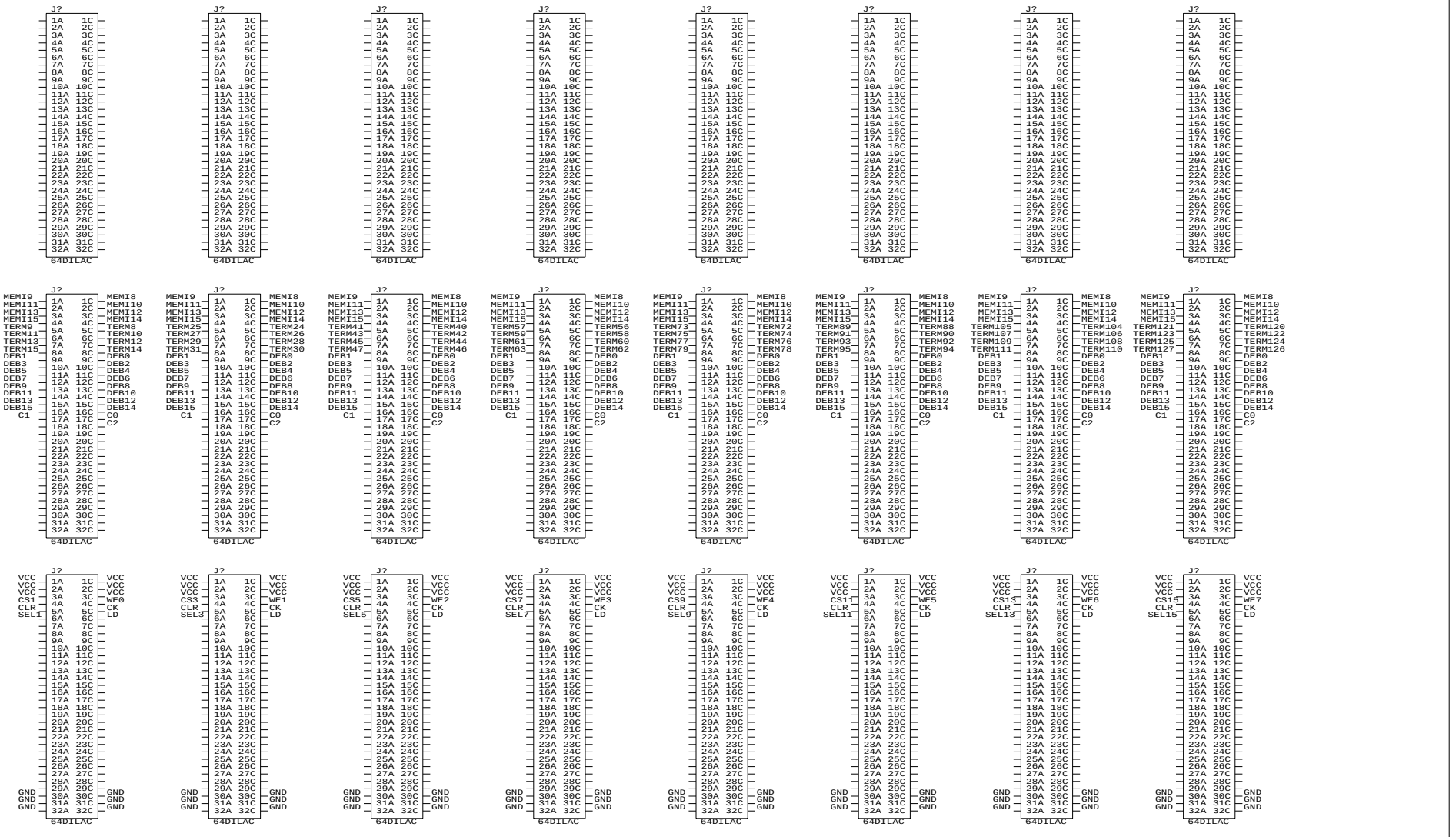
C.N.R. RADIOASTRONOMIA

RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
A.MACCAFERRI

Title: BACKPLANE SEZ.A WIRING ASSEMBLY

Size	Document Number	REV
C	FILE NAME: BACKPLANE.SCH	

Date: August 27, 1992 | Sheet of



VIEW FROM CARD SIDE

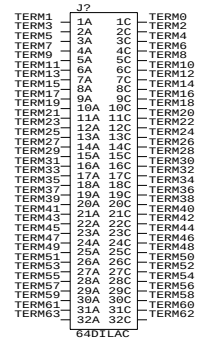
C.N.R. RADIOASTRONOMIA

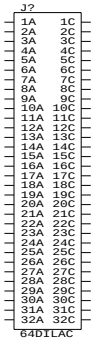
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
A.MACCAFERRI

