

**INFN-BARI
DICEMBRE 1995**

DESCRIZIONE DELLE FUNZIONI DEL NUOVO ASTROS.

F. LODDO, G. MAGGI, A. RANIERI, F. RUGGIERI

INFN-BARI

DESCRIZIONE DELLE FUNZIONI DEL CHIP PER IL NUOVO ASTROS E REGISTRI PRESENTI SUL MODULO,AD ESSO COLLEGATI.

•Introduzione

- | | |
|------------------------------------------------------|---------|
| 1. Specifiche del nuovo ASTROS | pag.iii |
| 2. Segnali di input output sul front panel di ASTROS | pag.iv |
| 3. Formato indirizzo del cluster | pag.iv |
| 3. Interazione tra l'ASTROS e il TSR | pag.v |

•Capitolo I

Principali componenti e loro funzioni del chip per il nuovo Astros

- | | |
|------------------------------------|-------|
| A. Memoria di riordino | pag.1 |
| B. Memoria di acquisizione (SHIM) | pag.3 |
| C. Circuito di gestione del Marker | pag.3 |
| D. Registro di TEST | pag.4 |
| E. FIFO | pag.4 |

•Capitolo II

Registri e contatori del modulo Astros

- | | |
|--------------------------------------------------------------|--------|
| A. Registro degli indirizzi (Next Transfer Address Register) | pag.8 |
| B. Clustering Counter | pag.9 |
| C. Mark Register + Pattern Register | pag.10 |
| D. Chain Length Register | pag.10 |
| E. Strip Counter Register | pag.10 |
| F. Mask Register | pag.11 |
| F1. CSR ID | pag.11 |
| F2. Delay Register | pag.11 |
| G. Segnali di comunicazione tra ASTROS e il processore | pag.12 |
| H. BUSY | pag.12 |
| I. LOAD | pag.12 |
| L. TRIGGER | pag.12 |
| M. EMPTY0, EMPTY1 | pag.12 |

•Capitolo III

Specifiche dei segnali necessari al protocollo di gestione del chip per il nuovo ASTROS

- | | |
|-------------------------------------------|--------|
| A. CLR | pag.13 |
| B. SHCK | pag.14 |
| C. RW_MEM | pag.15 |
| D. FBOP_ONMEM | pag.15 |
| E. DATA_RDWR | pag.15 |
| F. CHIP_SEL, TEST_REG_SEL, MARKER_REG_SEL | pag.15 |
| G. REORDER | pag.16 |
| H. DATA<12:0> | pag.16 |
| I. ADD<7:0> | pag.16 |

J. TRIG	pag.16
K. SERIALIN	pag.17
L. CKCLUST	pag.17
M.FIFOSEL	pag.17
N. EMPTY0	pag.18
O. EMPTY0_DEC_OUT	pag.18
P. READ_FIFO	pag.18
Q. EQUAL0	pag.18
R. TEST4_CHAN	pag.18
S. START_CLUST	pag.18
T. ENDWRITE	pag.19
U. WRITE_FIFO0	pag.19

•Capitolo IV

Control and Status Register del modulo ASTROS	pag.20
Specifiche dei bit del CSR0 di ASTROS	pag.21

A. CLR (bit 31)	pag.21
B. CLR (bit 30)	pag.21
C. CSR0<15>	pag.21
D. CSR0<16-31>	pag.21

Specifiche dei bit del CSR1800 pag.22

FBOP_ONMEM	pag.22
REORDER	pag.22
FIFOSEL	pag.22
PARALLEL_WRITE	pag.22
TEST_SPLITTER	pag.22
TEST4_CHAN	pag.22
TEST_MOD	pag.23
TEST_CHAIN	pag.23
LAYER_COUNT	pag.23
TRIG_SOFT	pag.23
BIT DI CLEAR	pag.23
BIT DI STATO	pag.23
BUSY	pag.23
EMPTY BITS	pag.23

NOTA ESPLICATIVA SUL MODO DI SELEZIONARE I REGISTRI INTERNI AL CHIP pag.26

• Figure Caption	pag.27
• References	pag.27

INTRODUZIONE

Specifiche del nuovo Astros.

Il nuovo Astros svolge tutte le funzioni del vecchio, ed inoltre

e' possibile effettuare il test della funzionalità dei singoli blocchi.

Il sistema attuale possiede una procedura di test che consente di verificare il funzionamento delle catene scrivendo un pattern in esse.

Nel nuovo sistema si e' implementato un sistema di test che consente di verificare ogni singolo elemento dell'acquisizione: l'Astros da solo, la connessione Astros-Splitter Board, ed infine le catene.

E' migliorata la procedura di lettura degli ASTROS in maniera da minimizzare i tempi di trasferimento dati.

Con il vecchio modulo è richiesta una inizializzazione per ogni sezione di Astros che ha dati da trasmettere, più un'operazione di T-pin per individuare quegli Astros che hanno dati da trasmettere. Questo over-head è molto pesante.

Il nuovo modulo contiene 4 sezioni corrispondenti a 2 vecchi moduli e la verifica del T-pin non viene più effettuata, ma vien fatta partire un'operazione di lettura Block-Transfer su ogni modulo, non appena viene rimosso il segnale di BUSY generale del crate.

Il riordino delle strips viene fatto direttamente ad hardware anziché a software come avviene attualmente.

Nel vecchio sistema in alcuni casi particolari non è stato possibile mantenere la corrispondenza tra la posizione fisica delle strips e la loro posizione all'interno della catena elettronica. Siccome è quest'ultima quella che viene determinata da ASTROS, per passare alla corretta posizione geometrica occorre correggere le informazioni determinate dall'elettronica di read-out. Questa correzione attualmente viene fatta dal processore.

La trasformazione tra posizione elettronica e posizione fisica viene effettuata direttamente all'interno del nuovo modulo.

Le catene con tutti i canali settati saranno soppresse.

Le catene non contenenti informazioni, quindi quelle che hanno tutti i canali settati, o tutti i canali a zero, non devono produrre dati da trasferire sul bus. Questo avviene per default per le catene con tutti i canali a zero. Questa condizione viene invece forzata ora per le catene con tutti i canali settati.

Le strip rumorose saranno soppresse direttamente ad hardware.

Attualmente non è prevista alcuna soppressione di strip rumorose se non molto in avanti in fase di ricostruzione. In realtà gran parte di queste strips rumorose sono dovute a schede inaccessibili, che pertanto non possono essere sostituite, o a schede che si rompono durante il run e che potranno essere sostituite solo durante le fasi di apertura.

Bisogna pertanto convivere durante il run con queste strips rumorose. La loro presenza allunga i tempi sia di acquisizione dati che di ricostruzione.

Queste strips vengono ora direttamente soppresse all'interno del nuovo Astros.

Sara' possibile controllare con continuità la sincronizzazione tra le diverse catene gestite dalla stessa splitter board.

Nel sistema attuale 24 catene, connesse alla stessa Splitter Board, vengono shiftate contemporaneamente ma non c'è la possibilità di controllare se le catene rispondono tutte allo stesso modo se non facendo un apposito Test.

Nel nuovo sistema sarà possibile controllare con continuità, anche durante la presa dati, la sincronizzazione tra le diverse catene.

Segnali di input output sul front panel di Astros.

Input:

Load (ECL,busable)

Il segnale di Load viene passato tale e quale alle Splitter Board e quindi all'elettronica di front end. E' necessario minimizzare il tempo di transito del Load attraverso l'Astros. Il ritardo di trasmissione del LOAD, viene comunque controllato a software all'interno del nuovo modulo, per tener conto delle differenti lunghezze dei cavi.

Trigger (ECL,Busable)

Questo segnale viene usato per far partire lo shifting delle catene.

Output:

Empty bit (section 1) (ECL)

Empty bit (section 2) (ECL)

La presenza di questo segnale indicherà che il trasferimento da questo modulo è completato. Esso potrà essere pertanto usato come Enable Transfer per il modulo successivo.

Busy (ECL)

Questa uscita viene settata con l'arrivo del segnale di Level2y e resettata al completamento della clusterizzazione.

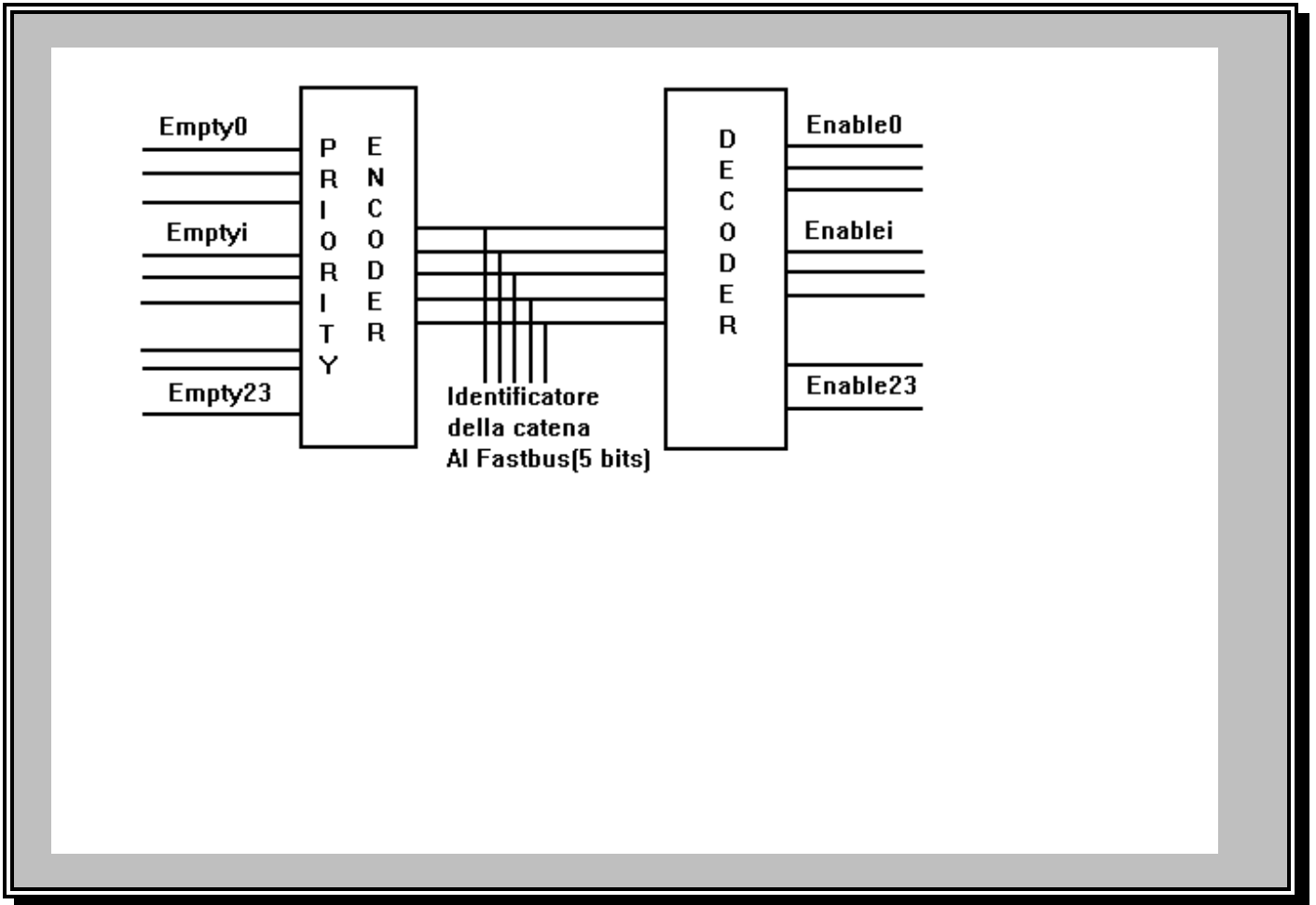
Le uscite di Busy di tutti gli Astros presenti nel crate sono connesse in Wired-Or sulla stessa linea: pertanto la linea risulta settata se almeno una delle uscite Busy è settata.

