

# Elettronica di Acquisizione di GIANO

---

## Interfaccia digitale

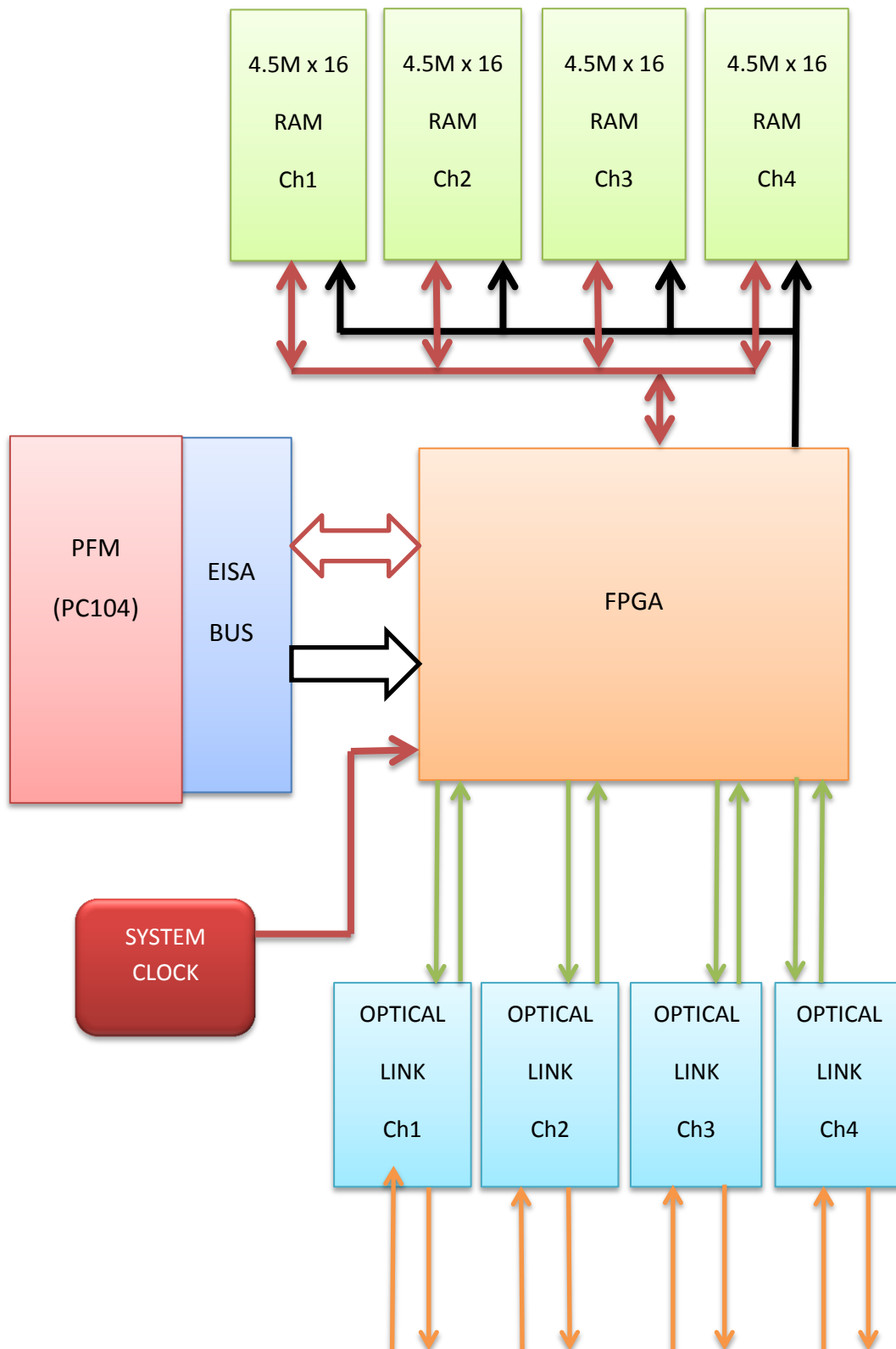
**Valdemaro Biliotti, Carlo Baffa, Elisabetta Giani**

**27/07/2012**

## Sommario

*Giano è uno spettrometro infrarosso ad alta risoluzione costruito attorno al sensore Hawaii2 PACE sensibile allo spettro compreso fra 0.95 e 2.5 $\mu$ m. Il sensore è formato da un mosaico di quattro quadranti con segnali di controllo, masse e alimentazioni, indipendenti tra loro. L'elettronica di acquisizione si adatta a questa caratteristica, perché composta da quattro canali di acquisizione indipendenti, ognuno dei quali è dedicato alla gestione di un quadrante del sensore. Il disegno che ne deriva è perciò innovativo rispetto ai sistemi di acquisizione costruiti per gli strumenti precedenti. L'uso di componenti di ultima generazione e un nuovo disegno per l'amplificazione del segnale a temperature criogeniche hanno portato a ottimi risultati per quanto riguarda il rumore di lettura, la stabilità a lungo termine e il cross-talk fra i canali, che risulta inferiore al misurabile. I componenti programmabili FPGA hanno permesso di implementare funzioni complesse che si traducono in elevata flessibilità in condizioni operative, ed elevata capacità di auto diagnosi, dalla quale deriva una ottima manutenibilità nel tempo. Il presente rapporto descrive la parte digitale di interfaccia.*

Figura1 : Schema a blocchi



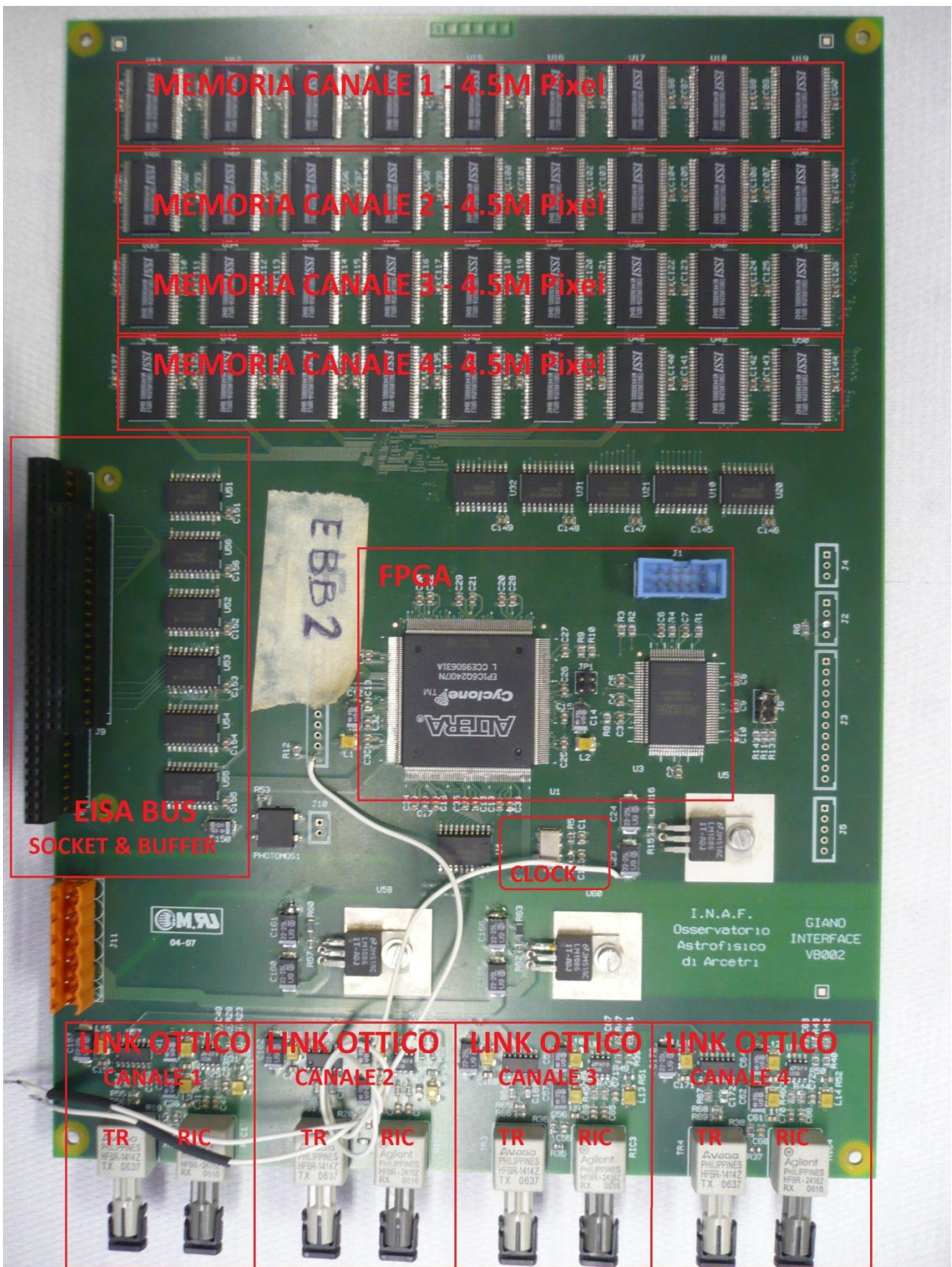
La figura 1 rappresenta lo schema a blocchi dell'interfaccia digitale di Giano. Essa è formata dal modulo PFM, un processore che comunica verso l'esterno con un collegamento Ethernet e dalla scheda Interface Board.

Questi due circuiti sono racchiusi nello stesso contenitore, che è alloggiato nel rack insieme alle parti elettroniche di controllo motori e temperature. Il PFM è un modulo standard, un single board computer conforme alle specifiche del consorzio PC104, che in dimensioni estremamente contenute incorpora anche i driver per il monitor, la tastiera, le interfacce seriali e Ethernet, ha inoltre la possibilità di leggere da un disco a stato solido. Oltre a queste interfacce di input output, il PFM è dotato di interfaccia di una interfaccia parallela PC104, che emula il bus EISA, e che permette lo scambio dati da e verso la scheda Interface Board. Questa scheda è un'interfaccia digitale fra il processore e le schede Analog Board, nelle quali risiede gran parte dei componenti elettronici dei canali di acquisizione di Giano. In quanto interfaccia, la scheda decodifica i comandi e i dati ricevuti dal Bus EISA e li trasmette verso le schede Analog Board, in modo broadcast o individuale; in direzione opposta raccoglie i dati ottenuti dal campionamento dei pixel, li organizza in memoria e li trasferisce al PC104. L'accesso alla memoria per la lettura dei dati avviene attraverso il bus EISA, con scambio dati a 16 bit in modo asincrono rispetto alle temporizzazioni di scansione e acquisizione.

Oltre a queste funzioni di scambio dati, la Interface Board fornisce una temporizzazione per tutti i canali di acquisizione, al fine di garantire una perfetta sincronizzazione tra loro. La parte digitale e di conversione dei canali di acquisizione è implementata sulle quattro schede Analog Board, che sono completamente indipendenti tra di loro, hanno perciò bisogno di un riferimento unico rispetto al quale sincronizzare il proprio clock interno, per generare i segnali di temporizzazione e scansione dei quadranti perfettamente all'unisono.

Questa prestazione è ottenuta a partire dal clock della Interface Board, che diventa clock di riferimento per il sistema di acquisizione, e dalle caratteristiche del protocollo usato per trasmettere i dati sul link a fibra ottica. Tale protocollo è stato sviluppato per permettere la trasmissione simultanea di dati e di un segnale di clock, senza ricorrere a dispositivi costosi. La velocità del link è 16 megabit al secondo, non è elevata se si considera lo stato dell'arte attuale, ma adeguata per le caratteristiche del sensore HAWAII2. La presenza continua del clock ha permesso anche di implementare la verifica in tempo reale dei link, di rivelare eventuali errori e misurare continuamente il Bit Error Rate, che risulta estremamente basso.

Figura 2: foto della scheda



La memoria è suddivisa in quattro parti, ognuna è dedicata ad un canale di acquisizione: in pratica ogni banco di memoria memorizza i dati ricevuti da una scheda Analog Board, che gestisce il quadrante del sensore ne acquisisce il segnale. La memoria è dimensionata in modo tale da permettere la memorizzazione fino a quattro immagini per ogni quadrante, è implementata con RAM statiche veloci che permettono una gestione di tipo FIFO. La lettura e scrittura avviene in tempi molto rapidi su tutti e quattro i banchi di memoria, ragione per cui le temporizzazioni di lettura dal bus possono essere asincrone rispetto alle temporizzazioni di scrittura dei dati.

Il cardine della scheda Interface Board è il componente programmabile FPGA (Field Programmable Gate Array), il quale svolge tutte le funzioni logiche per il protocollo di scambio dati a fibra ottica, per la gestione delle memorie e per lo scambio dati con il PC104 attraverso il bus EISA. Si tratta di un componente che dispone di circa 6000 elementi logici configurabili, 2 PLL (Phase Locked Loop) per la gestione avanzata di clock ad alta frequenza, e una memoria interna configurabile di circa 92Kbit. La sua configurazione, ovvero il suo programma, risiede in una memoria interna di tipo RAM, che perde il suo contenuto quando viene tolta l'alimentazione. Per questo motivo il componente FPGA si autoprogramma ogni volta che viene alimentato: lo fa andando a leggere i dati di configurazione che sono memorizzati in modo non volatile su un dispositivo basato su memoria di tipo FLASH. La coppia componente programmabile e componente di configurazione è connessa in modo da formare una catena JTAG, che è accessibile attraverso un connettore dedicato. Attraverso l'interfaccia JTAG e un dispositivo specifico è possibile scaricare il programma su entrambi i dispositivi, inoltre con le utility a corredo del software di sviluppo è possibile effettuare tutta una serie di test, compreso il debug e l'analisi degli stati logici interni del componente. Le caratteristiche dei componenti programmabili di questa generazione sono molto avanzate, le potenzialità d'impiego e la flessibilità che ne derivano sono veramente elevate: sono plasmabili a piacimento per quanto riguarda le funzioni e lo standard di input output dei segnali. Inoltre, il componente di configurazione può contenere fino a 8 programmi selezionabili, dunque basta spegnere il circuito, cambiare un ponticello, e alimentare nuovamente la scheda per vedere funzionare il medesimo circuito, lo stesso hardware, con una programmazione diversa.