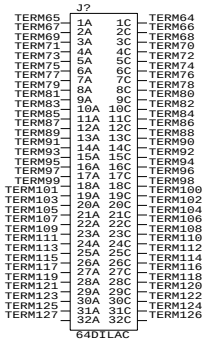
Title
BACKPLANE SEZ.B WIRING ASSEMBLY

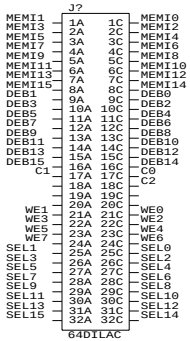
Size	Document Number	REV
C	FILE NAME: BACKPB.SCH	

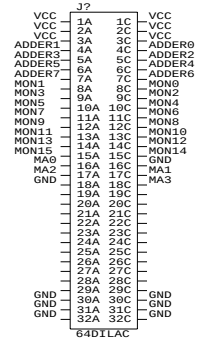
Date: August 27, 1992 Sheet of



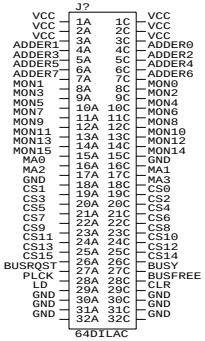








ADDER



PC MASTER
CONTROLLER
INTERFACE

VIEW FROM CARD SIDE

C.N.R. RADIOASTRONOMIA

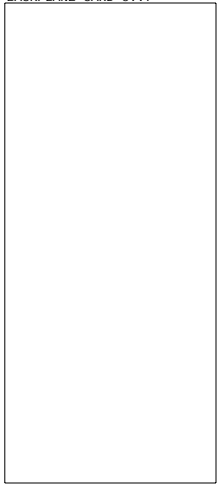
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
A.MACCAFERRI