Formato indirizzo del cluster.

1° byte	posizione dell'ultima strip del cluster nella catena (0÷255).
2° byte	bits 0÷3
bit 7	on
	lunghezza del cluster (1÷15)
	errore di allineamento (marker check)
	off
3° byte	bits 0÷4
4° byte	Identificatore dell'Astros
	allineamento o.k.
	Identificatore della catena.

Per quel che riguarda l'identificatore della catena questo viene formato inviando tutte le uscite empty dei 24 canali di una sezione, ad un circuito di encoder-decoder che abiliterà una alla volta tutte le FIFO non empty al trasferimento dei loro dati sul bus. A tal punto l'encoder conterra l'indirizzo della FIFO interessata che sarà anch'esso trasferito sul bus insieme con i dati della FIFO (vedi formato dei dati).

Nella figura seguente è indicato lo schema di principio di come viene realizzata la gestione dell'informazione del layer.



Interazione tra l'Astros ed il TSR.

Il TSR dopo un EGBX inviera', con l'opportuno timing, il segnale di Load agli Astros.

Il Busy del TSR (BUSY_TSR) viene settato non appena giunge il segnale di ATTN.

Con l'arrivo del segnale di ABORT, il BUSY_TSR puo' essere rimosso senza alcuna limitazione dall'elettronica specifica del sottorivelatore.

Il segnale di TRIGGER viene inviato agli ASTROS che a questo punto iniziano lo shifting.

Gli Astros all'arrivo del segnale di TRIGGER setteranno la linea di ASTROS_Busy. Questo segnale sara' resettato alla fine dello shifting e della clusterizzazione. La linea sara' connessa in wired-or, pertanto se essa e' settata significa che c'e' almeno un Astros che non ha completato lo shifting e la clusterizzazione.

[Il ROC (Read Out Controller: FIC processor), in risposta alla SR generata dal segnale di TRIGGER, dopo aver letto le informazioni relative al trigger dall'SR, controllera' lo stato della linea di ASTROS_Busy:

se la linea e' settata aspettera':

- ⇒ o che essa venga resettata
- ⇒ o un time-out

quando la linea e' resettata procedera' :

- ⇒ alla lettura del WCR
- ⇒ allo swapping delle FIFO negli ASTROS
- ⇒ alla rimozione del BUSY_TSR

⇒ alla lettura delle FIFO degli ASTROS. La lettura dovrà essere effettuata allo stesso modo di come viene fatta attualmente: prima la sezione 1 poi la sezione 2 e così via. In questo modo potrebbero anche essere conservate sia le Astros I.D. attuali : pari per le camere dei mu e dispari per il calorimetro che la cablatura dei cavi sul pannello frontale degli Astros. Si noti comunque che ci saranno due letture indipendenti.

In questo schema è lasciato al ROC il compito di gestire il BUSY_TSR. Deve essere il ROC che deve essere consapevole in risposta ad una service request (SR) se ha completato la lettura dell'evento precedente. Se sono attive due SR, di cui la seconda è in coda, il ROC non abiliterà un'ulteriore Trigger e pertanto non effettuerà lo swap delle FIFO degli ASTROS.

Se la lettura dell'evento precedente non è stata completata, la routine di servizio della SR, dopo aver letto le informazioni relative al trigger setterà un flag per indicare la presenza dell'evento e tornerà al completamento della lettura dell'evento. Una volta completata la lettura dell'evento precedente, il ROC si occuperà di eseguire la routine di SR vera e propria come descritta all'inizio di questo paragrafo, con il controllo della linea ASTROS_Busy e, conseguentemente, allo swapping delle FIFO, alla rimozione del BUSY_TSR ed alla lettura del nuovo evento. Si noti che il BUSY_TSR verso il Trigger Supervisor rimarrà settato per tutto il tempo fino al completamento della lettura dell'evento precedente.

CAPITOLO I

VERSIONE 6.0

Principali componenti e loro funzioni del chip per il nuovo ASTROS

A. Memoria di Riordino

Il diagramma a blocchi del chip per il nuovo Astros, indicato con tratteggio, insieme all'interfaccia FastBus e ai registri del modulo, e' mostrato in fig.1.

In questo schema la Memoria di Riordino, indicata come STAM (STrip Address Memory), e' una RAM da 256x10 bit il cui compito e' quello di fornire gli indirizzi al buffer di evento, in modo che, laddove e' necessario, venga fatto un Riordino di tutte le strip il cui indirizzo logico non corrisponde al proprio indirizzo fisico sul canale.

Pertanto tale memoria, è caricabile da Fastbus attraverso un'operazione di Block Transfer al momento dell'inizializzazione del modulo, del pattern di indirizzi corretti, secondo lo schema di connessione delle strip del calorimetro, per ogni canale.

La STAM e' leggibile via Fastbus, in modo da consentire in qualsiasi momento una verifica del suo contenuto, qualora l'operazione di Riordino non dovesse svolgersi correttamente. Pertanto si rende necessario un segnale che discrimini tra un'operazione Fastbus e il normale funzionamento di acquisizione. Attualmente a tale scopo, nell'implementazione del chip, vengono usati 6 ingressi per il controllo delle operazioni da effettuarsi sulla STAM come sopra detto, questi sono di seguito elencati:

1.FBOP_ON_MEM

Questo ingresso consente di discriminare tra un'operazione Fastbus esterna di R/W e l'operazione interna di lettura della STAM durante l'acquisizione. Se alto, e' selezionata una qualunque operazione Fastbus sulla STAM. Questo ingresso deve essere comunque tenuto basso, anche nel caso di una lettura Fastbus delle FIFO.

2.RW_MEM

Questo ingresso se alto pone i driver di uscita della STAM in modo lettura, se basso consente la scrittura della STAM. Esso deve comunque rimanere alto in caso d'acquisizione.

Esso viene agganciato alla linea di RD dell'interfaccia Fastbus usata sul modulo, se si vuole effettuare un'operazione Fastbus, **viceversa è forzato a uno, se in acquisizione.**

3.CHIP_SEL

Tramite questo ingresso si abilita, attraverso un Decoder esterno, uno dei 96 chip presenti su di un modulo. Questa operazione corrisponde a fissare il Secondary Address della memoria STAM su cui operare.

4.TEST_REG_SEL, MARKER_REG_SEL

Questi ingressi, se tenuti a zero, indirizzeranno la memoria del chip (SHIM+STAM), altrimenti, negli altri casi descritti, consentiranno l'indirizzamento dei registri interni del circuito. (Vedi Cap III, pargf. F.)

La lunghezza della parola della STAM, deriva dal fatto che si dovranno indirizzare 256 locazioni di memorie del Buffer di Front-End, che si ha bisogno di un bit (il nono), per effettuare la soppressione di strip rumorose, ed infine che l'uso di un decimo bit, indicherà il termine della scrittura nella memoria d'acquisizione qualora la catena ad essa associata e' piu' corta di 256.

Sono previsti due modi di funzionamento dello scanner; un **modo Riordino** in cui il buffer d'evento (di seguito in dicato come SHIM), viene indirizzato tramite la STAM e un **modo senza Riordino**, in cui l'indirizzamento della SHIM, avviene tramite lo Strip Counter (vedi Fig.1).

In caso Riordino, una volta terminata la fase di indirizzamento e scrittura dell'evento nella SHIM, all'interno della STAM la fine catena viene individuata dalla presenza del decimo bit a uno. Dal momento in cui questo bit va a uno, è fatta partire la procedura automatica di esame del Marker (vedi dopo) e l'azzeramento delle rimanenti locazioni della SHIM qualora la catena ad essa associata sia di lunghezza inferiore a 256, mentre l'overflow dello Strip Counter (vedi capitolo II), fa partire la fase di lettura e clusterizzazione.

Pertanto la presenza nella STAM di una parola del tipo 2XX, sta a significare procedura di esame del Marker (decimo bit a uno) e azzeramento della SHIM nelle locazioni il cui indirizzo e' indicato da XX (primi 8 bit della parola).

Questa procedura va avanti fino al raggiungimento della parola 2FF nella STAM.

Il nono bit della parola della STAM, viene usato per abbuiare delle strip o insieme di strip, che nel corso dell'acquisizione dovessero risultare rumorose. Questo bit deve essere posto a 1 dal processore, tramite una riscrittura Fastbus delle parole nella STAM, prima di effettuare un nuovo run.

L'esame del Marker e' automatico qualora uso il Riordino, viceversa nel caso senza Riordino il check del Marker e' possibile se tratto il caso come un falso caso Riordino, cioe' caricando nella STAM gli indirizzi logici in sequenza e non gli indirizzi fisici reali e abilitando il Riordino. Viceversa se tratto il caso senza Riordino senza passare dalla STAM, l'esame del Marker non posso piu' averlo.

Cosa identica vale per la soppressione delle strip rumorose, se voglio che questo funzioni anche nel caso senza Riordino, devo trattare questo caso come un finto Riordino.

E' possibile mascherare automaticamente una catena che risulti completamente a uno a seguito del primo bit bloccato a uno all'interno della catena, e che trasmette questo livello per tutti i 255 impulsi di clock, riempiendo cosi' il buffer di informazione inutile.

In entrambi i casi (Riordino o non Riordino), l'inizio della fase di lettura della SHIM e di clusterizzazione e scrittura nelle FIFO, parte dopo che lo Strip Counter ha contato 257 impulsi (vedi "Evento 1 fine scrittura SHIM..."), anche se la catena e' piu' corta, questo per permettere la sincronizzazione delle procedure di lettura-scrittura, su tutti i canali.

E' quindi il segnale di Overflow di tale contatore ad abilitare, la fase di clusterizzazione e relativa scrittura nelle FIFO.