## Lo schema elettrico:

Lo schema si compone di 5 disegni organizzati in modo gerarchico, al livello più alto c'è lo schema elettrico GIANO INTERFACE (Figura 3), con il componente programmabile FPGA (EP1C6; U1A, U1B, U1C e U1D), il componente per la configurazione (EPC4; U3) e la programmazione, il connettore per l'interfaccia JTAG (J1) e l'oscillatore al quarzo, che dà la base dei tempi a 16.384Mhz per tutto il sistema (U2; VCS25AXT). I blocchi di colore verde non sono componenti elettronici, ma rappresentano gli schemi a livello gerarchico più basso. Il componente EP1C6 ha 240 pin di connessione, troppi per stare in un unico simbolo grafico, quindi per avere una certa leggibilità, nello schema è stato suddiviso in 4 sezioni (A,B,C,D). Il suo programma viene scritto dentro il componente di configurazione (EPC4) per mezzo del connettore JTAG. Negli istanti seguenti l'accensione, il componente programmabile inizia una procedura con la quale legge i dati di configurazione da un file interno al componente EPC4. La selezione del file di configurazione dipende dallo stato dei ponticelli su J6, cambiando la disposizione di questi è possibile cambiare la programmazione

Ciò che nello schema a blocchi è indicato con EISA BUS, nello schema elettrico è rappresentato dal disegno chiamato EISA INTERFACE (Figura 4): si tratta di connettori secondo lo standard ISA (J8 e J9) e di componenti (74LVC245; U51, U52, U53, U54 e U55) che costituiscono un buffer tra il componente FPGA e i connettori. La Interface Board e il processore si accendono con lo stesso interruttore sull'alimentazione della scatola che li contiene, tuttavia la scheda non abilita i buffer del bus EISA per primi 16 secondi, al fine di non interferire con il bootstrap del PC; questo tempo di attesa si ripresenta anche nel caso di reset della scheda (pulsante RESET FPGA sul frontale). La connessione tra la scheda ed il pulsante di RESET FPGA viene fatta per mezzo dei cavi bianchi che si vedono sulla foto di Figura 2.

La parte di memoria, ovvero tutte e quattro le memorie dello schema a blocchi, sono inserite dentro lo schema elettrico denominato FIFO MEMORY (Figura 5). I 36 componenti IS61LV51216, che sono memorie RAM ad accesso rapido (12nSec) organizzate in 524288 locazioni da 16 bit ciascuna, sono disposte su quattro righe a formare una matrice 9x4 (Figura 2). In ogni riga troviamo i 9 componenti dedicati alla memorizzazione dei dati campionati di un quadrante, complessivamente la capacità di memoria è tale da

contenere quattro immagini. Le memorie U11, U12, U13, U14, U15, U16, U17, U18 e U19 costituiscono la FIFO che memorizza i dati acquisiti dal canale 1, le memorie che vanno da U22 a U30 memorizzano i dati acquisiti dal canale 2, mentre le memorie da U33 a U41 memorizzano i dati del canale 3, e infine le memorie da U42 a U50 memorizzano i dati del canale 4. E' stata prevista una procedura di test delle memorie: consiste nella scrittura del componente FPGA di dati in memoria secondo una sequenza ben definita, il PFM li legge e verifica se tale sequenza è corretta. I componenti 74LVC245 sono buffer per i dati (U10 e U20), per gli indirizzi (U21, U31 e U32) e per i bit di selezione delle memorie.

Per quanto riguarda i Link a fibre ottiche c'è differenza fra la rappresentazione dello schema a blocchi, che mette in un unico blocco sia il trasmettitore che il ricevitore di un canale, e la rappresentazione schematica dove i trasmettitori e ricevitori sono raggruppati per genere in due schemi elettrici. Nel disegno OPTICAL LINK RECEIVER (Figura 6) troviamo 4 ricevitori a fibre ottiche (HFBR-2416) accoppiati con 4 comparatori veloci (LT1016). I 4 ricevitori sono indicato come RIC1, RIC2, RIC3 e RIC4, in perfetto accordo con il canale di acquisizione: al canale 1 corrisponde RIC1, al canale 2 corrisponde RIC2 e così fino al canale 4.

Sul frontale della scatola che contiene la Interface Board il ricevitore del canale 1 è il primo iniziando da sinistra, mentre quello del canale 4 è l'ultimo a destra.

Nello schema OPTICAL LINK TRANSMITTER (Figura 7), oltre alla presenza dei regolatori di tensione che forniscono l'alimentazione dei componenti, ci sono 4 laser led (HFBR-1414) etichettati come TR1, TR2, TR3 e TR4 connessi a delle porte con elevato fan out (74ACT00) etichettate come U57, U59, U61 e U62. Anche per i trasmettitori vale la stessa convenzione: TR1 trasmette i dati per il canale 1, TR2 per il canale 2 e così via fino al canale 4. La disposizione sul frontale segue quella descritta per i ricevitori, TR1 è il primo iniziando da sinistra, TR2 il secondo, mentre TR4 è l'ultimo sulla destra.