Title
BACKPLANE SEZ.C WIRING ASSEMBLY

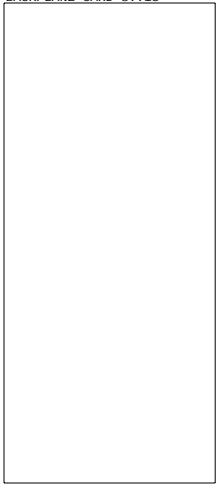
Size	Document Number	REV
C	BACKPC.SCH	

Date: August 3, 1993 Sheet of

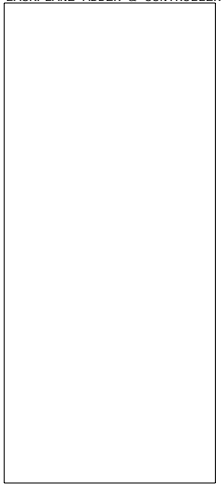
BACKPLANE CARD 0..7



BACKPLANE CARD 8..15

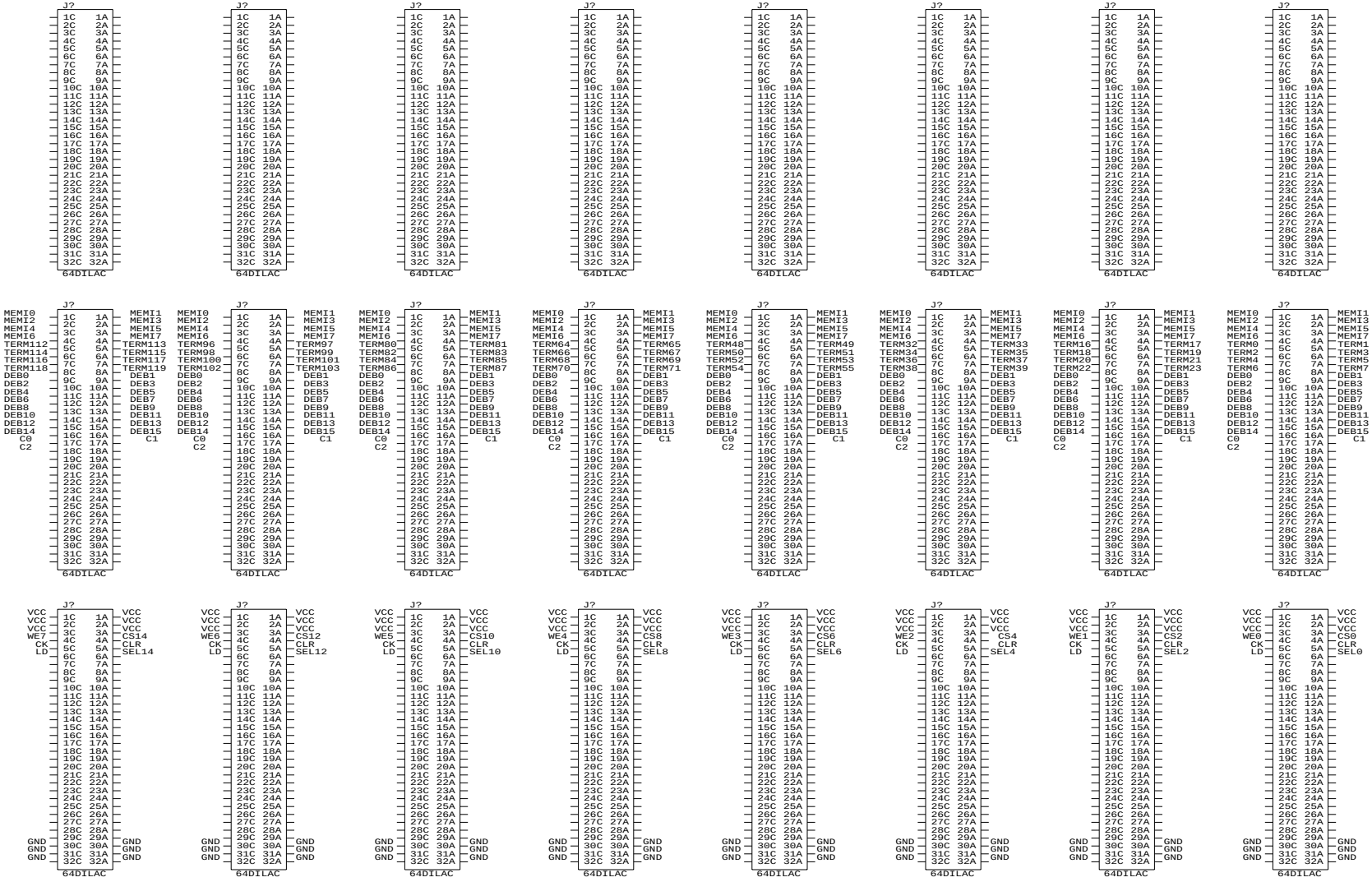


BACKPLANE ADDER & CONTROLLERS



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