Ogni STAM presente nello scanner, è indirizzabile da Fastbus, sia in modo Random che in Block Transfer. A tale scopo sul chip sono previste tre linee denominate rispettivamente CHIP_SEL, TEST_REG_SEL e MARKER_REG_SEL, che consentono di abilitare, tramite un Next Transfer Address (NTA) register Fastbus piu' un Decoder esterno, il chip e la STAM entro cui scrivere o leggere. In totale si devono scrivere o leggere, 96 STAM per modulo, corrispondenti a 1 STAM x 24 canali x 4 sezioni di ciascun modulo.

Si e' fatto in modo da vedere tutte le STAM di una sezione di un modulo, come un unico spazio dati, in modo da poter effettuare un' unica operazione di Block Transfer per il riempimento di tutta la memoria disponibile per sezione (vedi capitolo II, paragrafo del Contatore Indirizzi STAM).

B. Memoria di acquisizione (SHift In Memory)

Questa è una Ram da 256x1 bit, che ha funzione di Buffer di Front-End e che viene indirizzata (vedi Fig.1) dalla STAM, nel modo Riordino, dallo Strip Counter, nel modo senza Riordino e dal Clustering Counter (vedi Capitolo II) nella fase di rilettura e clusterizzazione.

La scrittura della RAM avviene utilizzando lo stesso clock di acquisizione SHCK a 1MHz, usato per trasferire l'informazione lungo la catena. La lettura e clusterizzazione avviene con un clock a frequenza piu' alta, in modo da essere la piu' veloce possibile. Infine, la rilettura dell'evento gia' clusterizzato nelle FIFO d'uscita, avviene con la frequenza tipica del Fastbus nel caso di un Block Transfer.

Nel caso di un'acquisizione con Riordino, una volta rivelata la fine della catena, se questa e' inferiore a 256, il resto della SHIM viene riempita con zero.

Vista l'uguale profondita' della SHIM rispetto alla STAM, è possibile leggere l'unico bit della SHIM, quando leggo la STAM via Fastbus e questo viene letto come bit <10> del bus DATA<12:0>, quando il segnale FBOP_ONMEM è bloccato a uno e usando l'NTA register come contatore indirizzi (vedi Cap.II).

C. Circuito di gestione del MARKER

Questo è costituito da uno shift register da 8 bit analogo al Marker Register presente sul modulo, da un contatore anch'esso da 8 bit, da un altro registro da 8 bit che contiene la parola di Marker che viene precaricata all'inizio dell'acquisizione e da un comparatore. Non appena viene rilevata la fine della catena parte il clock allo shift register che dopo otto impulsi contiene la parola di Marker fatta circolare all'interno della catena. Se alla fine di questi 8 impulsi la parola contenuta nello shift register, coincide con quella contenuta nel registro interno, il comparatore da' come uscita, su un flip flop interno, uno zero come convalida dell'avvenuto confronto, altrimenti uno in caso contrario.

La gestione del Marker anche nel caso senza Riordino, e' spiegata in dettaglio nel paragrafo dedicato alla STAM.

Per ogni canale e' pertanto prevista un'uscita che viene letta come 15-esimo bit del bus di uscita delle FIFO di evento, e che rappresenta il bit di confronto tra il marker impostato e quello letto.

La scrittura della parola di Marker nel registro interno su specificato, deve essere effettuata secondo le specifiche Fastbus in uno spazio indirizzi secondario, visto pero' come spazio dati e non di tipo "CSR", opportunamente assegnato ai registri. Una scelta potrebbe essere quella di avere indirizzi separati per ogni Marker Register interno ai chip, per avere la massima liberta' di scelta possibile e questa la si ottiene con lo schema di indirizzamento proposto nel Cap.II paraf. A, in modo da caricare in sequenza, i 96x6 registri.

Un'altra alternativa sarebbe quella di caricare in parallelo con un'unica operazione Fastbus write i 96x6 registri interni di Marker, ovvero di Test, di tutti i moduli presenti nel crate.

Questo lo si ottiene se si fa in modo che l'ingresso di CHIP_SEL ad ogni chip venga forzato a zero, nel momento in cui si scelga di fare il caricamento generale dei registri.

L'opportunita' del caricamento individuale verrebbe lasciata come opzione e le due operazioni verrebbero regolate attraverso un bit del CSR1800_H.

L'impulso di strobe per la scrittura da FastBus del registro e' l'impulso attivo basso di PB_CLK, inviato sull'ingresso DATA_RDWR del chip.

D. Registro di TEST

Questo e' uno shift register a 8 bit interno al chip, che viene usato per testare il canale singolarmente e utile per aumentare la testabilita' stessa del chip durante la fase preliminare di verifica del prototipo. Esso è caricabile in parallelo sfruttando lo stesso bus dati usato per leggere/scrivere le memorie e le FIFO e la sua uscita condivide, con l'uscita delle catene, l'ingresso alla memoria d'acquisizione. Per quanto riguarda l'indirizzabilita' dall'esterno di tale registro, vale lo stesso discorso fatto per il Marker register, cioe' a dire che anche a questo registro viene assegnato un indirizzo nello spazio secondario diverso per ogni chip. Pertanto per il suo indirizzamento vale il discorso fatto nel paraf. I.C e cioe' che vengono decodificati opportunamente i bit ADD<19:15>, cosi' come per l'indirizzamento del Marker Register. Nello stesso tempo si possono caricare in parallelo tutti i registri del modulo attraverso l'opzione fornita da un bit del CSR1800_H.

L'impulso di strobe per la scrittura da FastBus del registro e' l'impulso attivo basso di PB_CLK, inviato sull'ingresso DATA_RDWR del chip.

E. FIFO

Queste sono profonde 128x12 bit e due per canale per poter derandomizzare gli eventi, e la gestione di queste, avviene sotto il controllo del processore che le legge in Block Transfer mode via Fastbus.

La fase di acquisizione e di clusterizzazione di un evento, impiega, alla frequenza di clock prefissata, circa 280us dal segnale di trigger, 256us per l'acquisizione e qualche decina di us per la clusterizzazione, a secondo della frequenza con cui la si effettua. In tutta questa fase l'Astros mantiene accesa la linea di BUSY (vedi Capitolo II).

A seguito del trigger, un modulo esterno che gestisce le informazioni relative al trigger (vedi "TSR Functional Specification" CERN/ECP 9 Oct.1992), invia una Service Request al processore degli Astros che, dopo un certo tempo, serve tale richiesta andando a leggere l'evento gia' pronto nelle FIFO (vedi Fig.2 e 4).

Prima di ogni lettura, il processore, dopo aver controllato che la linea di Busy di tutti gli Astros presenti nel crate sia spenta (questa è una linea su un bus esterno che collega in Wired-Or le linee di Busy di tutti gli Astros), effettua un'operazione di swap delle FIFO (ce ne sono infatti due per canale), abilitando opportunamente la FIFO da leggere, e conservando

quest'informazione in un bit del CSR1800_H (vedi Capitolo IV), in modo da sapere esattamente, quale evento il processore sta leggendo in quel momento.

Durante il tempo di lettura del processore di tutto il crate, il trigger è comunque abilitato, pertanto gli Astros sono in grado di accettare un secondo evento, di clusterizzarlo e porlo nella seconda FIFO disponibile, abilitata anch'essa preventivamente, per la scrittura.

Se cio' si verifica, ci sara' una seconda Service Request, che viene servita non appena il processore ha terminato di leggere il primo evento.

Durante tutta la fase di lettura, il processore risulta busy verso il TSR (vedi specifiche), il quale non accetta un ulteriore terzo evento, finche' il processore non ha terminato di leggere il primo (vedi Fig.2 relativa al timing d'acquisizione e Fig 4 di sintesi delle operazioni di Data Acquisition).

Il servizio della seconda Service Request pendente, avviene con le stesse modalita' della prima e cosi' via.

Le FIFO, una volta scaricate del loro contenuto, emetteranno un segnale di EMPTY, e quello corrispondente all'ultimo canale letto dell'ultima sezione presente nel modulo, **segnalera' la fine dell'operazione di Block Transfer, emettendo un segnale di EMPTY generale della sezione (si hanno infatti quattro segnali di EMPTY, uno per sezione).**

Il clock per la lettura delle FIFO, è inviato sull'ingresso DATA_RDWR e pilotato dal segnale di PB_CLK dell'interfaccia Fastbus.

Il bus d'uscita delle FIFO viene multiplexato con il bus dati delle STAM all'interno del chip e questo per poter gestire un numero inferiore di pin e inoltre su questo stesso bus, trova posto anche un tredicesimo bit, che da' l'informazione relativa all'operazione di check con la parola di Marker.

Nota che all'interno del chip, l'uscita EMPTY (vedi capitolo III) della FIFO del primo canale, abilitano il clock della FIFO del secondo canale e l'uscita EMPTY del secondo canale, abilita il clock del terzo canale sul secondo chip e cosi' via, secondo lo schema proposto di seguito e illustrato in Fig.3. Pertanto il processore fa partire la lettura dell'evento iniziando con la FIFO del primo canale della prima sezione nel modulo e questa lettura prosegue in modo automatico, con quelle dei successivi canali, grazie ai segnali di EMPTY dei vari chip, che a turno abiliteranno in cascata l'ingresso di clock delle FIFO dei canali seguenti e delle sezioni seguenti.

Nota che nell'operazione di Block Transfer Read delle FIFO, a queste viene associato un Base Secondary Address pari 69F00_H (Sezione 1 del modulo), 6BF00_H (Sezione 2), 6DF00_H (sezione 3) e 6FF00_H (sezione 4).

Poiche' ad ogni sezione del modulo viene assegnato un indirizzo, **l'operazione di Block Transfer puo' partire indifferentemente da una qualsiasi di queste sezioni e procedere in sequenza con le sezioni di indirizzo piu' alto**, lasciando quelle con l'indirizzo piu' basso rispetto a quello di partenza, non lette, solo che **in questo caso il modulo non genera il segnale di PB_BL_END_B, di fine operazione e quindi questa ha termine in base al numero di trasferimenti impostati a software, dal comando di Block Transfer.**

E' chiaro che il segnale di EMPTY di una sezione, è visto come il segnale di Enable Read della sezione successiva e quindi bastera' che una sezione non invii questo Enable, a bloccare il Block Transfer e a far si' che il segnale di PB_BL_END_B (segnale che indica la fine dell'operazione di Block Transfer), non venga asserito sul Fastbus.

Nota bene che all'esterno del chip è prevista una logica che consenta di indirizzare opportunamente le FIFO e poiche' ognuna di questa corrisponde a un certo canale, nel trasferimento in memoria del processore viene fornito anche **il terzo byte del layer.**

Un'altra cosa importante da tenere presente, e' che volendo effettuare un'unica operazione di Block Transfer su 4 sezioni di un modulo e che le sezioni pari leggono la parte dell'apparato relativo alle camere a muoni, mentre le sezioni dispari leggono la parte calorimetrica dell'apparato, ad ognuna

delle sezioni viene assegnato un indirizzo distintivo che le identificherà, in modo poi che il software di gestione dei dati letti, sa a quale pezzo dell'apparato assegnare quei dati.