Figura 4 : SCHEMA EISA INTERFACE

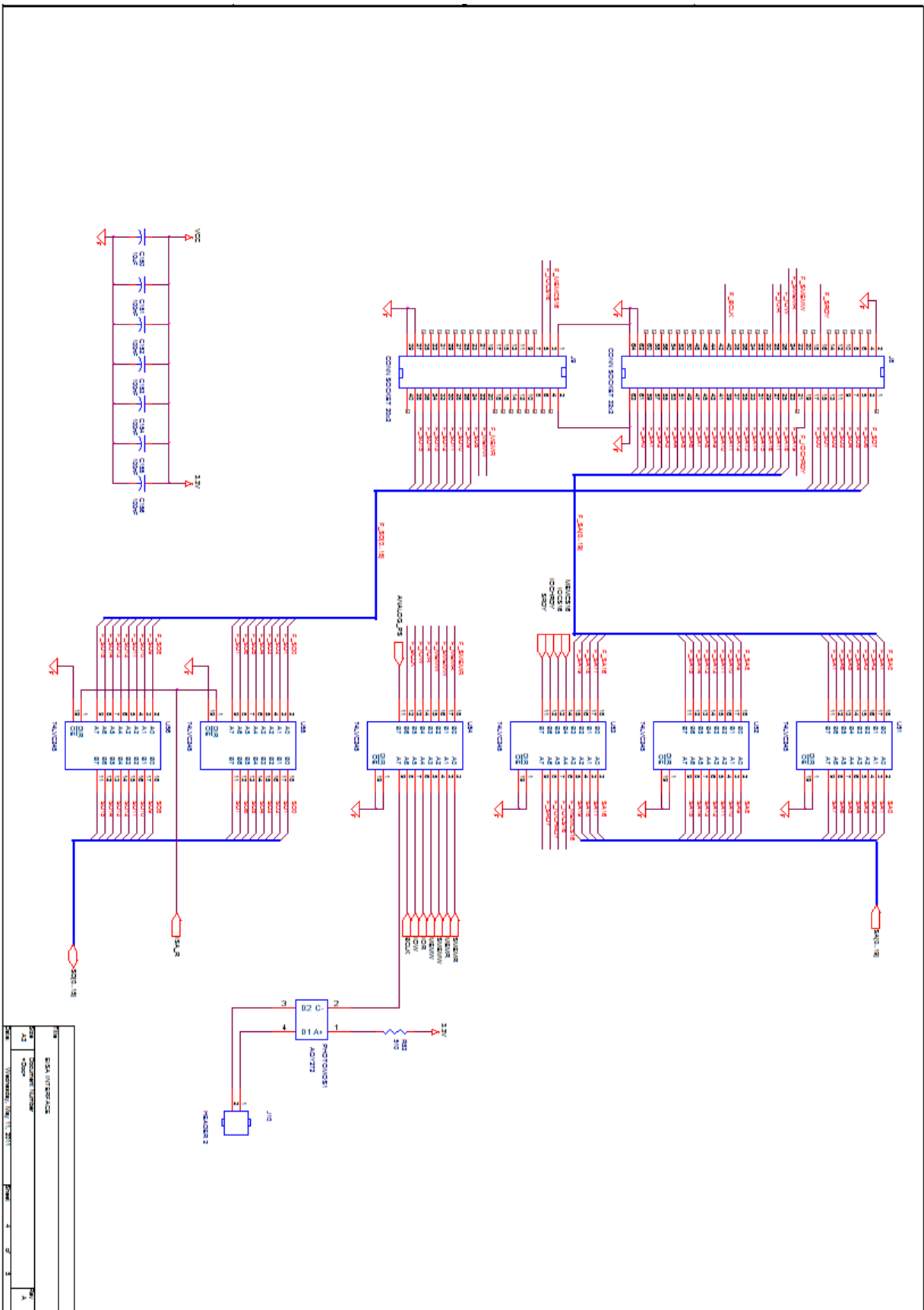


Figura 5 : SCHEMA FIFO MEMORY

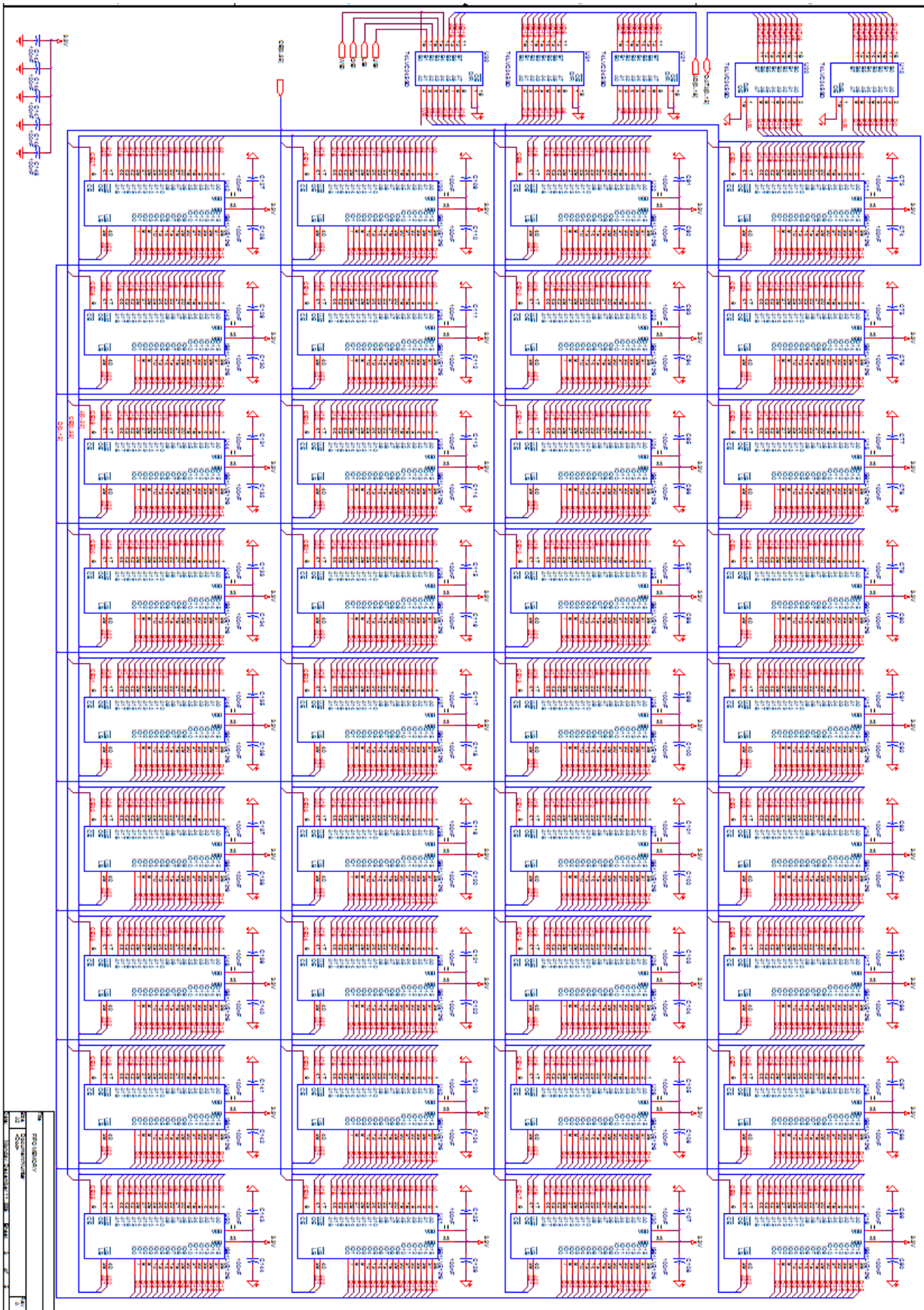


Figura 6: SCHEMA OPTICAL LINK RECEIVER

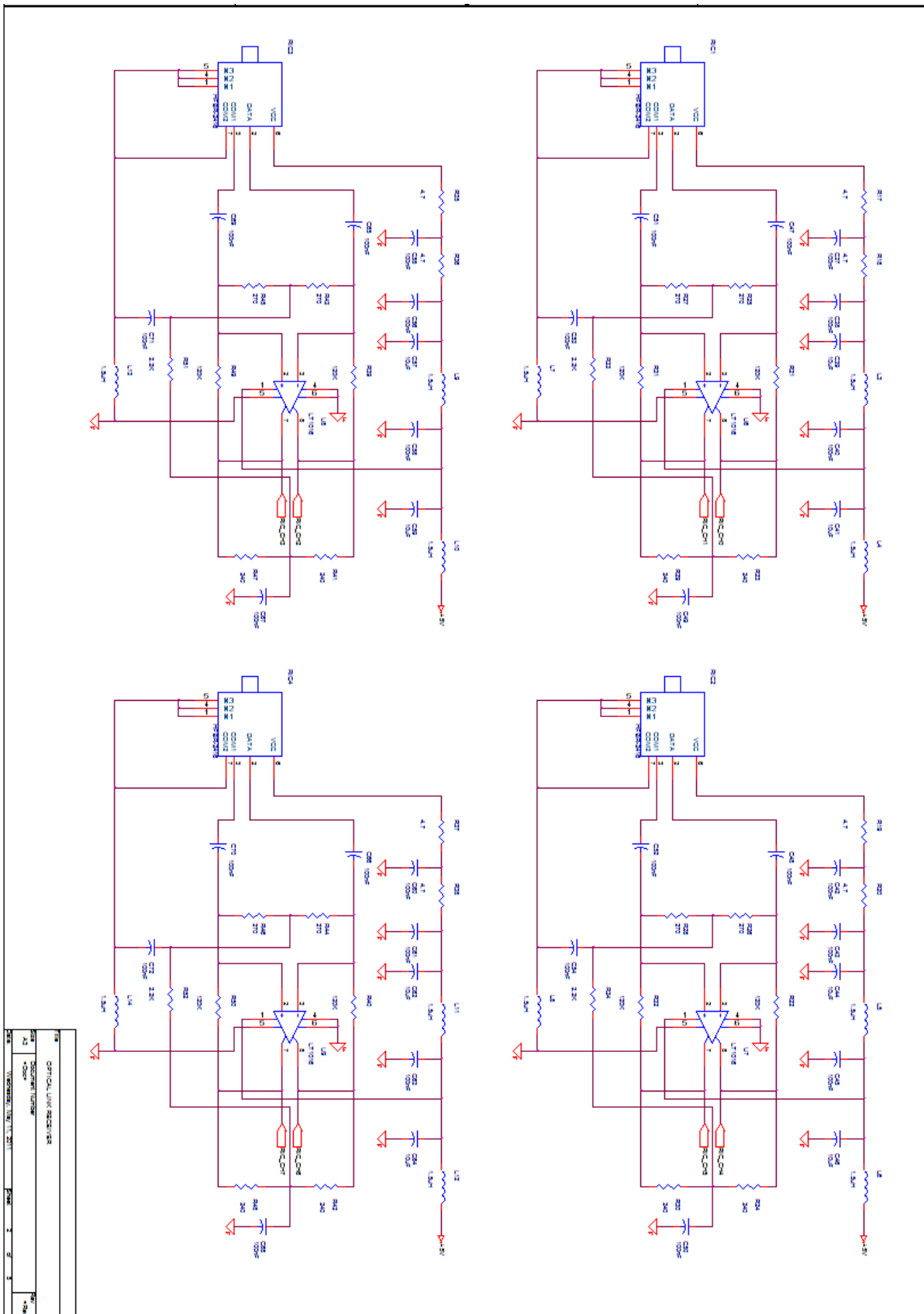


Figura 7: SCHEMA OPTICAL LINK TRANSMITTER

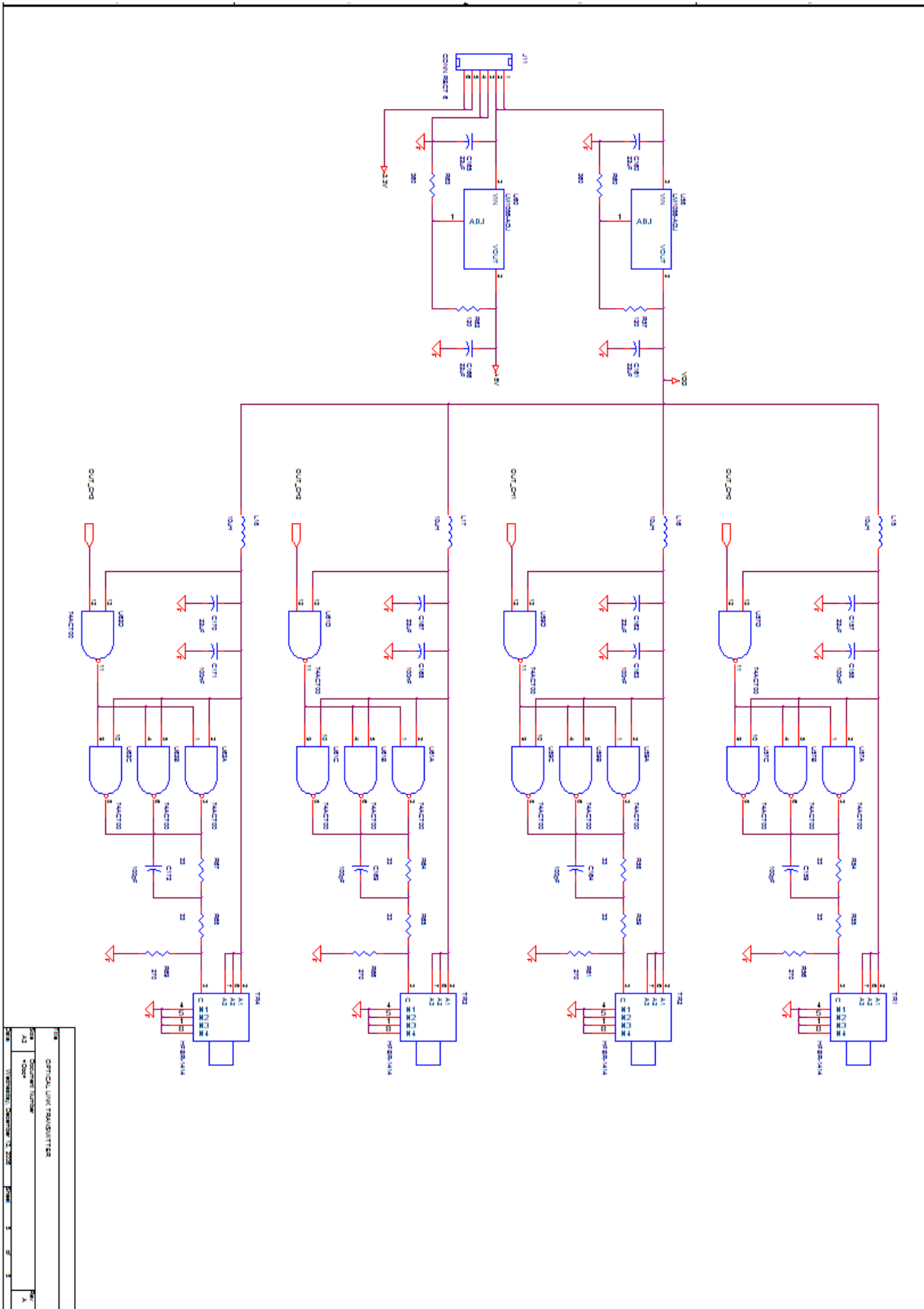
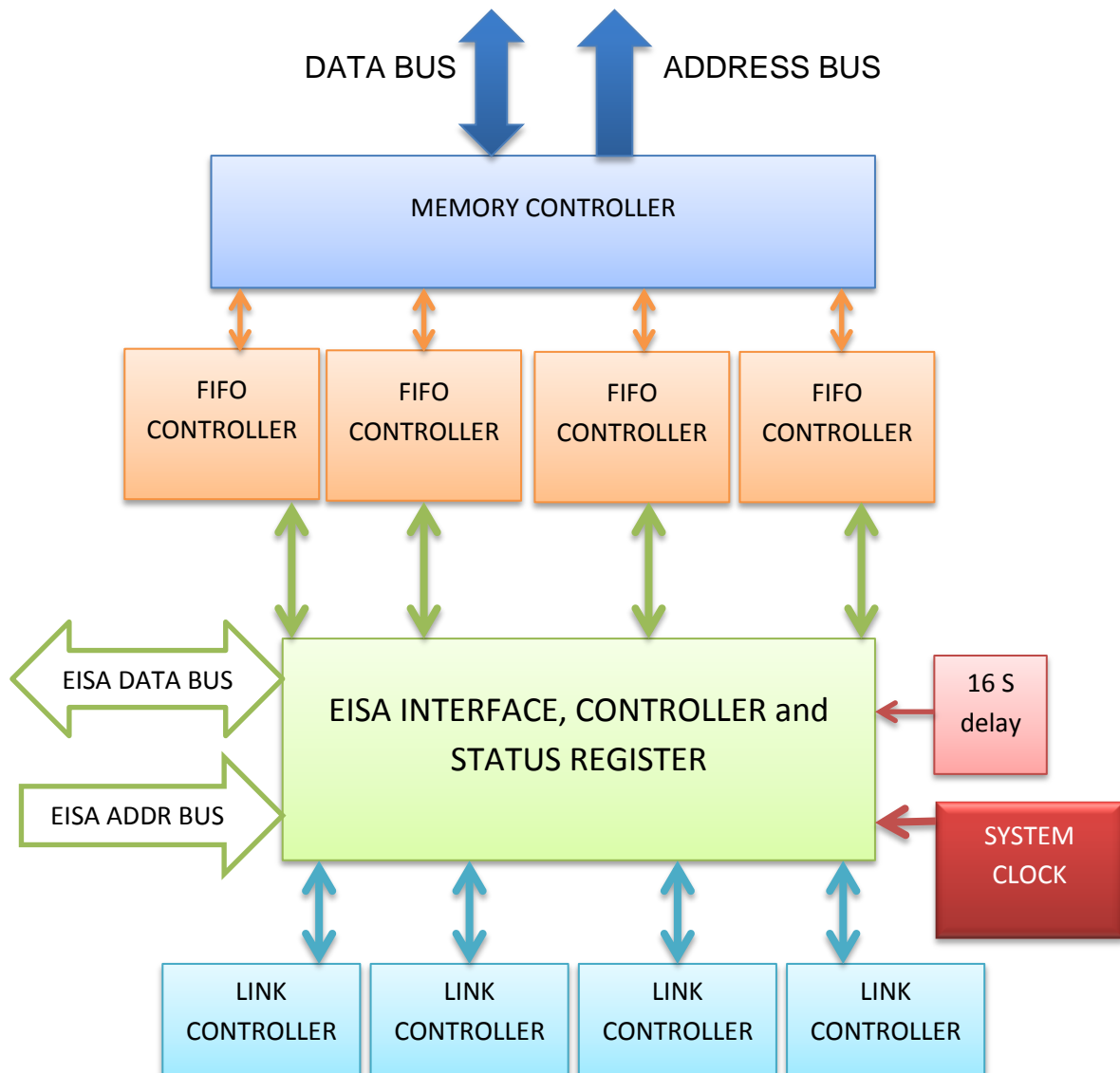


Tabella 1: Lista dei componenti principali e relativa documentazione.

SIGLA COMPONENTE	DOCUMENTAZIONE
PHOTOMOS AQY272	<a href="http://docs-europe.electrocomponents.com/webdocs/0cce/0900766b80cce32a.pdf">http://docs-europe.electrocomponents.com/webdocs/0cce/0900766b80cce32a.pdf</a>
HFBR-2416	<a href="http://docs-europe.electrocomponents.com/webdocs/04e6/0900766b804e66d6.pdf">http://docs-europe.electrocomponents.com/webdocs/04e6/0900766b804e66d6.pdf</a>
HFBR-1414	<a href="http://docs-europe.electrocomponents.com/webdocs/04e6/0900766b804e66d6.pdf">http://docs-europe.electrocomponents.com/webdocs/04e6/0900766b804e66d6.pdf</a>
VCS25AXT-163	<a href="http://docs-europe.electrocomponents.com/webdocs/0360/0900766b8036077d.pdf">http://docs-europe.electrocomponents.com/webdocs/0360/0900766b8036077d.pdf</a>
74LVC245	<a href="http://docs-europe.electrocomponents.com/webdocs/008f/0900766b8008f61c.pdf">http://docs-europe.electrocomponents.com/webdocs/008f/0900766b8008f61c.pdf</a>
IS61LV51216	<a href="http://docs-europe.electrocomponents.com/webdocs/0a59/0900766b80a598b6.pdf">http://docs-europe.electrocomponents.com/webdocs/0a59/0900766b80a598b6.pdf</a>
EP1C6Q240I7N	<a href="http://docs-europe.electrocomponents.com/webdocs/077b/0900766b8077b79d.pdf">http://docs-europe.electrocomponents.com/webdocs/077b/0900766b8077b79d.pdf</a>
EPC4PC100	<a href="http://docs-europe.electrocomponents.com/webdocs/077a/0900766b8077aa7f.pdf">http://docs-europe.electrocomponents.com/webdocs/077a/0900766b8077aa7f.pdf</a>
LT1016CS8	<a href="http://docs-europe.electrocomponents.com/webdocs/078f/0900766b8078f495.pdf">http://docs-europe.electrocomponents.com/webdocs/078f/0900766b8078f495.pdf</a>
PFM (PC104)	<a href="http://aaeon.industrialpartner.com/embedded-single-board-computers/pc104-cpu-module/pfm-620s-b10.htm">http://aaeon.industrialpartner.com/embedded-single-board-computers/pc104-cpu-module/pfm-620s-b10.htm</a>  <a href="http://www.delvingware.com/webservices/medialibrary/media/datasheets/pfm_620s_-1418625386l.pdf">http://www.delvingware.com/webservices/medialibrary/media/datasheets/pfm_620s_-1418625386l.pdf</a>