Size	Document Number	REV
B		
Date: August 27, 1992	Sheet of	



CARD 7

CH. 112..119

CARD 6

CH. 96..103

CARD 5

CH. 80..87

CARD 4

CH. 64..71

CARD 3

CH. 48..55

CARD 2

CH. 32..39

CARD 1

CH. 16..23

CARD 0

CH. 0..7

VIEW FROM REAR SIDE

C.N.R. RADIOASTRONOMIA

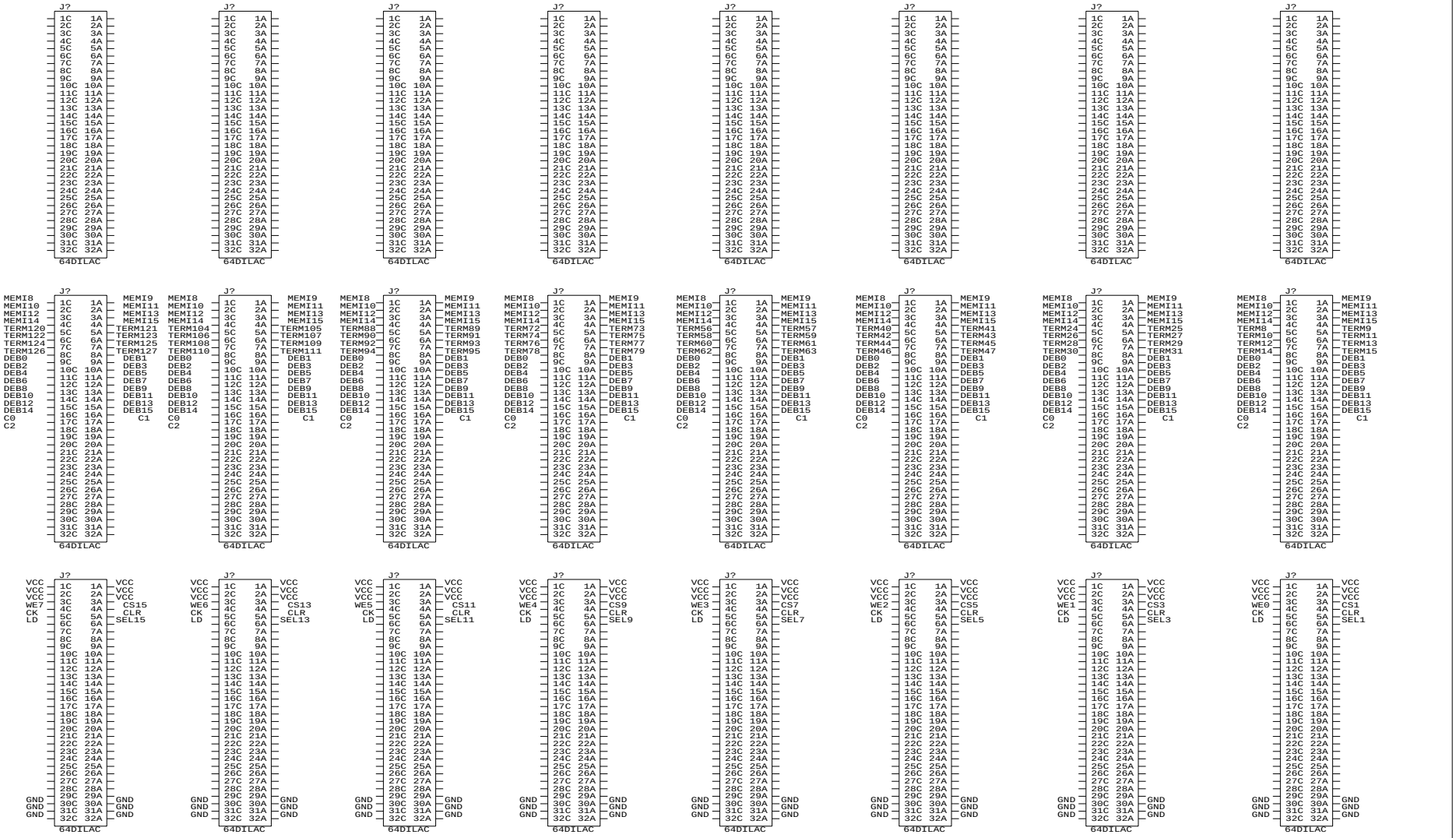
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
A.MACCAFERRI