Una nota viene fatta a proposito della gestione del Word Counter e cioè della gestione del numero di parole trasferite durante un Block Transfer.

Com'è noto il CSR5, che è il registro a cui secondo le specifiche Fastbus viene demandata la funzione di gestire il Word Count, è implementato nella vecchia versione di Astros, ma per i motivi su esposti (unico Block Transfer), non può più esserlo con la stessa funzione, nel nuovo modulo.

Volendo conservare comunque il numero di parole effettivamente lette per ogni evento, abbiamo implementato un registro sommatore che grazie a una logica sequenziale sincrona con il clock di clusterizzazione CKCLUST e sfruttando gli impulsi di "Write_Fifo" provenienti dal chip, darà il numero totale di scritture effettuate in tutte le FIFO presenti in un modulo. Ad ogni sezione pertanto, viene assegnato un "word counter" con il proprio indirizzo, il cui valore è indicato nella seguente tabella.

WCR Sezione 1	E1F00H
WCR Sezione 2	E1F01H
WCR Sezione 3	E1F02H
WCR Sezione 4	E1F03H

Questi registri verranno letti con un'operazione di Block Transfer, all'inizio di ogni trasferimento, per fissare il numero di parole da leggere nel trasferimento dati.

E.1 Struttura della parola nella FIFO

Il significato dei bit all'interno della parola nella FIFO è mostrato di seguito:

1^o Byte bit da 0-7 —————> strip-address
 2^o Byte bit da 8-15 —————> cluster size

gli altri due byte che compongono la parola di ASTROS, vengono costruiti sulla Mother Board secondo il solito significato:

3^o Byte bit da 16-23 —————> layer address
 4^o Byte bit da 24-31 —————> module identification.

Per quanto riguarda il 2^o byte, c'è da notare che poichè i bit 12,13 e 14 non sono significativi ai fini della lettura del buffer d'uscita, questi devono essere mascherati a software in fase di lettura. Viceversa questi hanno significato in un'operazione di scrittura.

CAPITOLO I

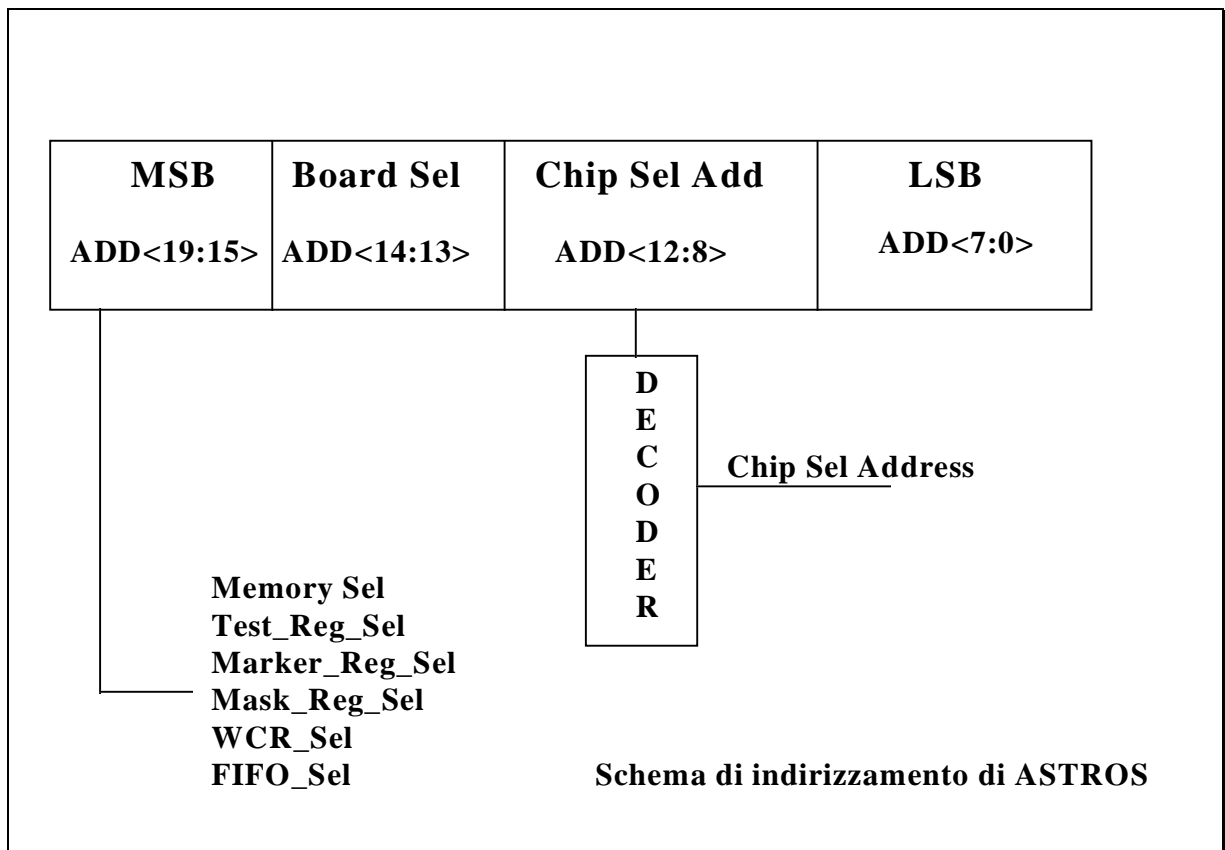
Principali componenti e loro funzioni del chip per il nuovo ASTROS

CAPITOLO II
REGISTRI E CONTATORI DEL MODULO ASTROS

A. Lo spazio indirizzi

Lo spazio indirizzi di ASTROS è uno spazio indirizzi a 20 bit, strutturato nella seguente maniera. I primi 8 bit, i meno significativi (ADD<7:0>), servono a indirizzare le 256 locazioni di memoria STAM usata per il riordino. I bit che vanno dall'ottavo al dodicesimo (ADD<12:8>), sono usati tramite un decoder, per indicare quale tra i 24 chip presenti su ogni sezione del modulo, voglio utilizzare.

I successivi due bit, (ADD<14:13>), vengono usati per selezionare quale sezione indirizzare e infine gli ultimi cinque bit, i più significativi (ADD<19:15>), sono usati per selezionare i vari registri interni ai chip, le FIFO, i Mask Register e i WCR, secondo lo schema di seguito indicato e la cui logica è stata posta all'interno di una PAL dell'ALTERA, del tipo E1800.



ADD<19:15>	<14:13>	<12:8>	<7:0>	REGISTRO	Campo Indirizzi
0 1 0 0 0	0 0	Chip Sel	X X	STAM address 1^	40000-417FF
0 1 0 0 0	0 1	“	X X	STAM “ 2^	42000-437FF
0 1 0 0 0	1 0	“	X X	STAM “ 3^	44000-457FF
0 1 0 0 0	1 1	“	X X	STAM “ 4^	46000-477FF
0 1 0 0 1	0 0	“	0 0	TEST REG 1^	48000-49700
0 1 0 0 1	0 1	“	0 0	“ 2^	4A000-4B700
0 1 0 0 1	1 0	“	0 0	“ 3^	4C000-4D700
0 1 0 0 1	1 1	“	0 0	“ 4^	4E000-4F700
0 1 0 1 0	0 0	“	0 0	MARKER REG 1^	50000-51700
0 1 0 1 0	0 1	“	0 0	“ 2^	52000-53700
0 1 0 1 0	1 0	“	0 0	“ 3^	54000-55700
0 1 0 1 0	1 1	“	0 0	“ 4^	56000-57700
0 1 0 1 1	0 0	1 1 1 1 1	0 0	MASK REG 1^	59F00
0 1 0 1 1	0 1	“	0 0	“ 2^	5BF00
0 1 0 1 1	1 0	“	0 0	“ 3^	5DF00
0 1 0 1 1	1 1	“	0 0	“ 4^	5FF00
1 1 1 0 0	0 0	“	0 0	WCR_REG SEL 1^	E1F00
1 1 1 0 0	0 0	“	0 1	“ 2^	E1F01
1 1 1 0 0	0 0	“	0 2	“ 3^	E1F02
1 1 1 0 0	0 0	“	0 3	“ 4^	E1F03
0 1 1 0 1	0 0	0 X	0 0	FIFO 1^	68000 - 6B000
0 1 1 0 1	1 0	0 “	0 0	“ 2^	6C000 - 6CC00
0 1 1 0 1	1 0	1 “	0 0	“ 3^	6D000 - 6DC00
0 1 1 0 1	1 1	0 “	0 0	“ 4^	6E000 - 6EC00
0 0 0 0 0	0 0	1 1 0 0 0	0 0	GENERAL	1800
0 0 0 0 0	0 0	1 1 0 0 1	0 0	CHAIN LENGTH	1900
0 0 0 0 0	0 0	1 1 0 1 0	0 0	MARK PATTERN	1A00
0 0 0 0 0	0 0	1 1 0 1 1	0 0	IDENTIFICATION	1B00
0 0 0 0 0	0 0	1 1 1 0 0	0 0	DELAY SEZ 1	1C00
0 0 0 0 0	0 0	1 1 1 0 1	0 0	DELAY SEZ 2	1D00
0 0 0 0 0	0 0	1 1 1 1 0	0 0	DELAY SEZ 3	1E00
0 0 0 0 0	0 0	1 1 1 1 1	0 0	DELAY SEZ 4	1F00

B. Clustering Counter

Questo e' un contatore a **8 bit**, che viene attivato, non appena lo Strip Counter ha contato 257 impulsi di clock, il che porta a zero il segnale di START CLUST (vedi Fig.5). Tale contatore viene usato come contatore indirizzi per la lettura della SHIM e relativa clusterizzazione dell'evento nella FIFO. Il primo impulso utile di conteggio per il Clustering Counter, è quello successivo alla transizione negativa di START_CLUST (vedi timing allegato "Evento 1 inizio lettura SHIM..."), pertanto quest'ultimo segnale, va opportunamente sincronizzato con CKCLUST, in modo da rispettare il timing suddetto e questo lo si ottiene con il circuito proposto in Fig. 5.

Il clock a tale contatore (CKCLUST) è ad una frequenza piu' alta rispetto a quello usato per l'acquisizione dalle catene e questo stesso clock viene usato anche come memory enable alla SHIM in questa fase. Tale clock è sempre inviato al Clustering Counter, il quale è

abilitato al conteggio dal segnale di ENDCLUST che coincide con il complemento del segnale di STARTCLUST suddetto.

Il clear a tale contatore, è dato dal segnale di TRIG (vedi CAP.III) durante l'acquisizione ovvero dal CSR0<31>, durante l'inizializzazione.