## Dentro il componente programmabile FPGA

Figura 8: Schema a blocchi

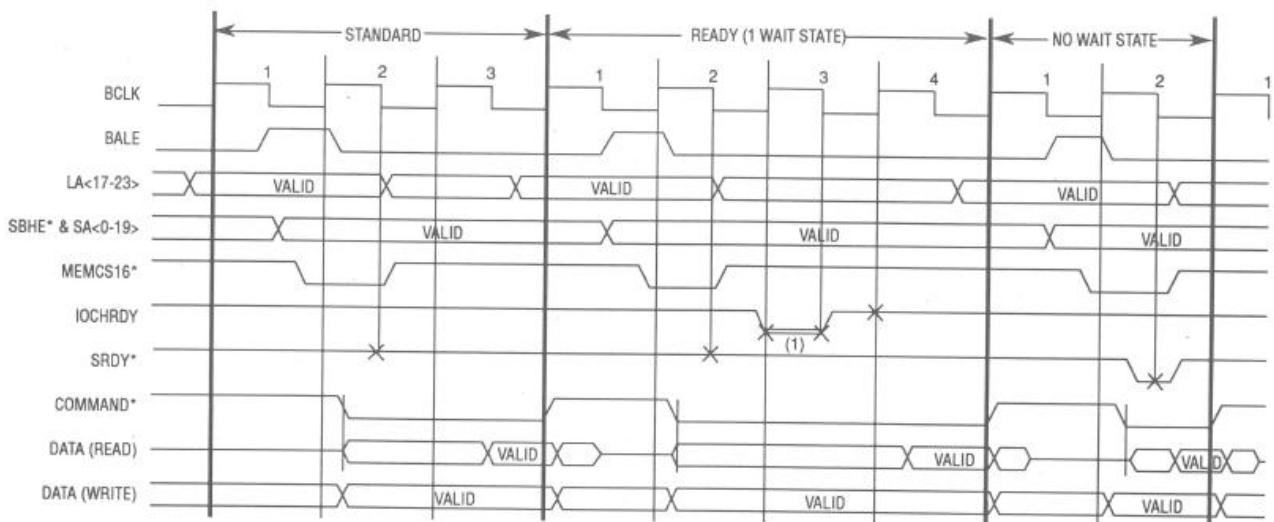


Gli schemi elettrici della scheda mostrano che la maggior parte dei componenti sono memorie, buffer o dispositivi di comunicazione per le fibre ottiche, nessuno di questi è in grado di espletare funzioni logiche e di controllo. Tutte le funzioni richiedono una certa elaborazione logica sono implementate dentro il componente programmabile FPGA. La realizzazione di tali funzioni richiede un software specifico, nel caso di Giano il sistema di sviluppo usato è QUARTUS, un software fornito dallo stesso produttore (ALTERA) dei componenti programmabili. Oltre agli strumenti di design entry, nel sistema di sviluppo c'è la simulazione e tante altre utility per lo sviluppo dei componenti FPGA. In particolare QUARTUS accetta la rappresentazione

di tipo schematico gerarchico, in cui i blocchi possono essere componenti e funzioni logiche standard parametrizzabili tipo contatori, porte logiche, mux eccetera. Oppure possono rappresentare le funzioni implementate con linguaggi descrittivi dello hardware, tipo Verilog o VHDL. Per realizzare le macchine a stati logici e le funzioni non standard dell'elettronica di acquisizione di Giano è stato usato il linguaggio VHDL. La figura 8 mostra una rappresentazione sinottica delle funzioni logiche presenti all'interno del dispositivo.

Iniziamo la descrizione dalla parte che gestisce lo scambio dati attraverso il bus EISA<sup>1</sup>; le sue specifiche prevedono che il bus dati possa lavorare a 8 oppure a 16 bit, nel caso di Giano il bus lavora sempre a 16 bit. Oltre al bus dati ci sono 20 bit di indirizzo da SA 0 a SA19, dei segnali di controllo per la scrittura (MEMWR e IOWR), dei segnali di controllo per la lettura (MEMRD e IORD), dei segnali che servono per stabilire la dimensione del bus e gli eventuali stati di attesa (MEMCS16, IOCS16, IOCHRD e SRDY), più un segnale di riferimento BCLK a 8.33MHz. Ogni operazione del bus avviene secondo lo schema di principio riportato in figura.

Figura 9: forme d'onda con riferimento a BCLK.



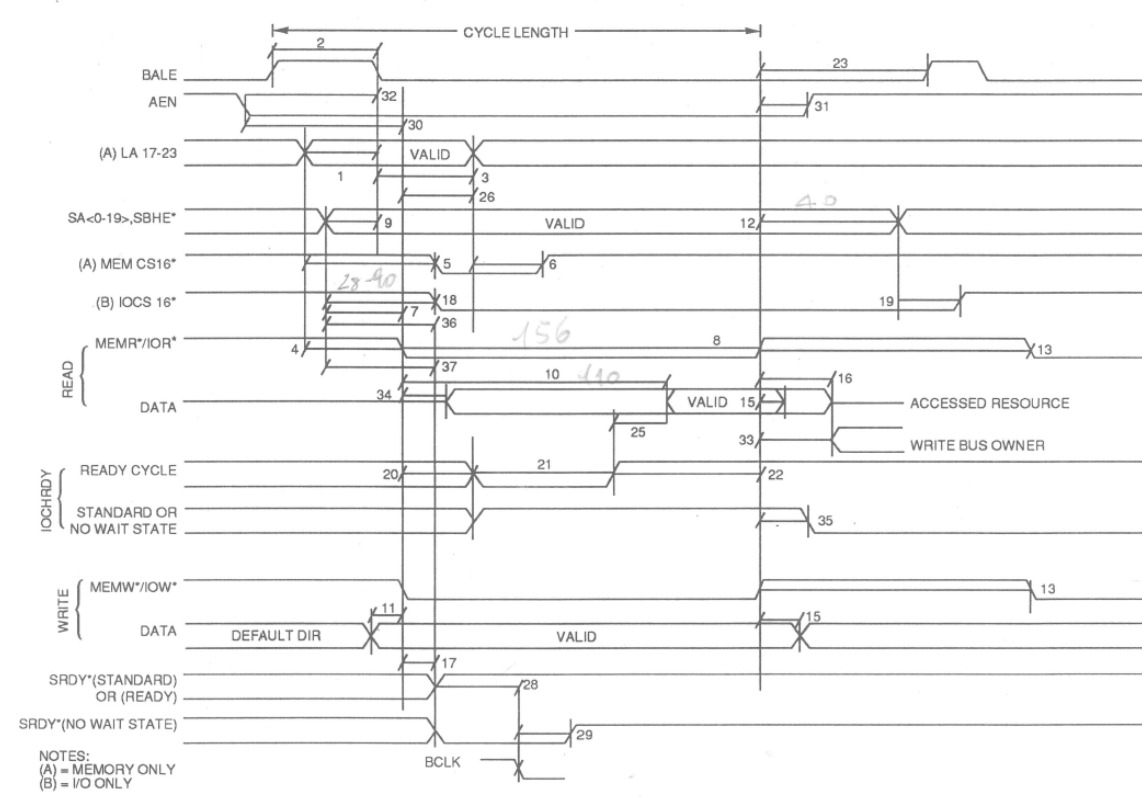
È importante notare che ogni ciclo dura, di default, 3 passi di clock BCLK, che può essere aumentato uno o più passi se il segnale IOCHRDY è basso, oppure può diventare di due passi se è basso il segnale SRDY. In ogni caso la durata minima del ciclo è due passi a 8.33MHz, ovvero l'intervallo di tempo teorico compreso fra due operazioni lettura-scrittura è

<sup>1</sup> AT BUS DESIGN, EDWARD SOLARI, ANNABOOKS



almeno 240 nano secondi, mentre la durata standard è pari a 360 nano secondi. Ogni operazione ha inizio quando gli indirizzi da SA0 a SA19 sono stabili ad un valore valido, dopo un piccolo intervallo di tempo il processore asserisce il comando (indicato con COMMAND\*) al livello '0', che può essere di scrittura, se il segnale MEMWR (oppure IOWR) diventa basso, oppure di lettura se il segnale MEMRD (oppure IORD) diventa basso. In funzione del tipo di operazione richiesta, la periferica legge il dato presente sul bus, oppure pone il proprio dato sul bus medesimo. Trascorso un certo intervallo di tempo, il comando (di lettura o di scrittura) torna a 1 e l'operazione finisce. La durata del ciclo è parte integrante dello standard EISA, ma può essere allungata o resa più veloce, in base alle esigenze e alla velocità della periferica; in dettaglio il ciclo può essere ridotto a due passi se la periferica pone a 0 il segnale SRDY, oppure può essere aumentato a piacimento, cioè per tutto il tempo in cui la periferica tiene a 0 il segnale IOCHRDY. Se entro un certo intervallo di tempo dall'inizio di un ciclo di lettura o scrittura, la periferica risponde ponendo a 0 il segnale MEMCS16, il bus assume che il dato sul bus sia a 16 bit, altrimenti lo scambio dati (default) avviene a 8 bit. In questa rappresentazione il PFM è il master del bus EISA e la Interface Board è la periferica.

Figura 10: forme d'onda senza il riferimento a BCLK:



Questa è una sintetica descrizione delle specifiche del bus EISA, premessa necessaria per capire come il componente programmabile gestisce l'interfacciamento a questo bus.

Prima di tutto va ricordato che per un intervallo di tempo pari a circa 16 secondi dall'accensione, o da quando viene premuto il pulsante RESET FPGA, la scheda rimane estranea al bus, nel senso che tutti i buffer sono tenuti in alta impedenza, a prescindere dai dati e dalle operazioni fatte dal processore. Questo ritardo è stato implementato al fine di evitare che la scheda potesse interferire con il processore e le sue operazioni durante il bootstrap all'accensione. Esaurito questo transitorio iniziale, la scheda campiona i segnali con un suo clock interno a 32.6MHz, ottenuto moltiplicando per 2 la frequenza del clock fornito dall'oscillatore esterno; poi, in base al tipo di comando e all'indirizzo presente sul bus, esegue diversi tipi di operazioni. Queste operazioni possono essere indirizzate alla scheda stessa, e in tal caso non ci sono cicli di attesa per il bus, oppure possono essere operazioni di lettura o scrittura che riguardano le schede Analog Board. Questo è il caso in cui il componente programmabile impone al bus un tempo di attesa tale da consentire la trasmissione dei dati sul link a fibra ottica. La durata dipende dal tipo di operazione: in caso di scrittura verso una Analog Board l'attesa supera il microsecondo, cioè il tempo necessario per spedire indirizzo e dato in forma seriale sulla fibra ottica. Se invece si tratta di una lettura il tempo di attesa potrebbe diventare doppio in particolari e rare circostanze. In generale, si tratta di spedire un indirizzo in forma seriale e ricevere il dato sempre in forma seriale, perciò il tempo di attesa equivale a quello della semplice scrittura; ma potrebbe anche verificarsi il caso in cui una lettura di un registro di stato del componente programmabile su una scheda Analog Board avvenga mentre questa sta già inviando i dati ottenuti dal campionamento dei pixel. Questo potenziale conflitto viene risolto attribuendo la priorità più alta alla trasmissione che è iniziata per prima, perché interromperla ne comporterebbe la perdita definitiva. La ovvia conseguenza è che: prima viene completata la trasmissione del pixel e poi viene eseguita quella del dato, è questo il caso nel quale il bus viene fatto attendere di più. Calcolare quanti cicli di attesa mettere, è compito del componente programmabile, che decide adattandosi alle circostanze e al tipo di operazione; al massimo l'attesa può arrivare al timeout di 6 micro secondi, fatto che va oltre le ipotesi previste, perché si può verificare solo e soltanto in caso di mal funzionamento del link a fibra ottica. La velocità del bus viene

rallentata solo per scrivere e leggere dati e parametri che riguardano le Analog Board, si tratta di dati per i quali la velocità di aggiornamento non è importante. Quando invece la velocità del bus è una prestazione cruciale, come per esempio nel caso di lettura delle immagini campionate, il trasferimento avviene senza stati di attesa. Di fatto, le immagini campionate stanno nella memoria della Interface Board proprio per questa ragione.

Riassumendo, il bus EISA può funzionare sia a 8 che a 16 bit, la scelta è demandata alla Interface Board dove il componente FPGA, che ne controlla il funzionamento, determina cicli di input output a 16 bit ponendo a '0' il segnale MEMCS16. Di conseguenza, non esiste la possibilità di indirizzare il byte, il bit meno significativo degli indirizzi perde di significato, è sempre a '0' durante i cicli di lettura o di scrittura. Per questa ragione, nelle mappe seguenti, l'indirizzo composto dai 20 bit che vanno da SA0 e SA19 è sempre pari, mentre la sua rappresentazione è esadecimale (Hex).

Il Bus PC104 "vede" la Interface Board e tutta l'elettronica di acquisizione come se fosse una zona di memoria, in termini tecnici la vede come "memory mapped", cioè ai registri e ai comandi di tutto il sistema elettronico sono assegnati degli indirizzi specifici, come se fossero delle locazioni di memoria. Per quanto riguarda registri e locazioni di memoria sono supportate sia operazioni di scrittura che di lettura, i comandi invece sono operazioni di sola scrittura ad una certa locazione, si tratta di una scrittura virtuale, nella quale nella maggior parte dei casi conta solo l'indirizzo e non il dato (che viene trascurato), unica eccezione è il comando RESET FIFO, per il quale viene data una dettagliata descrizione più avanti nel presente documento.

Tabella 2: mappa indirizzi del sistema di acquisizione

<i>INDIRIZZO (Hex)</i>	<i>Scheda indirizzata</i>
Da D0000 a D0FFF	Interface Board
Da D8000 a D8FFF	Analog Board - Canale 1
Da D9000 a D9FFF	Analog Board - Canale 2
Da DA000 a DAFFF	Analog Board - Canale 3
Da DB000 a DBFFF	Analog Board - Canale 4
Da DC000 a DCFFF	Comando o scrittura Broadcast

La zona di memoria compresa tra DC000 e DCFFF, è definita come broadcast, perché la scrittura ad una certa locazione di memoria compresa in quell'intervallo determina la scrittura simultanea, o l'esecuzione di un certo comando, su tutte e quattro le schede Analog Board contemporaneamente. Ovviamente può essere broadcast, cioè simultanea, solo la scrittura di un dato o un comando, la lettura broadcast non è prevista, in quanto si verrebbe a determinare un conflitto tra i canali di acquisizione.

Come è stato accennato, operazione di lettura o scrittura riferite alla Interface Board, come lo scambio dati dei pixel memorizzati nella FIFO, avvengono alla massima velocità consentita dallo standard del bus EISA; mentre le operazioni che riguardano le Analog Board hanno un certo quantitativo di passi di attesa per accomodare il ritardo dovuto alla trasmissione nel link. Tipicamente il tempo di attesa è circa 2 $\mu$ Sec, che possono superare 3 $\mu$ Sec se la lettura di un dato deve aspettare il trasferimento del pixel. In caso di errore del link a fibre ottiche, che si potrebbe verificare perché non ha trasmesso correttamente, o ha trasmesso ma non ha ricevuto correttamente il dato, il bus viene comunque rilasciato dopo un time out di 6 microsecondi.