Title

BACKPLANE SEZ.A REAR VIEW

Size	Document Number	REV
C	FILE NAME BACKPAR.SCH	

Date: August 27, 1992 | Sheet of



CARD 15

CH. 120..127

CARD 14

CH. 104..111

CARD 13

CH. 88..95

CARD 12

CH. 72..79

CARD 11

CH. 57..63

CARD 10

CH. 40..47

CARD 9

CH. 24..31

CARD 8

CH. 8..15

VIEW FROM REAR SIDE

C.N.R. RADIOASTRONOMIA

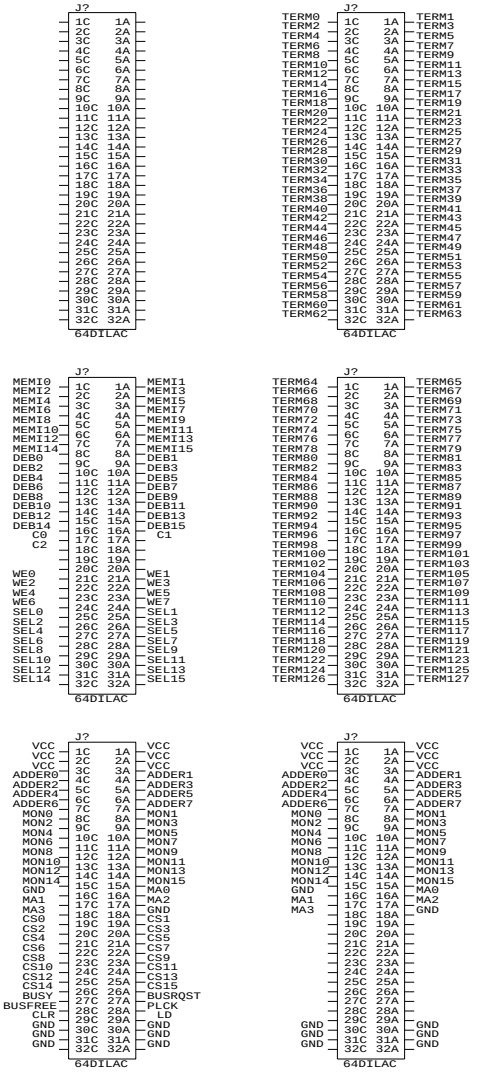
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
A.MACCAFERRI

Title

BACKPLANE SEZ.B REAR VIEW

Size	Document Number	REV
C	FILE NAME BACKPBR.SCH	

Date: August 27, 1992 | Sheet of



PC MASTER CONTROLLER INTERFACE

ADDER

VIEW FROM REAR SIDE

C.N.R. RADIOASTRONOMIA

RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
A.MACCAFERRI

Title: BACKPLANE SEZ.C REAR VIEW

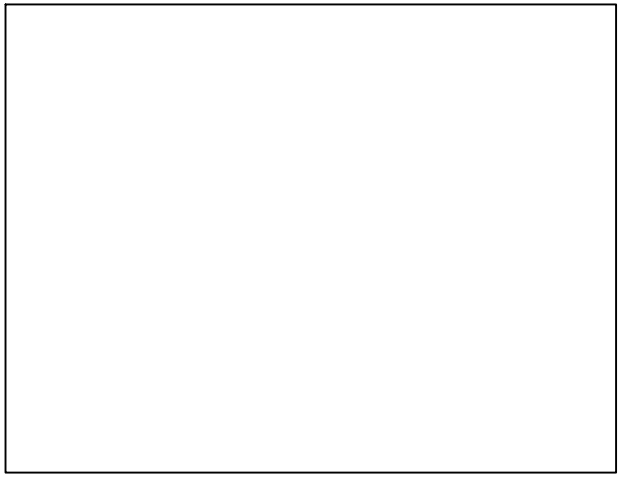
Size: C | Document Number: FILE NAME BACKPCR.SCH | REV:

Date: August 3, 1993 | Sheet: of

- La scheda Dedisp-Interface

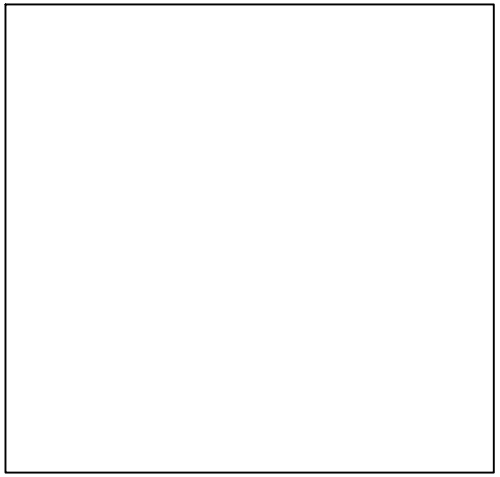
Questa scheda posta nel bus ISA del Personal Computer i386 host, collegandosi con la scheda Controller del dedispersore tramite un flat cable, permette ad un programma che opera sul PC di controllare tutto il funzionamento del dedispersore. In pratica i registri del dedispersore vengono visti come se 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



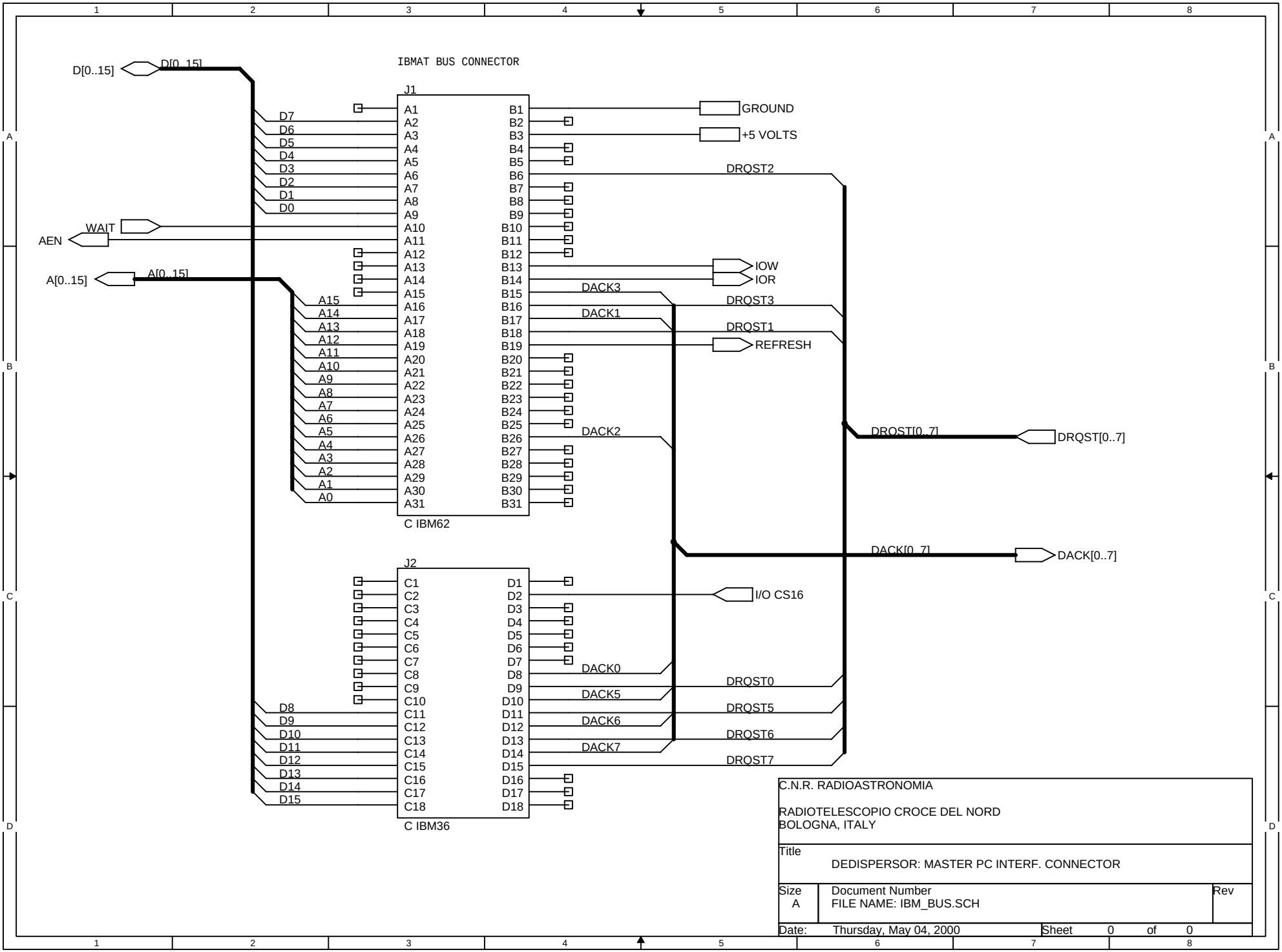
IBM_BUS.SCH

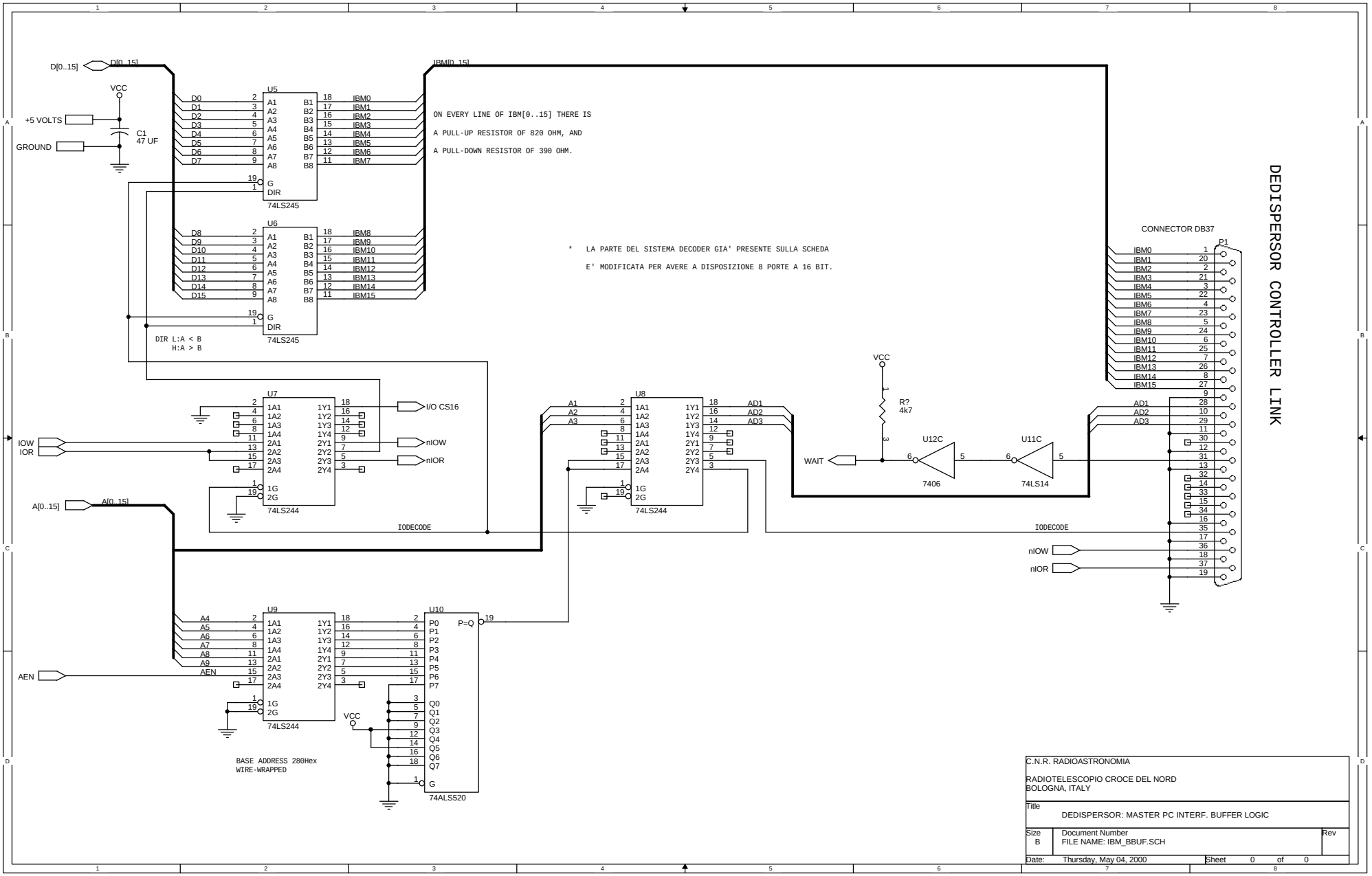
IBM INTERFACE BUFFERING



IBM_BBUF.SCH

C.N.R. RADIOASTRONOMIA RADIOTELESCOPIO CROCE DEL NORD BOLOGNA, ITALY		
Title: DEDISPERSOR: MASTER PC INTERFACE MAIN BLOCK		
Size: A	Document Number FILE NAME: PC0.SCH	Rev
Date: Thursday, May 04, 2000	Sheet 0 of 0	


IBMAT BUS CONNECTOR
D[0..15]
D[0..15]
J1
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
A20
A21
A22
A23
A24
A25
A26
A27
A28
A29
A30
A31
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
B25
B26
B27
B28
B29
B30
B31
D7
D6
D5
D4
D3
D2
D1
D0
WAIT
AEN
A[0..15]
A[0..15]
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
C IBM62