L'overflow di tale contatore, opportunamente "latchato" e ottenuto dopo **255** impulsi di clock opportunamente ritardato di un ciclo di CKCLUST (vedi "Evento 1 fine lettura SHIM..."), servirà a riportare ad 1 il segnale di STARTCLUST (vedi dopo), disabilitando in tal modo, il driver del contatore e fermando il conteggio dello stesso contatore e la fase di clusterizzazione.

C. Mark Register + Pattern Register

E' un registro a 8 bit circolare scrivibile e leggibile da Fastbus che, quando **in caso Test con Riordino**, contiene la parola di Test piu' quella di Marker, oppure solo il Marker quando in **modo acquisizione con Riordino** o ancora il Pattern di Test, quando **in modo acquisizione senza Riordino o in modo Test senza Riordino**. In ognuno di questi casi, bisogna far attenzione a scrivere nel registro in questione, sempre l'inverso del byte da impostare, in quanto sulle schedine della SGS, l'ingresso Test viene invertito.

In caso Riordino, la parola di Test scritta in tale registro, funge da Marker e pertanto viene trascritta anche nei Marker Register individuali di ogni canale d'acquisizione, presenti nel chip. (vedi Cap II paragrafo "Circuito di gestione del Marker"). Tale parola scritta in questo registro dalla duplice funzione, viene poi fatta circolare all'interno delle catene di shift register dell'apparato attraverso l'ingresso di Test delle schede e shiftata all'interno del Marker Register di canale, sul riconoscimento della fine della catena. Pertanto volendo usare la modalita' Test con il Riordino, si deve aver cura di caricare le STAM in modo tale che queste tengano conto della lunghezza della catena sotto test, cosi come spiegato nel CAP.I paraf. A.

Questo stesso registro deve essere usato come Pattern Register durante il run nel modo senza Riordino, per trasferire uno 0 in coda a tutte le catene, ovvero per trasferire lungo la catena il Test Pattern, quando si opera in modo Test, ma senza Riordino.

Il clock a tale registro viene ricavato invertendo quello di acquisizione (SHCK).

E' prevista la possibilita' di usare una procedura di test che escluda il collegamento dell'Astros dalle catene e che consente di testare il modulo da solo anche se fisicamente collegato alle catene stesse. In tal caso (vedi Fig.1) un MUX sulla scheda provvede, tramite un opportuno bit di controllo all'interno del registro CSR1800_H (TEST_MOD, vedi dopo), a scollegare il PATTERN REGISTER dalle catene e a collegarlo direttamente all'ingresso delle SHIM di ogni canale del modulo in modo che il test di pattern alimenti direttamente l'ingresso di ogni canale dell'Astros.

D. Chain Length Register

E' un registro a **8 bit**, e serve per fissare, insieme al Contatore di Strip la durata dell'acquisizione. E' leggibile/scrivibile via Fastbus. Questo viene usato secondo le modalita' descritte nel paragrafo seguente, nel caso di lunghezza standard d'acquisizione. Vedere piu' in dettaglio la Fig. 5, per la comprensione sull'uso di tale registro.

E. Strip Counter Register

E' un contatore a **9 bit** che contiene normalmente un numero di impulsi di clock, pari a **256+8** e determina a fine conteggio, la fine della fase di lettura delle catene e di scrittura nella SHIM. Il raggiungimento da parte dello STRIP COUNTER di un numero di conteggi pari a 256 impulsi di clock, fa interrompere l'acquisizione dalle catene e fa partire la fase di lettura della SHIM, clusterizzazione e scrittura FIFO cosi' come spiegato nel seguito.

Nel caso normale d'acquisizione, il numero di impulsi di clock che viene mandato alle catene nonche' allo stesso Strip Counter, è pari a **255+8**, per consentire di leggere l'informazione sulle catene piu' l'eventuale informazione del Marker se in caso Riordino, o del Test Pattern se in caso non Riordino. L'impulso TRIG (vedi dopo e Fig.5), porta a zero ENDWRITE (vedi dopo), che abilita il driver del contatore verso il bus indirizzi del chip e fa iniziare la fase di acquisizione, mentre al 256-esimo impulso di SHCK, il segnale di ENDWRITE ritorna a uno, inibendo sia il driver che la scrittura in memoria.

Sul 257-esimo impulso viene inviato sia al chip che alla logica di abilitazione del Clustering Counter un segnale attivo basso di START_CLUST, con cui viene dato avvio alla fase di clusterizzazione che viene regolata dal Clustering Counter come suddetto, mentre alle catene cosi' come allo stesso Strip Counter, vengono ancora inviati, dall'istante dopo in cui si ottiene l'END_WRITE, altri 8 impulsi di clock, per dar modo al byte di Marker di shiftare lungo la catena stessa e di venir quindi analizzato dal circuito del Marker del chip (vedi timing allegato "Evento 1 fine write SHIM e analisi Marker").

Il clock a tale contatore coincide con quello di acquisizione (vedi SHCK nel capitolo III), mentre il clear gli viene fornito dal segnale di (TRIG&CSR0<31>).

Il clock a tale contatore viene sempre fornito, tranne ad abilitarne il conteggio al momento opportuno, secondo lo schema indicato in Fig.5 e comunque sempre un ciclo di clock dopo la transizione negativa di TRIGDEL (vedi dopo) e disabilitato dopo i suddetti **255+8** impulsi di clock (vedi timing "Evento 1 inizio write SHIM...").

Notare che tale contatore è usato sia in modo Riordino che in non Riordino, poiche' in entrambi i casi, esso regola l'acquisizione e la successiva clusterizzazione.

Esso può essere azzerato sia da una scrittura del bit 31 del CSR0 (bit di CLR), che dal segnale di TRIG.

Il confronto tra il contenuto dello Strip Counter e il Chain Length Register viene usato solo ed esclusivamente in fase di Test, per consentire di verificare catene piu' corte di 256. Solo in questo caso si deve porre a 1 il segnale di **TEST_CHAIN** corrispondente a un bit del **CSR1800** dell'ASTROS e si può fissare nel Chain Length Register il numero di impulsi di clock voluti e fissati dalla lunghezza della catena sotto test. Il raggiungimento della lunghezza voluta, produce l'azzeramento del segnale di END_CHAIN, che da quel punto fino al raggiungimento del 256-esimo impulso di clock, riempie di zeri la memoria d'acquisizione SHIM.

F. Mask Register

Questo e' un registro che ha un bit per canale e che maschera i canali corrispondenti a catene che durante l'acquisizione dovessero risultare rumorose. E' un registro sia scrivibile che leggibile da Fastbus. Tale registro può essere usato solo durante una normale acquisizione con le catene e in nessuna fase di test.

Si hanno quattro registri per modulo, uno per sezione.

F1. CSR ID

Questo è un registro grande 6 bit, che contiene il codice identificativo del modulo e che assegna il modulo alla corrispondente parte dell'apparato. Ogni sezione all'interno di un modulo, viene a sua volta identificata attraverso altri due bit che insieme ai 6 bit precedenti, compongono il byte di identificazione del modulo. Questo byte viene letto come il più significativo, all'interno della parola a 32 bit letta dal Fastbus.

F2. DELAY REGISTER

Questi registri sono previsti, per tener conto del differente ritardo con cui il segnale di "LOAD" (vedi dopo), arriva sul rivelatore, a seguito delle differenti lunghezze dei cavi, su cui viene trasmesso. Il loro intervallo temporale varia tra 0 e 100 ns, a step di 10 ns.

NOTA:

Sono previsti anche un CSR0 e un CSR1800_H "user defined" i cui bit vengono specificati di seguito e il CSR7, che specificherà la "Classe di Broadcast" del modulo. Quest'ultimo viene implementato all'interno dell'interfaccia F/B, così come vien fatto per l'NTA register. Fare riferimento alla nota esplicativa sull'interfaccia F/B montata sul modulo (FASP [3]).

G. Segnali di comunicazione tra ASTROS e il processore**H. BUSY (Out)**

Questo segnale viene generato dal modulo e non viene trattato a livello di chip, e va attivo basso sull'inizio della scrittura dell'evento nella SHIM e viene riportato alto non appena conclusa la fase di clusterizzazione e scrittura dell'evento nella FIFO, circa 280us dopo il ricevimento del segnale di trigger e pertanto è portato a uno tramite un bit di "reset" ad esso associato nel CSR1800_H e tramite la transizione 0-1 di STARCLUST a clusterizzazione effettuata, e portato a zero all'inizio di ogni acquisizione dal segnale di TRIG (vedi dopo).

Questo segnale deve essere verificato dal processore, ogni qualvolta questi, dovendo servire una Service Request, inizia la lettura di un evento (vedi Fig.2 e 4).

Il segnale di BUSY inviato al processore, è formato dall'insieme dei BUSY di tutti i moduli presenti all'interno di un crate, attraverso un collegamento Wired-Or su di un bus esterno che collegherà tutti gli Astros di un intero crate con il processore, e la cui durata viene fissata dal tempo totale di scrittura evento e clusterizzazione.

Questo segnale alimenta un bit specifico del CSR1800_H (vedi dopo).

I. LOAD (In)

Il segnale di **LOAD** viene inviato alle catene di shift register presenti sul rivelatore e quindi è passante all'interno del modulo.

L1. TRIGGER (In)

Il segnale di **TRIGGER** viene abilitato opportunamente sulla scheda del processore e pertanto il modulo ASTROS deve riceverlo e trattarlo come spiegato nel par. III.J, ma comunque secondo le specifiche e lo schema indicato in Fig. 5.

CAPITOLO II

Registri e Contatori del Modulo Astros

M. EMPTY0, EMPTY1, EMPTY2, EMPTY3 (Out)

Questi quattro segnali, hanno il significato di "data ready" negato e hanno il compito di segnalare al processore l'avvenuta lettura di un evento all'interno di un modulo, come spiegato nel Cap. IV.