Tabella 3: Mappa indirizzi di scrittura e comandi.

<i>INDIRIZZO</i>	<i>MNEMONICO</i>	<i>DESCRIZIONE</i>
D0020	KEEP ALIVE CANALE 1	SOLO IN FASE DI TEST
D0022	KEEP ALIVE CANALE 2	SOLO IN FASE DI TEST
D0024	KEEP ALIVE CANALE 3	SOLO IN FASE DI TEST
D0026	KEEP ALIVE CANALE 4	SOLO IN FASE DI TEST
D0028	KEEP ALIVE BROADCAST	SOLO IN FASE DI TEST
D0060	TEST FIFO CANALE 1	INIZIO TEST FIFO DEL CANALE 1
D0062	TEST FIFO CANALE 2	INIZIO TEST FIFO DEL CANALE 2
D0064	TEST FIFO CANALE 3	INIZIO TEST FIFO DEL CANALE 3
D0066	TEST FIFO CANALE 4	INIZIO TEST FIFO DEL CANALE 4
D0068	TEST BROADCAST FIFO	INIZIO TEST DI TUTTE LE FIFO
D006A	RESET FIFO CANALE 1	INIZIALIZZA LA FIFO DEL CANALE 1
D006C	RESET FIFO CANALE 2	INIZIALIZZA LA FIFO DEL CANALE 2
D006E	RESET FIFO CANALE 3	INIZIALIZZA LA FIFO DEL CANALE 3
D0070	RESET FIFO CANALE 4	INIZIALIZZA LA FIFO DEL CANALE 4
D0072	RESET BROADCAST FIFO	INIZIALIZZA TUTTE LE FIFO
D8xxx	WR ANALOG BOARD 1	SCRIVE VERSO LA ANALOG BOARD CH1
D9xxx	WR ANALOG BOARD 2	SCRIVE VERSO LA ANALOG BOARD CH2
DAxxx	WR ANALOG BOARD 3	SCRIVE VERSO LA ANALOG BOARD CH3
DBxxx	WR ANALOG BOARD 4	SCRIVE VERSO LA ANALOG BOARD CH4
DCxxx	WR ANALOG BOARD BROADCAST	SCRIVE VERSO TUTTE LE ANALOG BOARD CONTEMPORANEAMENTE

Le scritture KEEP ALIVE sono state usate solo durante la fase di sviluppo dell'elettronica, ad esse corrisponde l'invio sul link a fibra ottica del valore presente sul bus dati del PC104. Non è usato in condizioni operative, è stato comunque riportato in tabella per completezza di informazione.

Il comando TEST FIFO fa partire il test automatico delle memorie del canale selezionato, il test continua fino a quando la FIFO va in overflow (vedi test dei canali di acquisizione), tipicamente perché il PC smette di leggerla.

Il comando RESET FIFO inizializza tutta la logica del FIFO CONTROLLER e del blocco MEMORY CONTROLLER: tutti i contatori sono azzerati e tutte le macchine a stati logici vengono poste nello stato iniziale. Essendo comunque una scrittura, durante tale comando la logica campiona

anche il bit meno significativo del dato: se questo vale '0' la FIFO CONTROLLER funziona normalmente, se invece vale '1', essa verifica che il valore ricevuto dal pixel corrisponda alla sua posizione all'interno della riga, in caso negativo alza il bit più significativo (Vedi test dei canali di acquisizione). RESET FIFO è l'unico comando per la Interface Board nel quale, oltre all'indirizzo, viene preso in considerazione il dato.

WR ANALOG BOARD è la scrittura verso le Analog Board: avviene in uno spazio di indirizzamento pari a 2048 locazioni, ossia gli 11 bit di indirizzo possono assumere qualunque valore, per questo motivo sono indicati con xxx.

Il comando WR ANALOG BOARD BROADCAST fa sì che la scrittura avvenga su tutte le schede contemporaneamente, tipicamente si tratta di un comando relativo alla scansione del quadrante.

Tabella 4 : Mappa degli indirizzi di lettura.

<i><b>INDIRIZZO</b></i>	<i><b>MNEMONICO</b></i>	<i><b>DESCRIZIONE</b></i>
D0000	READ BOARD ID	LEGGE L'IDENTIFICATIVO DELLA SCHEDA
D0002	READ STATUS	LEGGE IL REGISTRO DI STATO
D0030	KEEP ALIVE CANALE 1	LEGGE IL KEEP ALIVE DAL CANALE 1*
D0032	KEEP ALIVE CANALE 2	LEGGE IL KEEP ALIVE DAL CANALE 2*
D0034	KEEP ALIVE CANALE 3	LEGGE IL KEEP ALIVE DAL CANALE 3*
D0036	KEEP ALIVE CANALE 4	LEGGE IL KEEP ALIVE DAL CANALE 4*
D0040	READ DELAY 0	LEGGE CON RITARDO 0 PASSI *
D0042	READ DELAY 1	LEGGE CON RITARDO 1 PASSO *
D0044	READ DELAY 2	LEGGE CON RITARDO 2 PASSI *
D0046	READ DELAY 3	LEGGE CON RITARDO 3 PASSI *
D0048	READ DELAY 4	LEGGE CON RITARDO 4 PASSI *
D004A	READ DELAY 5	LEGGE CON RITARDO 5 PASSI *
D004C	READ DELAY 6	LEGGE CON RITARDO 6 PASSI *
D004E	READ DELAY 7	LEGGE CON RITARDO 7 PASSI *
D0050	READ DELAY 8	LEGGE CON RITARDO 8 PASSI *
D0052	READ DELAY 9	LEGGE CON RITARDO 9 PASSI *
D0054	READ DELAY 10	LEGGE CON RITARDO 10 PASSI *
D0056	READ DELAY 11	LEGGE CON RITARDO 11 PASSI *

<b>INDIRIZZO</b>	<b>MNEMONICO</b>	<b>DESCRIZIONE</b>
D0058	READ DELAY 12	LEGGE CON RITARDO 12 PASSI *
D005A	READ DELAY 13	LEGGE CON RITARDO 13 PASSI *
D005C	READ DELAY 14	LEGGE CON RITARDO 14 PASSI *
D005E	READ DELAY 15	LEGGE CON RITARDO 15 PASSI *
D0080	READ LSW FIFO CH1	LEGGE LSW DELLA FIFO CANALE 1
D0082	READ LSW FIFO CH2	LEGGE LSW DELLA FIFO CANALE 2
D0084	READ LSW FIFO CH3	LEGGE LSW DELLA FIFO CANALE 3
D0086	READ LSW FIFO CH4	LEGGE LSW DELLA FIFO CANALE 4
D0088	READ MSW FIFO CH1	LEGGE MSW DELLA FIFO CANALE 1
D008A	READ MSW FIFO CH2	LEGGE MSW DELLA FIFO CANALE 2
D008C	READ MSW FIFO CH3	LEGGE MSW DELLA FIFO CANALE 3
D008E	READ MSW FIFO CH4	LEGGE MSW DELLA FIFO CANALE 4
D0090	READ FRAME CH1	LEGGE IL NUMERO DI FRAME DEL CANALE 1
D0092	READ FRAME CH2	LEGGE IL NUMERO DI FRAME DEL CANALE 2
D0094	READ FRAME CH3	LEGGE IL NUMERO DI FRAME DEL CANALE 3
D0096	READ FRAME CH4	LEGGE IL NUMERO DI FRAME DEL CANALE 4
D0098	READ DATA FIFO CH1	LEGGE I DATI DALLA FIFO DEL CANALE 1
D009A	READ DATA FIFO CH2	LEGGE I DATI DALLA FIFO DEL CANALE 2
D009C	READ DATA FIFO CH3	LEGGE I DATI DALLA FIFO DEL CANALE 3
D009E	READ DATA FIFO CH4	LEGGE I DATI DALLA FIFO DEL CANALE 4
D8xxx	READ ANALOG CH1	LEGGE DALLA ANALOG BOARD CANALE 1
D9xxx	READ ANALOG CH2	LEGGE DALLA ANALOG BOARD CANALE 2
DAxxx	READ ANALOG CH3	LEGGE DALLA ANALOG BOARD CANALE 3
DBxxx	READ ANALOG CH4	LEGGE DALLA ANALOG BOARD CANALE 4

Le operazioni di lettura contrassegnate con asterisco\* sono state riportate per completezza di informazione, ma non hanno più un ruolo attivo nel funzionamento della scheda. Furono implementate durante le fasi di test iniziali della scheda stessa, ma non sono più usate. La lettura KEEP ALIVE lavora in combinazione con la scrittura omonima, nel senso che collegando con la fibra ottica il trasmettitore ed il ricevitore della scheda Interface si realizza un loop locale nel quale con l'operazione di lettura KEEP ALIVE si legge il dato scritto con l'omonima operazione di scrittura.

L'operazione READ DELAY è stata implementata per verificare se e come il PFM supporta un certo numero di stati di attesa durante la lettura, in

particolare con le operazioni READ DELAY si legge un dato (da 0000 a 000F esadecimale) che viene reso disponibile sul bus EISA dopo un ritardo che varia da 0 a 15 passi di clock BCLK 8,33 MHz.

Con le operazioni READ LSW FIFO e READ MSW FIFO il PFM può leggere un numero che indica quanti dati ci sono nella memoria FIFO. Poiché la memoria di un canale si avvicina ai 5 Mega, sono necessari 23 bit per coprirne la dinamica, perciò una singola lettura di 16 bit non è sufficiente. In realtà, dovendo leggere con due parole a 16 bit, ci sono 32 bit a disposizione per contenere un dato formato da 23, ciò ha reso possibile aggiungere nella lettura della parola altri segnali e informazioni ad essi associate. Questi segnali aggiunti si chiamano ER, EM, OR, mentre i 23 bit che rappresentano la quantità di dati presenti nella FIFO sono etichettati da ND0 a ND22.

Il segnale ER al livello logico '1' indica che si è verificato un errore di scrittura nella memoria FIFO, perché essendo piena il valore di almeno un pixel è stato perso. È un segnale che inizia nell'istante in cui il componente programmabile riceve il dato del pixel e verifica che non può scriverlo in memoria perché piena, si mantiene fino alla lettura del PFM e si cancella automaticamente ritornando a '0' dopo esser stato letto.

Il bit EM sta al livello logico '1' per indicare che il registro temporaneo dei dati in uscita dalla FIFO non è vuoto, ovvero contiene un dato da leggere.

Il segnale OR è, di fatto, un or logico dei bit compresi tra ND13 e ND22. In pratica il suo significato è semplice: se al livello '1' indica che ci sono almeno altri 8192 dati da leggere, oltre quelli indicati dai bit LSW0 - LSW12.

Tabella 5 : Allineamento dei bit all'interno della lettura LSW.

LSW15	LSW14	LSW13	LSW12	LSW11	LSW10	LSW9	LSW8	LSW7	LSW6	LSW5	LSW4	LSW3	LSW2	LSW1	LSW0
<b>ER</b>	<b>EM</b>	<b>OR</b>	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0



Tabella 6 : Allineamento dei bit all'interno della lettura MSW. Nota: ND17 – ND22 sono ripetuti.

MSW15	MSW14	MSW13	MSW12	MSW11	MSW10	MSW9	MSW8	MSW7	MSW6	MSW5	MSW4	MSW3	MSW2	MSW1	MSW0
ND22	ND21	ND20	ND19	ND18	ND17	ND22	ND21	ND20	ND19	ND18	ND17	ND16	ND15	ND14	ND13

READ BOARD ID è la lettura dell'identificativo di scheda, con questa operazione il PFM riconosce quale delle tre Interface Board prodotte è realmente connessa. L'identificativo può essere uno di questi tre valori, in codice esadecimale: EBB1, EBB2 o EBB3, a seconda della scheda. EBB sono le iniziali delle persone che hanno lavorato per lo sviluppo dell'elettronica: Elisabetta Giani e Baffa Carlo per il software, Biliotti Valdemaro per la parte hardware.

READ STATUS è un'operazione che consente di leggere un registro nel quale sono memorizzati dei bit di stato o di errore del sistema.

Tabella 7: registro di STATUS

ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
DER	DER	DER	DER	ERR	ERR	ERR	ERR	OK	OK	OK	OK	TST	TST	TST	TST
CH4	CH3	CH2	CH1	CH4	CH3	CH2	CH1	CH4	CH3	CH2	CH1	CH4	CH3	CH2	CH1

Sui bit ST0, ST1, ST2 e ST3 si trovano i segnali TST CH1, TST CH2, TST CH3 e TST CH4, questi segnali sono al livello logico '1' mentre viene eseguito il test della memoria FIFO.

Sui bit da ST4 a ST7 ci sono i segnali da OK CH1 a OK CH4, questi segnali indicano, se al livello logico '1', che il ricevitore del link a fibre ottiche non ha rilevato errori, che tutto funziona regolarmente. Come sarà descritto più avanti, questi segnali copiano i segnali LINK\_OK generati dai ricevitori dei link, rivelando se il funzionamento è regolare oppure no. Data la logica di funzionamento dei link, in cui il riferimento del clock di sistema è inviato alle schede Analog Board e da loro ritorna indietro, questo segnale indica che il link funziona correttamente nel suo complesso, sia in trasmissione che in ricezione. Ovviamente la sua negazione, ossia quando il segnale è al livello logico '0', significa che il link non sta funzionando correttamente, per esempio perché la fibra ottica non è connessa o le schede analogiche non sono alimentate.