GROUND
+5 VOLTS
IOW
IOR
REFRESH
DRQST2
DRQST3
DRQST1
DACK3
DACK1
DACK2
DRQST[0..7]
DRQST[0..7]
DACK[0..7]
DACK[0..7]
J2
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
C16
C17
C18
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
D16
D17
D18
D8
D9
D10
D11
D12
D13
D14
D15
C IBM36
I/O CS16
DACK0
DACK5
DACK6
DACK7
DRQST0
DRQST5
DRQST6
DRQST7
C.N.R. RADIOASTRONOMIA
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
Title
DEDISPERSOR: MASTER PC INTERF. CONNECTOR
Size
A
Document Number
FILE NAME: IBM_BUS.SCH
Rev
Date:
Thursday, May 04, 2000
Sheet
0
of
0


DEDISPERSOR CONTROLLER LINK
D[0..15]
D[0..15]
VCC
+5 VOLTS
GROUND
C1
47 UF
U5
U6
74LS245
IBM[0..15]
ON EVERY LINE OF IBM[0..15] THERE IS
A PULL-UP RESISTOR OF 820 OHM, AND
A PULL-DOWN RESISTOR OF 390 OHM.
* LA PARTE DEL SISTEMA DECODER GIA' PRESENTE SULLA SCHEDA
E' MODIFICATA PER AVERE A DISPOSIZIONE 8 PORTE A 16 BIT.
CONNECTOR DB37
P1
DIR L:A < B
H:A > B
U7
U8
74LS244
I/O CS16
nIOW
nIOR
IOW
IOR
A[0..15]
A[0..15]
IODECODE
WAIT
R?
4k7
U12C
U11C
7406
74LS14
U9
U10
74ALS520
AEN
P=Q
BASE ADDRESS 280Hex
WIRE-WRAPPED
C.N.R. RADIOASTRONOMIA
RADIOTELESCOPIO CROCE DEL NORD
BOLOGNA, ITALY
Title
DEDISPERSOR: MASTER PC INTERF. BUFFER LOGIC
Size
B
Document Number
FILE NAME: IBM_BBUF.SCH
Rev
Date:
Thursday, May 04, 2000
Sheet
0
of
0

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),addervec(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
      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=adder+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,(lenght-maxoffset-1-8000),8000
       write(*,*)' block,longtest,longloop ',k,longtest,longloop
       address=640
       nwords=8000
       call inp$b(buffer,nwords,address)
       do nloop=0,7999
        if ((k+nloop).gt.lenght)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...',longtest,longloop
       enddo
c      fine loop longtest  (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]

        cld
 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.Maccaferri, 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.Maccaferri
IRA 170/92

3) **"Scheda di interfaccia per acquisizione dati ad alta velocitá su bus ISA"**
A.Maccaferri, 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.Maccaferri, S.Montebugnoli, M.Roma, G.Tomassetti.
IRA 137/90