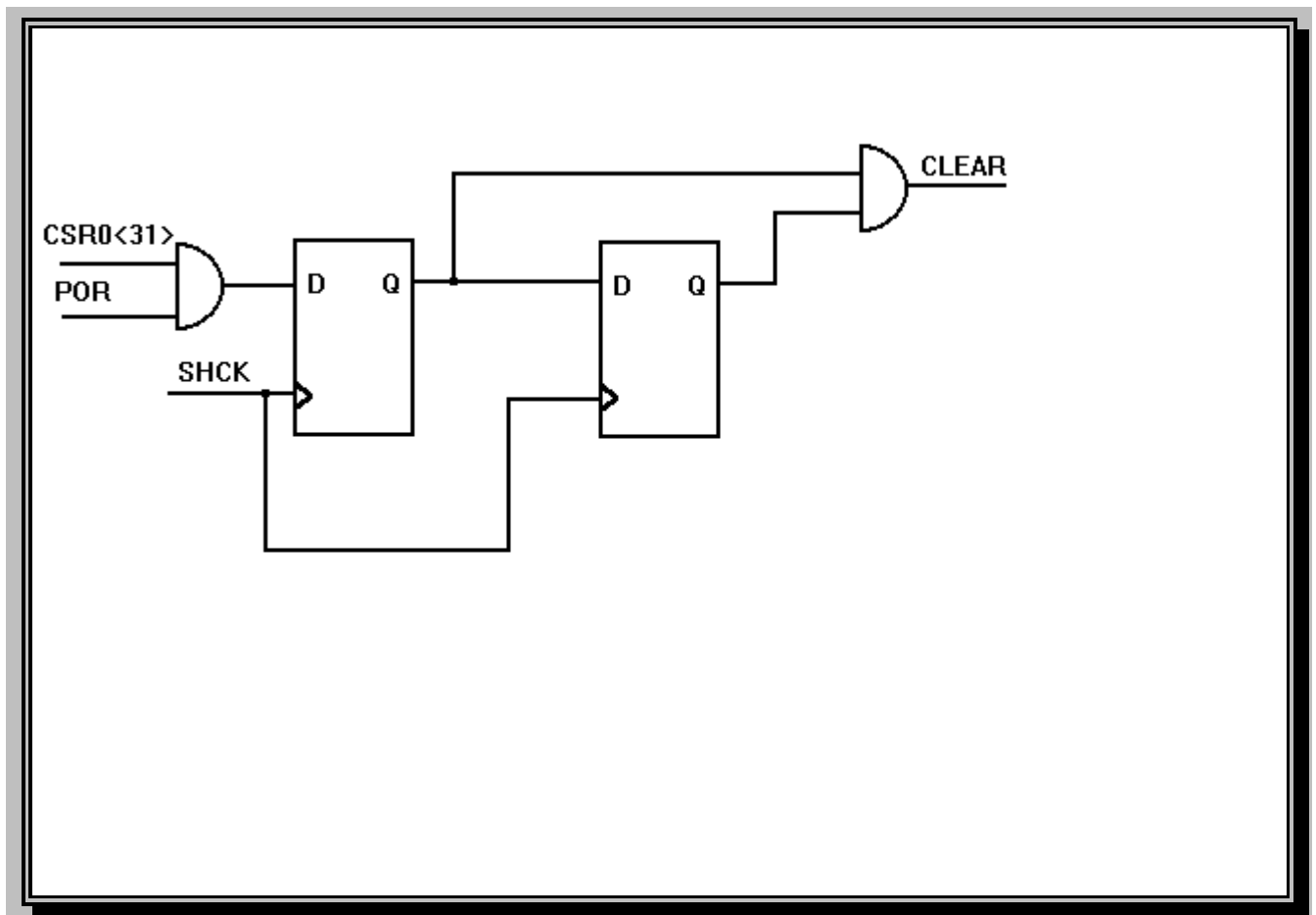
CAPITOLO III
Specifiche dei segnali necessari al protocollo di gestione del chip per il nuovo ASTROS

I segnali necessari alla gestione del chip per il nuovo Astros, sono qui di seguito elencati. Tutti i segnali specificati di seguito, sono in logica TTL.

A. -CLR

Questo e' l'ingresso di CLR della logica interna al chip, e che coincide con l'**AND logico** tra il bit 31 del CSR0, che rappresenta il clear software generale del modulo (attivo basso) e un impulso attivo basso di Power-On-Reset, attivato al momento dell'accensione della scheda. Una considerazione per quanto riguarda la SHIM. E' necessario effettuare un azzeramento di tutte le locazioni SHIM, che corrispondono a strip fisicamente inesistenti e poiche' le catene non sono tutte lunghe 256, questa procedura viene fatta automaticamente in tutti e tre i casi di utilizzo del chip e quindi del modulo. Questo azzeramento si rende necessario poiche', durante la fase di clusterizzazione, vengono lette tutte le 256 locazioni della SHIM, e quelle non interessate, devono contenere zero. In caso Riordino, durante l'acquisizione, tale azzeramento viene effettuato, una volta rivelata dalla logica del chip attraverso il gia' citato decimo bit della parola d'uscita della STAM, la fine della catena in questione. La stessa cosa si verifica anche quando si e' in caso non Riordino, grazie all'azzeramento del Pattern Register sul modulo, che infila quindi zero in coda a tutte le catene ovvero ancora, tramite il segnale di END_CHAIN, che viene prodotto secondo lo schema indicato in Fig. 5, qualora viceversa si voglia procedere con il modo Test su di una catena piu' corta di 256.

Nota che l'impulso di CLR è sempre sincronizzato con entrambi i clock di funzionamento del chip (vedi figura seguente), in modo da fornire comunque un clear sincrono a tutta la logica interna del chip.



B. - SHCK(In)

Questo è il clock di shift dell'informazione sulle catene e pertanto viene anche usato in acquisizione, come impulso di memory enable della SHIM e della STAM. Viceversa il clock per la procedura di clusterizzazione (usato per leggere la SHIM e per scrivere la FIFO) e quello per la lettura delle FIFO, sono diversi e con una frequenza maggiore rispetto a quella di shift, in modo che le due fasi precedenti sono svolte ad una velocità più alta rispetto a quella di acquisizione vera e propria, una volta che l'evento è stato già memorizzato nel Buffer di Front-End.

Questo clock viene sempre inviato allo Strip Counter, che viene abilitato al conteggio grazie al segnale di EN_STRIP come indicato in Fig. 5. Quest'ultimo segnale viene pertanto usato anche per abilitare l'invio di $SHCK$ alle catene, in questo modo il clock parte solo mezzo ciclo dopo dalla generazione dell'impulso di $TRIGDEL$ (vedi timing "Evento 1 inizio write SHIM...") e questo per consentire la necessaria sincronizzazione delle varie parti del circuito, mentre viene spento solo esattamente 264 cicli dopo il primo impulso inviato.

C. -RW MEM(In)

Se tenuto a zero logico consente di usare il bus dati, DATA<12:0>, per scrivere all'interno delle STAM, se a uno logico, consente di leggere il contenuto delle STAM su Fastbus. Questo ingresso viene pertanto collegato alla linea di RD del Fastbus ed è associato ad un livello alto del segnale di FBOP_ONMEM, che abilita contemporaneamente i driver della STAM, in ingresso/uscita verso il Fastbus, qualora si opera con tale memoria. Durante l'acquisizione tale ingresso deve essere comunque tenuto alto, per consentire al bus dati della STAM, di accedere al bus indirizzi della SHIM (vedi Fig.1).

D. -FBOP ONMEM(In)

Se a uno logico consente di effettuare una qualunque transazione Fastbus sulle STAM. Deve essere mantenuto a zero durante l'acquisizione e durante la lettura via Fastbus delle FIFO.

E. -DATA RDWR(In)

Questo ingresso viene usato, in scrittura, come enable della memoria di indirizzo di strip (STrip Address Memory), in lettura, come enable di tutta la RAM compresa la SHIM e viene pilotato dall'uscita PB_CLK del FASIC. Se in acquisizione, la STAM può venire solo letta, e pertanto l'ingresso di memory enable, in questo caso coincide con il clock dell'acquisizione SHCK. Il segnale di FBOP_ONMEM controlla l'operazione da effettuarsi sulla STAM, se a uno logico si può leggere o scrivere da Fastbus tutta la memoria presente nel chip in modo random o block transfer, se a zero non posso effettuare alcuna operazione Fastbus sulla memoria e in questo caso la STAM è in modo acquisizione e pertanto può essere solo letta. La fase di scrittura effettuata da Fastbus, viene eseguita per il precaricamento degli indirizzi di Riordino, la seconda fase di lettura, effettuata durante l'acquisizione, parte non appena ricevuto il segnale di Trigger (vedi dopo).

Questo stesso ingresso viene usato sia come strobe per il caricamento dei registri interni del chip (Test Register e Marker Register), sia come ingresso di clock per la lettura delle FIFO su Fastbus. In quest'ultimo caso, il segnale di FBOP_ONMEM è a zero e c'è un segnale esterno che abilita i driver di uscita delle FIFO verso il Fastbus (vedi capitolo II, paragrafo della FIFO).

F. -CHIP SEL, TEST REG SEL, MARKER REG SEL(In)

Questi sono tre ingressi pilotati secondo lo schema di connessione dell'NTA register, come indicato nel capitolo II e usati per l'abilitazione della RAM (STAM+SHIM) e dei registri interni del chip (Test register e Marker Register) e usati secondo il seguente schema:

- | | | |
|---|-------------------------------------------------|------------------------|
| • | ADD<19>,ADD<18>,ADD<17>,ADD<16>,ADD<15> = 01000 | RAM selezionata |
| • | ADD<19>,ADD<18>,ADD<17>,ADD<16>,ADD<15> = 01001 | Test Reg selezionato |
| • | ADD<19>,ADD<18>,ADD<17>,ADD<16>,ADD<15> = 01010 | Marker Reg selezionato |

L'operazione di caricamento delle STAM, viene effettuata via Fastbus e il processore fa i necessari trasferimenti di "Data Write" in Block Transfer, fino al totale riempimento delle STAM di Astros.

G. - REORDER(In)

Questo è il segnale d'ingresso che abilita o meno tutte le STAM, al fine di usare la procedura di Riordino durante l'acquisizione. Viene controllato da un bit del CSR0 esterno e se a uno logico l'operazione di Riordino viene abilitata. Esso quando è a zero, fa sì che il controllo del bus indirizzi della SHIM, venga assunto dallo Strip Counter (vedi Fig.1).

H. - DATA<12:0>(In/Out)

Questo e' un bus dati interno condiviso dalle STAM e dalle FIFO. Questo viene usato in lettura/scrittura dal Fastbus, se accedo alla STAM, in sola lettura del bus d'uscita delle FIFO, quando leggo l'evento gia' clusterizzato verso il processore.

Per quanto riguarda le FIFO, i 24 bus di tutti i canali, sono posti in three-state e abilitati a turno dalla logica di lettura delle FIFO su Astros, attraverso il segnale di FIFOSEL (vedi dopo), piu' il segnale di indirizzamento della FIFO, ricavato dall'uscita EMPTY, canale per canale (vedi capitolo II). Il 15-esimo bit contiene l'informazione del Marker come spiegato in seguito.

I. - ADD<7:0>

Questo e' il bus indirizzi usato per caricare dall'NTA register, secondo lo schema illustrato nel paragrafo dedicato al Registro Indirizzi del Capitolo II, gli indirizzi della STAM. All'interno poi questo stesso bus, viene condiviso dagli 8 bit dello Strip Counter (Fig.1), e usato da quest'ultimo per fornire gli indirizzi alla SHIM, nel caso di un'acquisizione senza Riordino ovvero dal contatore di Cluster, quando il chip è in fase di lettura SHIM e scrittura FIFO.

Le uscite dell'NTA, dello Strip Counter e del Clustering Counter, vengono multiplexate (nel Gate Array denominato ASTRIX), per avere un maggior controllo degli indirizzi e secondo le modalita' indicate nella seguente tabella.