Sui bit ST8, ST9, ST10, ST11 trovano posto i segnali da ERR CH1, ERR CH2, ERR CH3 e ERR CH4. Come sarà spiegato nella descrizione dei link, anche un minimo errore (solo un passo di clock), genera il segnale LINK ERROR, che è presente per la sola durata dell'errore medesimo, dunque si tratta di un segnale istantaneo, che potrebbe non essere rilevato dal PC104. Per questo motivo dai segnali LINK ERROR si ricavano i segnali ERR, secondo la seguente logica: se il link Receiver genera il segnale LINK ERROR, questi a sua volta fa andare al livello logico '1' il segnale ERR corrispondente, questo '1' logico viene mantenuto fino al completamento della lettura di stato. Pertanto il suo significato è il seguente: “dalla precedente lettura si è verificato un errore (di almeno un passo di clock) nel link”.

Sui bit ST12, ST13, ST14 e ST15 trovano posto i segnali da DER CH1, DER CH2, DER CH3 e DER CH4. Questi segnali derivano dai segnali DATA ERROR generati dal ricevitore del link a fibre ottiche, la differenza tra DATA ERROR e LINK ERROR dipende dal momento in cui avviene l'errore: se l'errore avviene durante la ricezione di un dato, allora è il segnale DATA ERROR che va a '1', se invece l'errore accade in altro contesto, per esempio quando il link riceve indietro solo il clock di sistema dalla scheda analogica, allora è il segnale LINK ERROR che va a '1'. In conclusione, quando un link Receiver si accorge di un errore durante la ricezione di un dato, mette il segnale DATA ERROR al livello logico '1'. A sua volta il segnale DATA ERROR fa andare al livello logico '1' il segnale DER corrispondente, questo '1' viene mantenuto fino al completamento della lettura di stato. Pertanto il suo significato è il seguente: “dalla precedente lettura si è verificato un errore nel link durante la ricezione di un dato”.

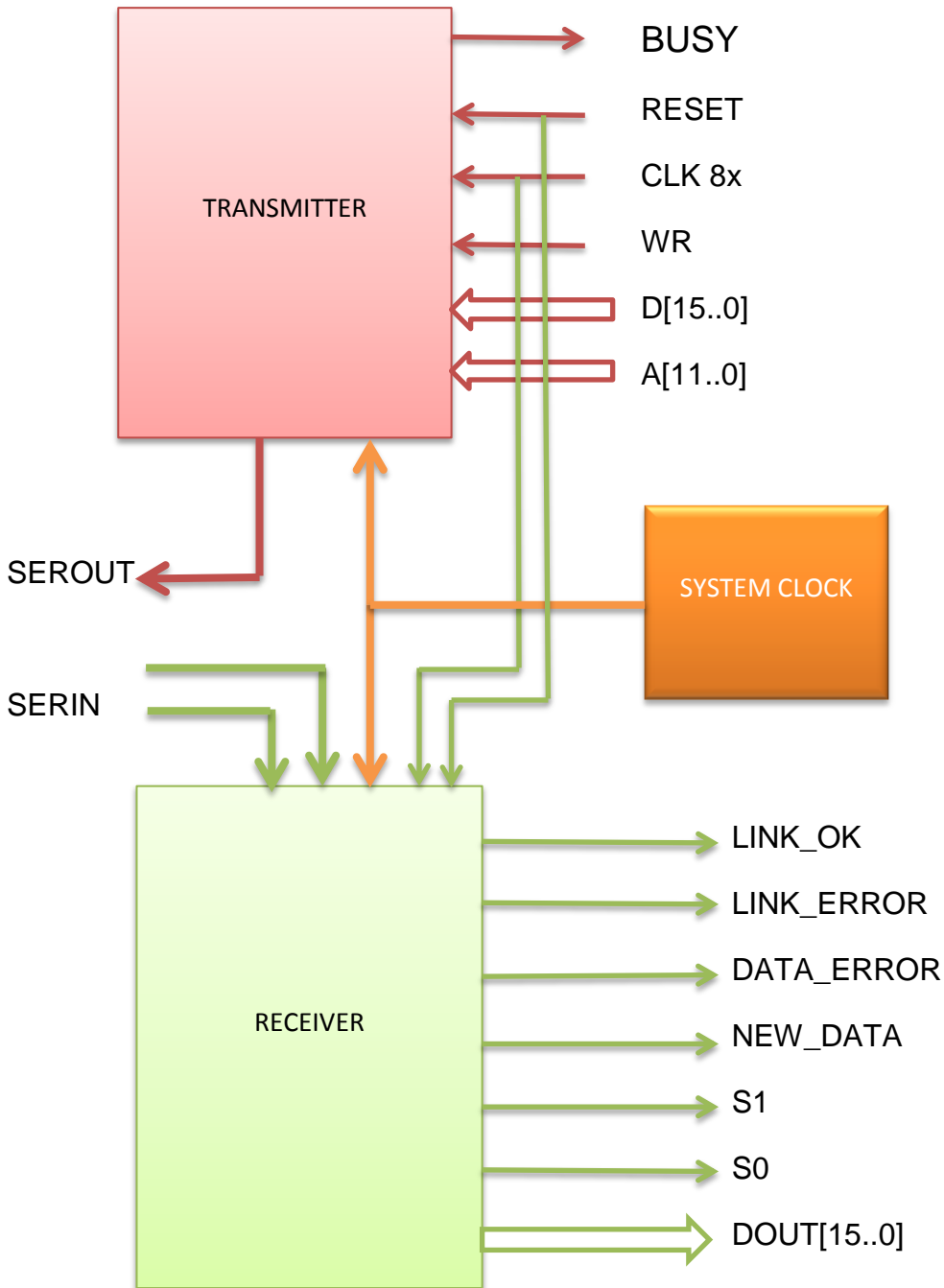
READ FRAME è un comando di lettura con il quale il PFM ha accesso al contatore progressivo di immagini. Tale contatore viene azzerato dal comando RESET FIFO ed incrementa di uno il suo valore ogni volta che riceve una nuova immagine. Il suo ruolo sarà descritto più avanti, nella sezione dedicata alla gestione di tipo FIFO della memoria.

READ DATA FIFO è il comando per mezzo del quale vengono lette le immagini contenute in memoria. La codifica dei dati e la descrizione esatta del funzionamento sarà data più avanti, nella sezione dedicata alla gestione delle memorie, per il momento due sono le cose importanti da evidenziare: la prima è che la lettura avviene alla velocità massima consentita dal bus EISA, la seconda è che il PFM vede la memoria come una porta di input, non ha la

possibilità di indirizzare singolarmente una locazione, la dimensione del bus non è sufficiente per farlo, dunque il PFM legge l'immagine come una sequenza di dati organizzati dalla prima riga all'ultima.

READ ANALOG è la lettura da una scheda Analog Board, come si può notare, la parte meno significativa degli indirizzi è xxx (3 cifre esadecimali), perché in realtà non si tratta di un indirizzo singolo, bensì di una zona di indirizzamento a cui corrispondono aree di memoria e locazioni di registri o comandi. Sebbene tre cifre esadecimali rappresentino 12 bit, in questo caso sono 11 i bit veramente utilizzati, manca appunto il bit meno significativo, che ha perso la sua ragione d'esistere visto che lo scambio dati sul bus avviene a 16 bit. In realtà la rappresentazione continua ad essere a 12 bit, ma è sempre pari, ovvero il bit meno significativo è sempre al livello '0' logico. Anche se questo comando è il simmetrico di WR ANALOG BOARD, a differenza di quest'ultimo non è prevista l'opzione broadcast: non avrebbe senso logicamente e, seppur realizzabile nel progetto, avrebbe comunque generato un conflitto di attribuzione del bus EISA. Per queste due ragioni non è stata implementata.

Figura 11 : Il link controller.



Come si vede dalla figura 11, il link controller di un canale è formato da un trasmettitore e da un ricevitore. Il compito principale del trasmettitore è quello di trasmettere dati opportunamente miscelati a un clock, che è sempre presente. La frequenza del clock che chiameremo di di sistema, perché

costituisce il riferimento rispetto al quale ogni singolo canale di acquisizione si sincronizza, è fissata dall'oscillatore sulla scheda Interface Board a 16.384 MHz. La miscelazione del clock con i dati avviene dopo che questi sono stati serializzati e sulla base di un clock a frequenza 8 volte tanto quella del clock medesimo (131MHz), la moltiplicazione in frequenza è ottenuta con un PLL interno al componente FPGA. Mentre la trasmissione del clock di sistema è continua, quella dei dati avviene perché la logica interna del componente programmabile decodifica un comando destinato ad una, o tutte se broadcast, le schede Analog Board. In dettaglio, se si tratta di un comando di lettura, la logica interna preleva dal bus EISA copia dell'indirizzo, lo invia all'ingresso del blocco TRANSMITTER ed asserisce il segnale WR. Da quell'istante inizia la trasmissione del dato e contemporaneamente il segnale BUSY va al livello logico '1' per tutto il tempo necessario alla trasmissione. In modo analogo succede per un comando di scrittura, la logica interna preleva dal bus copia dell'indirizzo e del dato, li invia al blocco TRANSMITTER ed asserisce il segnale WR. Anche in questo caso in segnale BUSY sta al livello logico '1' per tutto il tempo necessario alla trasmissione.

La codifica per la lettura è descritta nella tabella 8, la trasmissione dei 16 bit avviene procedendo da sinistra verso destra.

Tabella 8:

0	0	1	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	1	1
---	---	---	-----	----	----	----	----	----	----	----	----	----	----	---	---

La trasmissione inizia con due bit al livello '0', poi il terzo dato trasmesso è un '1', che significa operazione di lettura, seguono 11 bit di indirizzo, dal più al meno significativo. La trasmissione termina con due bit al livello '1'. La stringa di bit da A10 al A0 copia esattamente i bit del bus indirizzi EISA da SA11 a SA1, in pratica all'interno della FPGA viene fatto lo shift di un bit rispetto al bus, come conseguenza del fatto che il bit meno significativo del bus EISA perde valore e significato lavorando a 16 bit.

La codifica della trasmissione per la scrittura consiste di 32 bit, è descritta nella tabella 9, la sequenza dei dati va da sinistra verso destra e dall'alto verso il basso.

Tabella 9:

0	0	0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	D15	D14
D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	1	1

Anche in questo caso la trasmissione inizia con due bit al livello '0', il terzo bit trasmesso in questo caso è uno '0', che significa operazione di scrittura; seguono 11 bit di indirizzo, dal più significativo al meno, e 16 bit di dato sempre in ordine decrescente di peso. Due bit al livello '1' completano e chiudono la sequenza dei dati trasmessi. Il segnale SEROUT esce dal componente programmabile miscelato con il clock di sistema e va all'ingresso del driver per la fibra ottica.

Il segnale RESET deriva direttamente dal segnale RESET FPGA presente sul pannello della scatola che contiene la scheda, è presente di default all'accensione; come è stato accennato ha una durata di circa 16 secondi, durante i quali sia la logica di trasmissione che quella di ricezione sono inoperative.

La Interface Board assume che sia canale di acquisizione 1 la scheda Analog Board che è connessa al Link 1 (più a sinistra sul frontale), canale 2 quella connessa al Link 2, e via così fino al link più a destra del frontale, che la scheda considera link 4 e perciò riferito al canale 4. Nota: non è detto che la numerazione dei canali di acquisizione coincida con la numerazione dei quadranti del sensore.

Trasmettere continuamente un segnale, in questo caso il clock, mentre da un lato permette di sincronizzare i canali di acquisizione, dall'altro da modo di implementare dei meccanismi di auto test per il link a fibre ottiche. Nell'elettronica di GIANO, ogni link si riferisce ad un canale di acquisizione ed è full duplex: il clock parte dal trasmettitore della scheda Interface, arriva al ricevitore di una scheda Analog Board, e da questo inviato al componente programmabile, che lo usa per sincronizzare se stesso e i segnali di scansione per il quadrante del sensore. Naturalmente il clock ricevuto deve superare un processo di verifica prima di essere usato come riferimento, questa operazione è svolta dal componente programmabile FPGA che gestisce la Analog Board. Se il clock ricevuto è conforme a quanto aspettato il componente programmabile lo usa come riferimento e, fra le altre cose, lo rimanda indietro verso la Interface Board, come una specie di eco; in

presenza di errori, il clock di sistema non viene spedito indietro. Come conseguenza di tale meccanismo, se il componente programmabile della Interface Board rileva che il clock di sistema è tornato indietro, può giustamente ritenere che tutto il link sta funzionando correttamente. Questa organizzazione del clock ha richiesto un metodo non convenzionale per la codifica dei dati su fibra ottica, ma una volta implementata offre due vantaggi. Il primo, quello per il quale è stato pensato, è che i segnali per la gestione del sensore possono essere sincronizzati tra di loro entro 1 nano secondo. Il secondo è che il sistema così congegnato, non solo è in grado di riconoscere gli errori, ma anche di misurare quanti ne sono avvenuti, misurando perciò il Bit Error Rate di se stesso in tempo reale.

Il blocco logico Link Receiver è collegato alle uscite del dispositivo che traduce il segnale ottico ricevuto dalle Analog Board in una sequenza bilanciata di dati digitali. Trattandosi di due segnali bilanciati essi sono, per definizione, l'uno l'inverso dell'altro. Questa coppia di segnali viene campionata ad una frequenza di 131MHz, il clock interno 8x, per separare il clock di sistema dai dati eventualmente presenti. Questa operazione conduce a due verifiche: che la coppia sia veramente bilanciata e che sia presente il clock di sistema. Se viene rivelata l'assenza del clock, anche di un solo passo, la logica del ricevitore mette il segnale LINK\_ERROR al livello logico '1'. I passi di clock vengono continuamente verificati e contati, per mezzo di un contatore che si incrementa ogni volta che manca un passo, e che si decrementa ogni volta che il passo di clock è presente e corretto. Se il valore del contatore è uguale a zero il segnale LINK\_OK viene asserito al livello logico '1', se il contatore supera  $10^4$  il segnale LINK\_OK va al livello logico '0'. La commutazione fra i due livelli avviene con memoria di isteresi, pertanto mentre LINK\_ERROR è un segnale che indica un solo passo di clock perso, LINK\_OK indica un malfunzionamento decisamente più grave, tipicamente perché la fibra non è correttamente collegata, oppure perché la scheda analogica non è alimentata. Va detto che questo meccanismo di controllo degli errori nel link a fibre ottiche rivela sistematicamente anche il fatto che le schede analogiche non siano alimentate. Siccome l'alimentazione della parte interfaccia è diversa da quella delle parti analogiche, quando l'elettronica viene accesa è praticamente certo che una delle due parti rilevi errori di link, perché chi viene alimentato prima si accorge che il link di chi viene alimentato dopo non funziona. Vedere questa segnalazione all'accensione (o dopo un

RESET FPGA) è da ritenersi cosa normale, significa che la logica di test dei link sta funzionando correttamente.