FBOP_ONMEM = 1	==> NTA register abilitato
FBOP_ONMEM = 0 e END_WRITE = 0	==> NTA register disabilitato e Strip Counter abilitato
FBOP_ONMEM = 0 e END_WRITE = 1	==> Strip Counter disabilitato e Clustering Counter abilitato

J. - TRIGDEL(In)

Un impulso di **TRIGGER**, asincrono e di durata minima pari a due cicli di clock di acquisizione (SHCK) e inviato dal modulo che nella catena d'acquisizione globale, decide se accettare o meno l'evento (vedi nota "TSR Functional Specifications" CERN/ECP Oct.1992 [1]), viene utilizzato per generare sul modulo Astros un altro impulso negativo, **TRIG**, di durata pari a mezzo ciclo di SHCK (vedi Fig 5). Quest'ultimo impulso viene utilizzato per resettare lo STRIP COUNTER e il CLUSTERING COUNTER e inoltre, sempre all'interno del modulo, per generare un impulso negativo **TRIGDEL**, di durata pari questa volta, ad un ciclo di SHCK, ma sfasato di mezzo ciclo dello stesso clock rispetto a **TRIG**.

TRIGDEL, inviato come input a tutti i chip del modulo, avvia la fase di memorizzazione dell'evento (lettura delle catene) ed arriva esattamente nello stesso istante a tutti i chip, pertanto si è fatto in modo che l'impulso di **TRIGDEL** non arrivi agli ingressi dei chip, con uno skew-time superiore di qualche decina di nsec, per essere abbastanza conservativi. Questo stesso impulso come indicato in Fig. 5, così come per l'abilitazione dello Strip Counter come già detto, abilita il clock SHCK alle catene del front-end rispettando il timing suddetto.

L'impulso di **TRIGGER** insieme a quello di **TRIGDEL**, viene inviato ad Astros solo dopo che la condizione di BUSY e' stata rimossa e comunque sempreche' il processore non ne abbia inibito l'invio a seguito di un secondo trigger ricevuto e già elaborato da Astros, mentre esso sta ancora leggendo quello associato al trigger precedente (vedi Fig. 2).

Un altro impulso di **TRIGGER** a solo scopo di test, viene ricavato come bit 6 del CSR1800 (**TRIG_SOFT**) e inviato al modulo, al fine di far partire un'acquisizione.

K. - SERIALIN0 (In)

Questo e' l'ingresso su cui viaggia l'informazione delle catene.

Questo può essere mascherato tramite un Mask Register, che viene indirizzato e precaricato via Fastbus al momento dell'inizializzazione di tutto il modulo Astros, nel caso quel canale debba venir spento durante l'acquisizione.

Una facility in piu' prevista rispetto al vecchio modulo (vedi articolo "The Aleph Hadron Calorimeter..." [2]), e' quella di fornire l'ingresso di ogni canale, di uno shift register precaricabile in parallelo con un pattern ciclico da 0 a 255.

Questo oltre a consentire in fase di simulazione un test esaustivo e automatico circa il comportamento del chip, dara' anche la possibilita' in fase di test effettivo del modulo, di analizzare piu' da vicino il comportamento di ogni singolo canale.

E' da notare che sul modulo Astros e' previsto un circuito che abilita il clock alle catene in modo tale che questo parta esattamente, mezzo ciclo dopo dal momento in cui viene generato l'impulso di TRIGDEL su descritto. Questa stessa abilitazione si deve usare anche per abilitare il conteggio dello Strip Counter. Questo circuito di sincronizzazione di tutte le operazione e' indicato in Fig. 5 e in fase di simulazione del chip, simulato con Verilog HDL, un linguaggio di Hardware Description , con cui e' stato descritto e simulato il funzionamento dell'intero chip.

L. -CKCLUST (In)

Ingresso di clock usato nella fase di clusterizzazione.

Questo ingresso viene pilotato da un oscillatore ad una frequenza piu' alta di quella usata durante l'acquisizione dalle catene.

Tale clock, cosi' come nel caso di SHCK per lo Strip Counter, viene sempre inviato sull'ingresso di clock del Clustering Counter e il contatore viene abilitato al conteggio solo un ciclo dopo la transizione negativa del segnale di START_CLUSTER (vedi dopo) come e' mostrato nel timing "Evento 1 inizio lettura..." e disabilitato, dallo stesso contatore, dopo 256 cicli di clock (vedi "Evento 1 fine lettura...").

M. -FIFOSEL(In)

Questo ingresso viene pilotato dal corrispondente bit del CSR1800_H e serve a indirizzare una delle due FIFO presenti per canale abilitando, l'ingresso di clock di lettura, di quella contenente l'evento da leggere e abilitando l'ingresso di clock di scrittura di quella libera ad accettare il successivo evento. Pertanto FIFOSEL a zero significhera' abilitare in scrittura la FIFO0 e in lettura la FIFO1 di ogni canale, FIFOSEL a uno indichera' esattamente la condizione opposta.

In scrittura questo bit a uno fa cambiare stato a un T flip-flop, quindi ogni volta che bisogna procedere allo swap delle FIFO, basta che il processore scriva uno in questo bit, viceversa lo stato di tale flip-flop viene riletto come bit di FIFOSEL.

N. EMPTY0 (Out)

Ogni chip fornisce un segnale di EMPTY **attivo alto**, che segnala quando la FIFO sottoposta a lettura, non ha piu' dati da trasferire.

O. EMPTY0 DEC OUT (In)

E' un ingresso **attivo basso** alimentato dall'uscita dell'Encoder-Decoder (gate array ENCDEC) del chip, che viene usato per abilitare la FIFO che viene letta, secondo il principio indicato nel Cap. I paragrafo relativo alle FIFO (vedi Fig 3).

P. READ FIFO

Ingresso usato per abilitare la lettura delle FIFO insieme ai due ingressi succitati e che alimenta un flip-flop interno che viene posto a uno da questo ingresso pilotato con un **impulso attivo basso**.

Questo impulso viene ricavato dal corrispondente bit del CSR1800 e viene portato a uno, prima di iniziare la lettura dello spazio dati (delle FIFO), del modulo.

Q. -EQUAL0 (Out)

Non e' un pin di ingresso/uscita del chip, ma e' citato qui perche', se a zero, indica l'avvenuto riconoscimento della parola di Marker. Questo segnale fa parte del bus dati d'uscita (15-esimo bit aggiunto al bus dati interno DATA<12:0>) e informa circa il corretto trasferimento del Marker, lungo tutta la catena.

Questo bit viene posizionato come bit 15, all'interno della parola a 32 bit da trasferire al processore, in modo da riconoscerne il significato.

R. -TEST4 CHAN(In)

Questo ingresso **attivo alto**, multiplexa gli ingressi SHIN, tra l'ingresso seriale della catena e l'uscita seriale del Test Register interno a scopo di test del canale. Questo ingresso e' collegato al corrispondente bit del **CSR1800** (vedi capitolo IV).

S. -START CLUST

Questo ingresso va a zero sul **257-esimo** impulso di SHCK, a partire dal primo impulso di clock utile inviato allo Strip Counter e alle catene. A tale istante esso abilita il driver del Clustering Counter e fa partire la fase di clusterizzazione (vedi Fig. 5).

START_CLUST torna a 1 con il segnale che il Contatore di Cluster emette dopo che esso ha contato **255** impulsi di CKCLUST, interrompendo la fase di lettura della SHIM e la clusterizzazione e inibendo al conteggio il contatore stesso (vedi "Evento 1 fine lettura SHIM..." e Fig. 5).

T. -ENDWRITE (In)

E' un ingresso, proveniente direttamente dallo Strip Counter dopo che questi ha contato **255** impulsi di clock ma ritardato di un ciclo rispetto al 255-esimo impulso e che serve a disabilitare il clock alla STAM e alla SHIM terminata la fase di scrittura dell'evento in quest'ultima e ad esaminare, in fase Riordino, il contenuto del Marker (vedi Fig.1e Fig. 5).

L'ENDWRITE viene "latchato" opportunamente, in modo da mantenere disabilitato il driver che pilota il bus indirizzi del chip, attraverso lo Strip Counter e riabilitato con il Trigger, che riazzera anche lo Strip Counter e il Clustering Counter. Esso è attivo basso durante l'acquisizione.

U. WRITE FIFO0 (Out)

Questa uscita coincide con l'impulso **attivo alto** usato per l'operazione di write delle FIFO all'interno del chip. Questa uscita, attraverso un'opportuna logica sequenziale sincrona con il clock di clusterizzazione posta sul modulo, serve a ricavare il numero di parole complessivamente scritte all'interno di un modulo. Attraverso un circuito di campionamento di questo segnale e tramite dei sommatore, si può ricavare un word count, caricato nei "Word Counter" individuali di ogni sezione del modulo, a ognuno del quale viene attribuito uno specifico indirizzo di CSR sulFastbus, utile al fine di verificare, a fine Block Transfer, il numero di parole trasferite con quelle realmente scritte per ogni evento, nelle FIFO del modulo.

CAPITOLO III

Specifiche dei Segnali necessari al protocollo di gestione del chip per il nuovo Astros

Sono previsti inoltre

BIST TEST

Ingresso di Test per l'opzione "Built-In Self Test" delle memorie.

BIST CK

Ingresso di Clock per l'opzione "Built-In Self Test" delle memorie.

BIST RES

Saranno 4 pin d'uscita per il risultato del BIST, uno per ogni elemento di memoria previsto.

POWER

Saranno in totale 8 pin, tra alimentazione e massa.

CAPITOLO IV
CONTROL AND STATUS REGISTER DEL MODULO ASTROS

In totale avremo 8 registri di tipo CSR sul modulo a cui viene assegnato, nello spazio indirizzi riservato ai CSR, un indirizzo appropriato, secondo le specifiche del Fastbus.