Il formato seriale dei dati ricevuti è riportato nella tabella 10, da sinistra verso destra.

Tabella 10 : Formato seriale dei dati in ingresso al link

0	0	S1	S0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	1	1
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		

La serie di dati inizia con una coppia di '0' , poi seguono due bit S1 e S0 di selezione, poi i 16 bit del dato trasferito e infine una coppia di '1' a chiusura.

Il dato che arriva in ingresso al ricevitore della Interface Board proviene da una Analog Board, pertanto può rappresentare il contenuto di registri interni del componente programmabile, può rappresentare il contenuto della memoria del Sequencer Processor (vedi descrizione Elettronica di acquisizione di Giano, sezione analogica di conversione); può rappresentare il campionamento dei pixel, oppure informazioni di inizio frame o inizio riga del quadrante. Per essere riconosciuti e gestiti correttamente questi dati in arrivo devono essere accompagnati da un identificativo, che permetta il loro riconoscimento. I bit S1 e S0 svolgono di fatto questa funzione, come da tabella 11.

Tabella 11: Significato dei bit S1 e S0.

<i>CONFIGURAZIONE</i>		<i>TIPO DI DATO</i>
<i>S1</i>	<i>S0</i>	
0	0	DATO DI UN REGISTRO O MEMORIA
0	1	VALORE DI UN PIXEL CAMPIONATO
1	0	IDENTIFICATIVO INIZIO RIGA
1	1	IDENTIFICATIVO INIZIO FRAME

La logica Receiver del link separa il clock di sistema ricevuto come eco dai dati seriali, verifica le sue temporizzazioni e contemporaneamente converte il dato ricevuto da seriale a parallelo. In questa fase viene misurato



anche il ritardo del clock fra la trasmissione e la ricezione, per verificare la stabilità nel tempo, tale parametro è scritto in un registro interno e concorre, insieme alla presenza o assenza del clock di sistema, per stabilire se la ricezione dei dati è avvenuta correttamente.

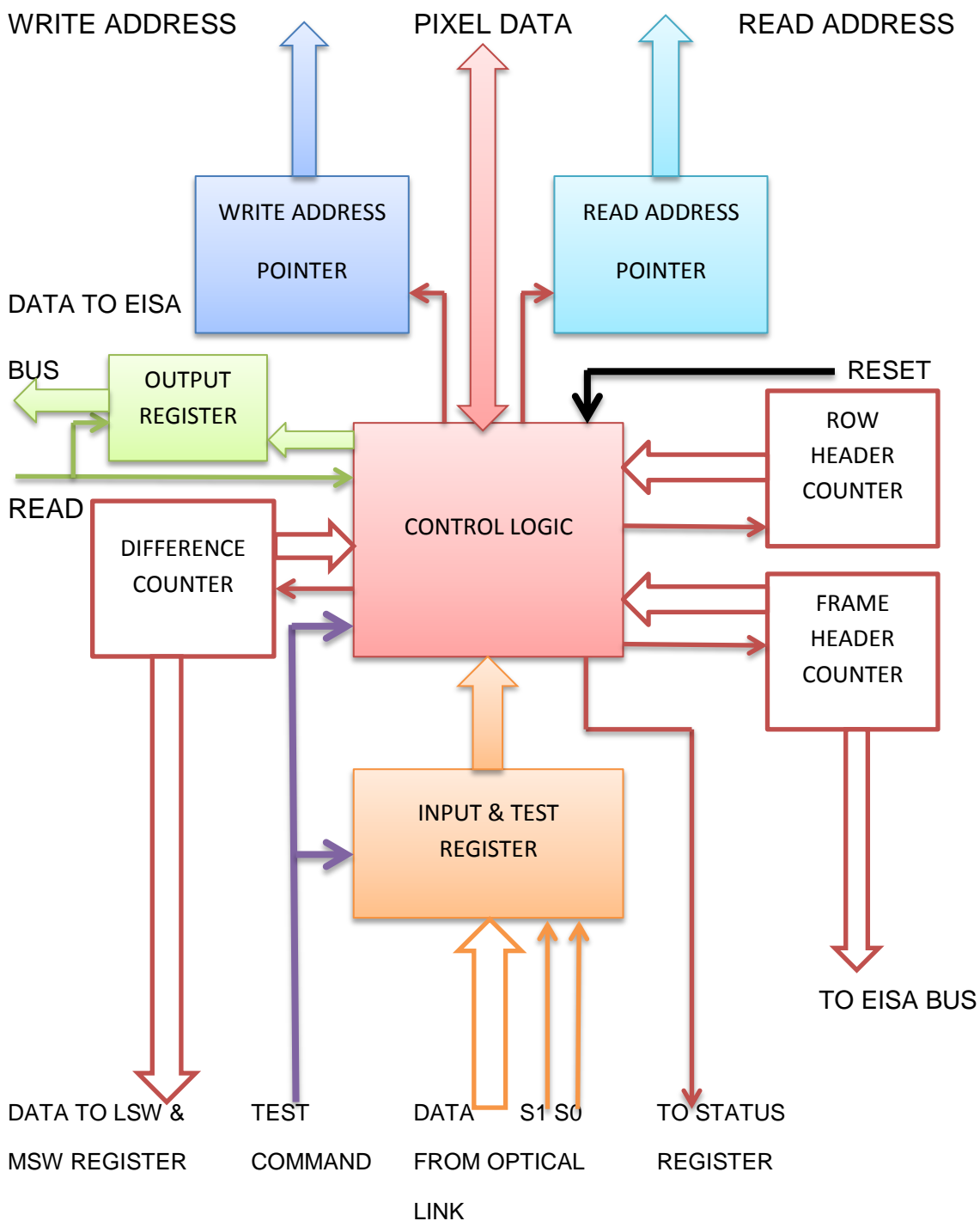
In caso di errore durante la ricezione di un dato viene messo al livello logico '1' il segnale DATA\_ERROR, la procedura di riconoscimento dei dati viene abortita e il bus interno non viene aggiornato. Se nessun errore viene rilevato e la sequenza seriale è corretta, allora il segnale NEW\_DATA va al livello logico '1' per indicare che sul bus interno D0 – D15 c'è un nuovo dato etichettato dai bit S1 e S0.

Il dato presente sul bus interno del componente programmabile viene trattato e gestito in maniera diversa, in funzione dell'etichetta definita dalla configurazione dei segnali S0 e S1. La situazione S1='0' e S0='0' si verifica quando il PFM esegue una lettura da una scheda Analog Board, in questo caso il dato viene prelevato dal bus interno e presentato sul bus EISA per completare l'operazione di lettura del PC104. L'attesa del bus EISA termina automaticamente quando viene ricevuto un dato con la giusta etichetta, è stato previsto comunque un time out di 6 microsecondi, oltre il quale la logica a bordo del componente programmabile considera che la ricezione non sia andata a buon fine per effetto di un errore. L'evento è facilmente verificabile dal PFM leggendo il registro di stato ed in particolare i bit DER, che derivano dai segnali DATA\_ERROR.

Quando la configurazione di S1 e S0 indica che il dato ricevuto si riferisce alla scansione del quadrante, entra in gioco un'altra logica interna al componente FPGA, quella che gestisce la memoria e l'organizzazione dell'immagine dei quadranti. Sono le parti che nello schema a blocchi del componente programmabile sono indicate come FIFO CONTROLLER e MEMORY CONTROLLER (vedi figura 8). Mentre i blocchi FIFO CONTROLLER sono 4, ossia uno per ogni link a fibra ottica, ovvero uno per ogni canale di acquisizione, c'è un solo blocco MEMORY CONTROLLER, perché la memoria è unica: tutti i dispositivi di memoria condividono i bus (dati e indirizzi), come un array convenzionale di memorie. La separazione logica in canali dipende dalla gestione, che in definitiva dipende dalla logica interna al componente programmabile, il quale mappa i diversi canali di acquisizione in quattro zone di memoria ben distinte tra loro, anche se i segnali di controllo e il bus sono in comune.

Come è stato anticipato, i dati coinvolti in queste parti di logica sono sostanzialmente legati alla scansione del quadrante: principalmente i valori campionati dei pixel e le informazioni di frame (inizio quadro e inizio riga). Visti come sequenza temporale, questi dati iniziano con il sincronismo di frame, che indica l'inizio della scansione del quadrante, poi ci sono 1024 sequenze (una per ogni riga) di dati, che a loro volta sono composte dal sincronismo di riga seguito da 1040 valori campionati. L'immagine del quadrante è formata da 1024x1040 dati, organizzate in 1024 righe x 1024 colonne di pixel veri, a cui vengono aggiunti 16 extra-pixel per ogni riga. Gli extra-pixel derivano da una caratteristica del sensore HAWAII2: se pilotato opportunamente, dopo i 1024 pixel reali, il sensore mette in uscita dei "pixel virtuali", il cui valore è ottenuto da una tensione di riferimento interna e non da un elemento sensibile alla radiazione infrarossa. Tali valori possono tornare utili in fase di riduzione dati, per tenere conto di fluttuazioni legate alla variazione della temperatura, che si traducono in rumore in bassa e bassissima frequenza.

Figura 12: FIFO CONTROLLER



Quando inizia la scansione di un quadrante, il primo dato che arriva dal ricevitore del link a fibre ottiche è un dato, che chiameremo FRAME SYNC, il cui scopo è quello di informare l'Interface Board che stanno per arrivare i dati di una immagine. Come si desume dalla Tabella 11, il dato che arriva è

etichettato con  $S1='1'$  e  $S0='1'$ . Questo dato non viene scritto nella memoria, ma serve per incrementare di 1 il valore contenuto nel contatore FRAME HEADER COUNTER, azzerare il contatore ROW HEADER COUNTER, e verificare che in memoria ci sia spazio sufficiente per contenere un'immagine. Questa operazione viene effettuata dalla CONTROL LOGIC leggendo il valore del DIFFERENCE COUNTER. Se lo spazio in memoria è adeguato, la logica di controllo si predispone a scrivere i dati che seguiranno, altrimenti genera un segnale che inibisce le scritture successive. Tale segnale nasce per effetto della verifica effettuata per l'arrivo del segnale FRAME SYNC, e si mantiene automaticamente fino al prossimo sincronismo di frame. Data la logica di generazione e permanenza, la presenza di questo segnale implica che tutti i dati di un'immagine vengano scartati fino a prossima verifica (il prossimo FRAME SYNC), questo meccanismo garantisce che i dati scritti in memoria siano un'immagine completa. Nel dettaglio, quando il valore campionato di un pixel viene scartato, perché lo spazio disponibile in memoria non è sufficiente per contenere un'immagine completa, il segnale ER va al livello logico '1'. Il livello del segnale ER può essere letto dal PC104 con il comando READ LSW, vedi Tabelle 4 e 5

Dopo il segnale FRAME SYNC, arriva dal ricevitore il segnale LINE SYNC, riconosciuto come tale per l'etichetta  $S1='1'$  e  $S0='0'$ , che produce due effetti: per prima cosa viene incrementato di 1 il valore del contatore ROW COUNTER, successivamente il blocco INPUT & TEST REGISTER genera una sequenza di 4 parole a 16 bit che costituiscono un identificativo di inizio riga, d'ora in avanti chiamato HEADER.

Un HEADER è un gruppo ordinato di 4 parole a 16 bit, definite in codice esadecimale come segue: la prima parola è FFFF, la seconda è il contenuto del contatore FRAME COUNTER, la terza parola è il valore del contatore ROW COUNTER, la quarta è 0000. Dunque gli HEADER di un'immagine riportano tutti al secondo posto lo stesso numero di FRAME, mentre al terzo c'è la numerazione progressiva per le righe. Dopo la scrittura dello HEADER, la logica INPUT & TEST REGISTER in condizioni operative riceve tutti i pixel della riga seguiti dai 16 extra-pixel, tutti questi dati sono dati etichettati con  $S1='0'$  e  $S0='1'$ .

Durante la ricezione dei pixel viene fatto un controllo temporale: la logica conosce l'intervallo di tempo compreso tra l'arrivo di un pixel e il successivo. Se tale intervallo viene superato senza un motivo valido, come

potrebbe essere la lettura di un registro di una Analog Board, che può ritardare la trasmissione di un pixel, assume che tale ritardo sia causato da un errore nella ricezione del dato: verifica perciò la presenza del segnale DATA\_ERROR, e se del caso, inserisce un pixel al valore esadecimale 000F. In definitiva, un pixel eventualmente perso a causa di un errore nel link a fibra ottica viene sostituito da un pixel al valore 15 (decimale) e viene data segnalazione con il bit DER CH nel registro STATUS.

Sommando le 4 parole di HEADER all'inizio di una riga si ottiene che un'immagine è composta da 1024 righe per 1044 dati, cioè 1069056 parole a 16 bit.

La codifica dello HEADER è univoca e facilmente riconoscibile, i valori 0000 e FFFF non sono consentiti per i pixel campionati: c'è infatti una piccola compressione della dinamica dovuta al fatto che ogni Analog Board converte in 0001 un campione al valore 0000, e converte in FFFE il campione al valore FFFF.

Che si tratti dei dati HEADER o dati campionati dei pixel, ogni volta che un nuovo valore deve essere scritto in memoria, la CONTROL LOGIC invia alla logica MEMORY CONTROLLER il dato da scrivere e il suo indirizzo, fornito dal WRITE ADDRESS POINTER. Ogni volta che un nuovo dato deve essere scritto il contenuto del WRITE ADDRESS POINTER viene incrementato di 1, lo stesso incremento succede per il DIFFERENCE COUNTER. Mentre i contatori WRITE e READ ADDRESS POINTER possono solo incrementare di 1 il loro valore, il primo per effetto di una scrittura, il secondo per una lettura dalla memoria, il DIFFERENCE COUNTER è un contatore up-down, che incrementa di 1 per effetto di una scrittura e decrementa di 1 come conseguenza di una lettura. In pratica mentre i POINTER forniscono gli indirizzi di scrittura o lettura, il DIFFERENCE COUNTER contiene un valore che indica quanti dati da leggere ci sono in memoria. Il suo contenuto viene valutato dalla CONTROL LOGIC per stabilire, all'arrivo del segnale FRAME SYNC, se lo spazio in memoria è sufficiente per contenere l'immagine in procinto per arrivare. Tale valore è, come abbiamo visto, rappresentato con 23 bit, che possono essere letti dal PC104 con i comandi READ LSW e READ MSW. Inoltre, il contenuto del DIFFERENCE COUNTER viene costantemente valutato dalla CONTROL LOGIC e dalla logica MEMORY CONTROLLER, prima di iniziare il trasferimento di un dato dalla memoria al registro temporaneo di uscita.

Quando il bus EISA esegue una lettura dei dati della FIFO, accede in realtà al blocco OUTPUT REGISTER, un registro temporaneo dei dati, questa struttura permette al PFM di eseguire la lettura dei pixel alla massima velocità consentita dal bus. Dopo che la lettura è stata effettuata, la CONTROL LOGIC, insieme al MEMORY CONTROLLER, si accorge che il contenuto del registro temporaneo è stato letto e quindi può essere aggiornato con un nuovo dato prelevato dalla memoria. Se il valore del DIFFERENCE COUNTER è diverso da zero (ci sono dati da leggere) incrementa di 1 il contatore READ ADDRESS POINTER, decrementa di 1 il DIFFERENCE COUNTER ed effettua il trasferimento dalla memoria all'OUTPUT REGISTER. I dati sono organizzati con gli HEADER, sono letti nell'ordine seguendo l'ordine temporale con cui sono stati scritti, ma con temporizzazioni di scrittura e lettura che possono essere completamente diverse: questa gestione viene definita FIFO, dall'acronimo First In First Out.

Il comando RESET FIFO coinvolge questa parte logica: azzerare tutti i contatori (WRITE e READ POINTER, DIFFERENCE COUNTER), inizializza la logica INPUT & TEST REGISTER e la CONTROL LOGIC, i dati presenti nella memoria non vengono cancellati, ma non possono essere più letti. Come già accennato, comando RESET FIFO scaturisce dalla decodifica di un indirizzo ma, unico fra i comandi, anche il dato ha un suo ruolo. In particolare il bit meno significativo viene campionato durante questo comando, se il bit è al livello logico '0', la logica INPUT & TEST REGISTER si comporta come un normale registro, se invece il bit meno significativo del dato è al livello logico '1', la logica continua a campionare i dati in arrivo, ma verifica anche che il valore del dato ricevuto corrisponda alla sua posizione nella riga. È un test messo a punto in laboratorio per verificare la qualità della trasmissione, e che coinvolge anche le Analog Board: in pratica esse trasmettono una serie di dati di test in cui il valore del pixel corrisponde alla sua posizione nella riga. La Interface Board verifica, se il bit meno significativo è '1' durante il RESET, che effettivamente la serie di dati sia così strutturata e, se rivela una non coerenza tra posizione e valore, mette il bit più significativo del valore al livello logico '1'. Questo rende immediatamente visibili i dati sbagliati sul monitor del computer di controllo dello strumento.

Il comando TEST FIFO agisce sulla logica INPUT & TEST REGISTER, trasformandone il comportamento da semplice registro a una macchina di stati in grado di generare sequenze ben definite di dati. In condizioni operative la logica INPUT & TEST REGISTER acquisisce i dati dal link e li

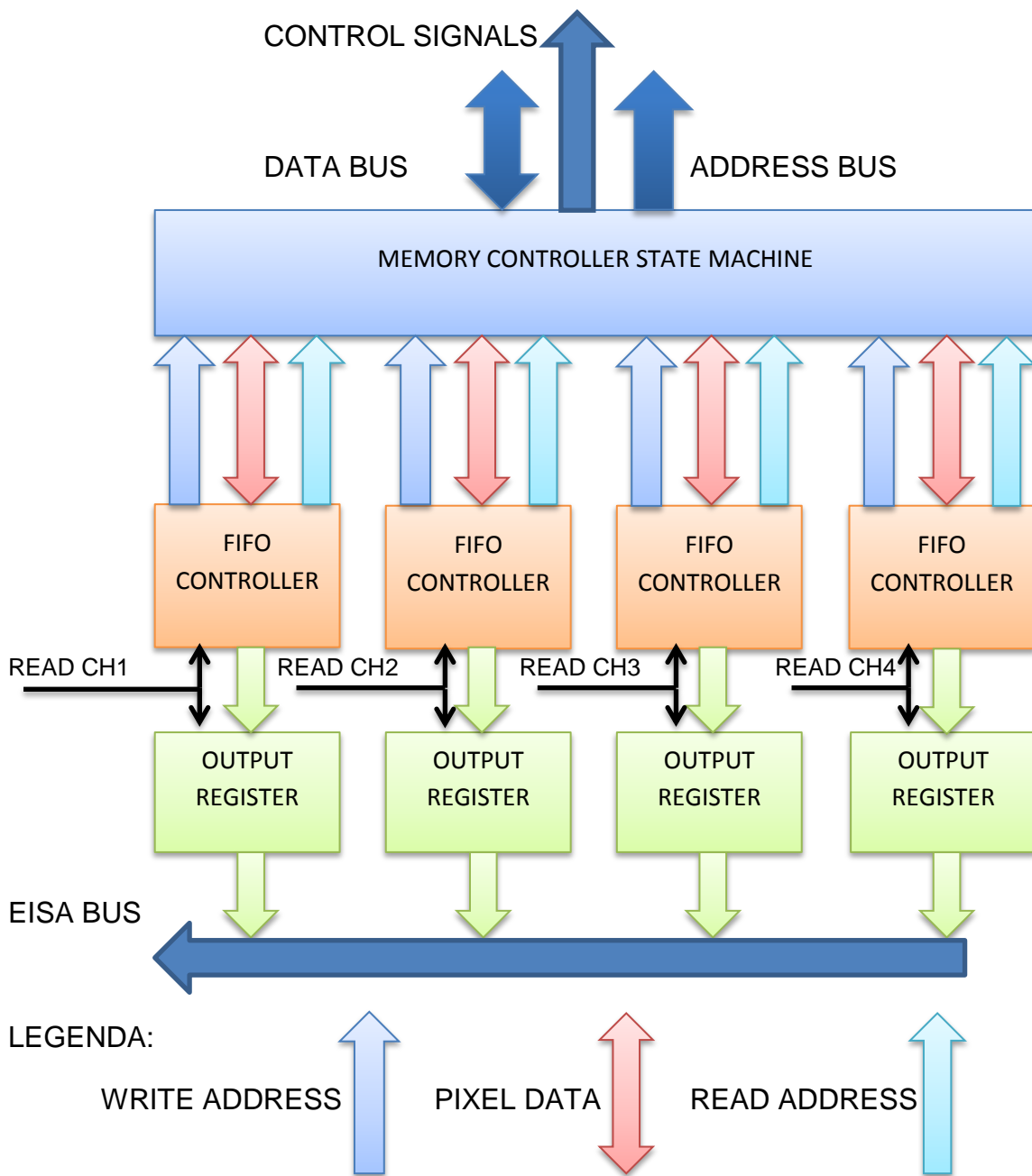
trasferisce verso la logica di controllo aggiungendo eventualmente gli HEADER come il riferimento di inizio riga. Durante il test della memoria questa logica continua a generare gli HEADER, ma al posto dei dati ricavati campionando i pixel mette la sequenza di dati riportata in tabella 12.

Tabella 12: Pattern di test.

Posizione	Dato (HEX)
1	0000
2	0001
3	0002
4	0004
5	0008
6	0010
7	0020
8	0040
9	0080
10	0100
11	0200
12	0400
13	0800
14	1000
15	2000
16	4000
17	8000

Come si può facilmente intuire, a parte il primo dato, gli altri sono ottenuti facendo muovere un '1' dal bit meno al più significativo. Il pattern ha periodo 17 e si ripete continuamente. La scrittura dei pixel di test avviene ogni 4,5  $\mu$ Sec e va avanti fino a quando la memoria non è più in grado di contenere dati, in pratica è il blocco CONTROL LOGIC che ferma il test con la stessa logica descritta per le condizioni operative. Non esiste il comando che ferma il test dal Bus EISA, perché essendo asincrono rispetto alla generazione del pattern, la avrebbe con molta probabilità interrotta in un punto casuale, lasciando l'ultima immagine incompiuta. Se il PFM legge i dati di test in continuazione, la memoria non si riempie mai, pertanto il test continua senza sosta fino a quando il PC104 smette di leggere.

Figura 13: MEMORY CONTROLLER



Il blocco logico MEMORY CONTROLLER è una macchina a stati logici che esegue in continuazione operazioni di scrittura e lettura sulle memorie. Le esegue sulla base di alcune informazioni, in particolare scrive in memoria se almeno una logica FIFO CONTROLLER indica che un nuovo dato (HEADER o pixel) è presente, legge dalla memoria se almeno un OUTPUT REGISTER è stato letto e va aggiornato con un nuovo dato. È il crocevia dove si concentrano tutti i dati in arrivo dai 4 link a fibre ottiche e da dove ripartono verso il bus EISA, gli arrivi e le partenze non sono correlati fra loro, perché le temporizzazioni di arrivo dipendono dalla frequenza di campionamento dei quadranti, mentre la temporizzazione di partenza



dipende dal bus EISA, esse sono temporizzazioni assolutamente asincrone. La logica MEMORY CONTROLLER riesce a risolvere eventuali conflitti di temporizzazione lavorando con clock 2x a 32.7MHz, ricavato internamente al componente programmabile con un PLL che moltiplica per 2 la frequenza dell'oscillatore esterno.

La logica controlla se è arrivato un nuovo dato da scrivere da un certo canale di acquisizione, successivamente controlla se il registro OUTPUT REGISTER di quel canale è stato letto, ripetendo questa coppia di verifiche ciclicamente per tutti e quattro i canali. Quando c'è un nuovo dato da scrivere, preleva il PIXEL DATA (frece rosse) e il WRITE ADDRESS (frece blu), trasforma l'indirizzo logico WRITE ADDRESS in indirizzo fisico per le memorie ed esegue il ciclo di scrittura. In modo del tutto analogo si comporta quando il registro di uscita è stato letto dal bus EISA, la macchina a stati logici preleva il READ ADDRESS (frece azzurre) dalla logica FIFO CONTROLLER corrispondente, lo trasforma in indirizzo fisico per la memoria, legge i dati dalla memoria (frece rosse) e li invia alla logica OUTPUT REGISTER. La MEMORY CONTROLLER esegue un ciclo di lettura in 5 passi di clock a 32.7MHz, di conseguenza riesce ad aggiornare un OUTPUT REGISTER in circa 153 nSec dopo che è stato letto dal PFM, perciò l'aggiornamento avviene molto in anticipo rispetto alla lettura successiva, che teoricamente non succede prima che siano trascorsi 360 nSec.

Riferimenti bibliografici:

- **“The GIANO control software system”** Emanuel Rossetti\* , Ernesto Oliva, Livia Origlia [http://www.arcetri.astro.it/irlab/doc/giano/spie/giano\\_control\\_system\\_2008.pdf](http://www.arcetri.astro.it/irlab/doc/giano/spie/giano_control_system_2008.pdf)
- **“The Versatile Acquisition System of Giano”** Carlo Baffa, Valdemaro Biliotti, Sandro Gennari, Elisabetta Giani, Iacopo Mochi, Ernesto Oliva, Livia Origlia, Emanuel Rossetti, Mauro Sozzi [http://www.arcetri.astro.it/irlab/doc/spie\\_06/6274-33\\_t.pdf](http://www.arcetri.astro.it/irlab/doc/spie_06/6274-33_t.pdf)
- **“Progetto Giano – Memo Specifiche temporizzazioni del sequencer”** V.Biliotti [http://www.arcetri.astro.it/irlab/doc/giano/elettronica/tempi\\_sequencer.pdf](http://www.arcetri.astro.it/irlab/doc/giano/elettronica/tempi_sequencer.pdf)
- **“Software structure of server104”** E. Giani, C. Baffa, V.Biliotti [http://www.arcetri.astro.it/pubblicazioni/Reports/10/testi/3\\_10.pdf](http://www.arcetri.astro.it/pubblicazioni/Reports/10/testi/3_10.pdf)
- **“Performances and results of the detector acquisition system of the GIANO spectrometer”** E. Oliva, V. Biliotti, C. Baffa, E. Giani, M. Gonzalez, M. Sozzi, A. Tozzi, L. Origlia [http://www.arcetri.astro.it/irlab/doc/spie\\_12/giano\\_acquisition\\_system\\_8453\\_103.pdf](http://www.arcetri.astro.it/irlab/doc/spie_12/giano_acquisition_system_8453_103.pdf)
- **“Tabelle dei valori e degli indirizzi dell'elettronica di Giano Versione 1.24”**  
C.Baffa, V.Biliotti, E.Giani [http://www.arcetri.astro.it/irlab/doc/giano/elettronica/Protocollo\\_ISA\\_1\\_24.pdf](http://www.arcetri.astro.it/irlab/doc/giano/elettronica/Protocollo_ISA_1_24.pdf)
- **“Il programma server104 ed il sistema embedded”** C. Baffa, V. Biliotti, E. Giani [http://www.arcetri.astro.it/pubblicazioni/Reports/10/testi/2\\_10.pdf](http://www.arcetri.astro.it/pubblicazioni/Reports/10/testi/2_10.pdf)