I CSR, sono di seguito elencati:

- CSR0 User Defined Control Status Register
- CSR3 Logical Address Register,
- CSR7 registro che fissa la "Classe di Broadcast" del modulo.
- CSR1800_H User Defined Control Status Register
- CSR1900_H Chain Length Register.
- CSR1A00_H Marker+Pattern Register
- CSR1B00_H Registro di Identificazione della Sezione di ASTROS.
- CSR1C00_H Delay Register Sezione 1
- CSR1D00_H Delay Register Sezione 2
- CSR1E00_H Delay Register Sezione 3
- CSR1F00_H Delay Register Sezione 4

Specifiche dei bit del CSR0 di ASTROS

BIT DI CONTROLLO

- CLR (BIT 31)** Bit di clear generale del modulo (**attivo alto**). L'attivazione di questo bit determina l'azzeramento di tutti i contatori del modulo e della logica interna del chip e l'azzeramento di tutti i bit che compongono il CSR0, anche se poi ogni bit sarà del tipo "set and clear", secondo le specifiche FastBus, inoltre l'operazione di clear può essere effettuata oltre che a software attraverso questo bit, anche attraverso il segnale hardware di Power On Reset (POR), al momento dell'accensione del modulo. Questo azzeramento di tipo software, può effettuarsi anche tramite un'operazione Broadcast. Il CLR comunque non viene attivato qualora risulta attiva un'acquisizione (BUSY Astros attivo). Questo bit coincide con l'uscita denominata USR6_CLR_DATA della scheda FASP (FAstbus Slave Piggy-back sub-card), usata come interfaccia al Fastbus. (Vedi Ref. 4).
- CLR (BIT 30)** Reset generale del modulo, che genera un impulso positivo (**attivo alto**), pari a due periodi di clock, quando questo bit viene scritto. Coincide con l'uscita USR5_M_RESET, della scheda FASP.
- CSR0<15>** Module active bit e può venire solo letto. Coincide con l'uscita USR2_ACTIVE del FASP e viene usato per pilotare il led di indicazione di selezione del modulo.
- CSR0<16.31>** Module ID. Si possono solo leggere, ma sono scrivibili in fase di programmazione del CSR0, dovendo assegnare l'identificatore del modulo.

Specifiche dei bit del CSR1800 di ASTROS

- FBOP_ONMEM** Bit per l'abilitazione delle operazioni Fastbus sulle STAM (**attivo alto**). Va mantenuto basso durante la lettura via Fastbus delle FIFO e durante l'acquisizione.
- REORDER** Indica l'operazione di Riordino effettuato su tutti i canali (**attivo alto**). Questo bit non viene resettato dal bit di CLR del CSR0, ma da un bit di "selective clear" nel CSR1800.
- FIFOSEL** Questo bit abilita, a secondo del suo stato la FIFO da scrivere sul successivo evento, all'interno del chip, pertanto restituisce in ogni istante, lo stato circa la FIFO indirizzata in quel momento. Inoltre questa operazione consente di porre sugli ingressi del Priority Encoder usato per la lettura del layer, citato nel paragrafo dedicato alla FIFO del capitolo I, le uscite di EMTPY delle FIFO da leggere. Il bit quando a zero, indica un'operazione di scrittura sulla FIFO0 e di lettura sulla FIFO1, viceversa se a uno. Non viene resettato dal bit di CLR del CSR0, ma da un bit di "selective clear" ad esso dedicato.
- PARALLEL_WRITE** Questo bit **se a uno logico**, consente l'indirizzamento unico e in parallelo sia del Marker Register che del Test Register, nel caso di una scrittura F/B per il loro setting, sia nel caso di un indirizzamento Geografico che Broadcast.
- TEST_SPLITTER** Questo bit corrisponde come funzionalita' al bit 9 del CSR0 del vecchio modulo, con il quale si mette in modalita' "Test" la nuova Splitter Board. Pertanto, dovendo corrispondere al bit succitato, ha sul connettore esterno del modulo, la posizione occupata dall'ex bit 9, corrispondente al vecchio "mask register load". Esso è **attivo alto**.
- TEST4_CHAN** Questo bit (**attivo alto**) se a uno, seleziona il modo test del chip (Fig.1). Esso fa circolare una parola precaricata in uno Shift Register interno al chip, che alimenta l'ingresso delle SHIM e consente di verificare quindi il risultato ottenuto dopo l'acquisizione di tale parola di test. **Nel caso in cui questo bit venga selezionato, per effettuare il test descritto, si deve aver cura di caricare zero nel Mask Register.**

- TEST_MOD** Questo bit (**attivo alto**) se a uno, seleziona il modo test di tutto il modulo. Questo consente di collegare gli ingressi SERIALIN del chip al Pattern Register di Astros, controllando un mux esterno (Fig.1). Quando questo bit e' a uno, il bit TEST4_CHAN, deve essere a zero.
- TEST_CHAIN** Questo bit (**attivo alto**) viene portato a uno se si vuole effettuare il test con le catene. Esso viene azzerato da CLR ed è di tipo read/write. Quando a uno gli altri due bit di TEST devono essere a zero.
- LAYER_COUNT** Questo bit (**attivo alto**) occupa sul connettore esterno del nuovo modulo la posizione occupata sul vecchio, dall'ex bit 4 del vecchio CSR10, corrispondente a "preset mask register". La nuova funzione implementata da questo bit sulla nuova Splitter Board, è quella di incrementare un contatore interno alla Splitter, per la selezione del Layer. Questo bit è di tipo write-only.
- TRIG_SOFT** Questo bit a solo scopo di test, serve a generare un impulso di trigger software, per il test dell'acquisizione.

I bit di controllo suddetti sono di tipo read/write secondo le specifiche dei CSR mostrate di seguito, tranne il bit di READ_FIFO e tutti i "selective clear" , il bit di CLR compreso, che saranno solo di tipo write.

BIT DI CLEAR

Sono previsti per tutti i bit del CSR1800, in modo che tutti i bit siano del tipo "set and clear", secondo le specifiche FastBus.

BIT DI STATO

BUSY

Attivo basso, indica lo stato di Busy del modulo. Esso è portato a uno tramite un bit di "reset" ad esso associato e tramite la transizione 0-1 di STARCLUST a clusterizzazione effettuata, e azzerato dal segnale di TRIG.

EMPTY0, EMPTY1, EMPTY2, EMPTY3

Per ogni sezione di Astros, l'uscita Encoder Out (EO) piu' significativa, degli Encoder usati per abilitare a turno le FIFO in lettura secondo lo schema di Fig. 3, alimenta il bit di EMPTY corrispondente per sezione del CSR1800 , campionato dal segnale di clock PB_CLK e che indica lo stato effettivo di "Empty" del modulo. Sull'impulso di CLEAR o all'accensione del modulo, questi bit vanno "on" stando a significare appunto lo stato di tutto vuoto del modulo.

CAPITOLO IV

Control and Status Register del modulo ASTROS

Questi segnali, che sono **attivi alti**, corrispondono in un certo senso a un segnale di "data ready" negato e hanno il seguente significato. Se in un'operazione di Block Transfer in lettura delle FIFO, si verifica un "time-out" per qualche motivo legato a un malfunzionamento di un modulo e pertanto non arriva in tempo il segnale di PB_BL_END_B sul Fastbus, il processore, indirizzando geograficamente ogni modulo, può risalire all'offending module andando a leggere i bit di EMPTY e identificando il modulo incriminato, con il primo modulo nel crate, a partire da quello letto per primo nella catena di lettura, il cui bit di EMPTY e' rimasto settato a zero.

L'uscita individuale di EMPTY di ogni FIFO all'interno di un chip, segnala alla logica di lettura esterna sul modulo, che il processore deve leggere il canale successivo, pertanto oltre ad abilitare l'ingresso di clock FIFO del canale successivo, questo segnale fa cambiare l'indirizzo del layer corrispondente al canale da leggere, che rappresenta, all'interno della parola a 32 bit dell'evento, il terzo byte. Le modalita' con cui questo accade, sono descritte nel Capitolo I, nel paragrafo dedicato alla FIFO. Un'operazione di CLEAR effettuata in Broadcast, provvede a spegnere (in realta' a porre a uno logico), i bit di EMPTY di tutti i moduli nel crate.

Ricapitolando la situazione dei **bit del CSR0**, e' la seguente:

<u>Bit con funzioni di READ</u>		<u>Bit con funzioni di WRITE</u>
0	ERROR FLAG	SET ERROR FLAG
1		
14	PARITY ERROR	SET PARITY ERROR
16	MODULE ID (LSB)	CLEAR ERROR FLAG
17	"	
18	"	
19	"	
20	"	
21	"	
22	"	
23	"	
24	"	
25	"	
26	"	
27	"	
28	"	
29	"	
30	"	RESET MODULE
31	"	GENERAL RESET

La situazione dei bit del **CSR1800**, e' la seguente:

<u>Bit con funzioni di READ</u>		<u>Bit con funzioni di WRITE</u>
0	TEST4_CHAN	SET TEST4_CHAN
1	TEST_MOD	SET TEST_MOD
2	TEST_CHAIN	SET TEST_CHAIN
3	TEST_SPLITTER	SET TEST_SPLITTER
4	FB_OP_ONMEM	SET FB_OP_ONMEM
5	REORDER	SET REORDER
6	FIFOSEL	SET FIFOSEL
7	PAR_WRITE	SET PAR_WRITE
8	BUSY	SET_TRIG_SOFT
9	EMPTY0	READ_FIFO0
10	EMPTY1	READ_FIFO1
11	EMPTY2	READ_FIFO2
12	EMPTY3	READ_FIFO3
13		SET_LAYER_COUNT
16		CLEAR TEST4_CHAN
17		CLEAR TEST_MOD
18		CLEAR TEST_CHAIN
19		CLEAR TEST_SPLITTER
20		CLEAR FB_OP_ONMEM
21		CLEAR REORDER
22		CLEAR FIFOSEL
23		CLEAR PAR_WRITE
24		CLEAR BUSY

NOTA ESPLICATIVA SUL MODO DI SELEZIONARE I REGISTRI INTERNI AL CHIP

I registri interni al chip (Test_Register e Marker_register), vengono selezionati attraverso un multiplexer a 2 ingressi e 4 uscite, controllate da un ingresso di selezione, secondo la seguente tabella di stati logici:

S0	S1	Y3	Y2	Y1	Y0	
0	0	0	0	0	1	Memoria selezionata
0	1	0	0	1	0	Nessun registro selezionato
1	0	0	1	0	0	Test Register selezionato
1	1	1	0	0	0	Marker Register selezionato

Pertanto si e' provveduto a realizzare all'interno del registro NTA un circuito equivalente alle seguenti equazioni logiche:

$$S0 = !(AD18*!AD17*!AD16*!AD15)*!(AD18*!AD17*!AD16*AD15)* \\ (AD18*!AD17*AD16*!AD15)$$

$$S1 = (!(AD18*!AD17*!AD16*!AD15)*(AD18*!AD17*!AD16*AD15)) \\ \# (AD18*!AD17*AD16*!AD15)$$

FIGURE CAPTION

1. Diagrammi a blocchi generale di un canale d'acquisizione e interfaccia Fastbus del modulo Astros
2. Timing d'acquisizione
3. Diagramma a blocchi dello schema di decodifica del Layer
4. Diagramma di flusso delle operazioni di acquisizione
5. Schemi di principio per l'utilizzo dei contatori di Astros
6. Schema di abilitazione per la scrittura parallela dei registri interni al chip

REFERENCES

1. "TSR Functional Specifications". Beat Jost CERN/ECP. Draft Version 0.994 of 9 October 1992.
2. "The Aleph hadron calorimeter strip readout scanner". M.G.Catanesi et. al.. Nucl.Instr.Meth. A297 (1990) 390-395.
3. "A programmable high performance integrated Fastbus slave interface chip"
J.Christiansen, C.Ljuslin, CERN/LAA