



**Politecnico
di Torino**

Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Biomedica

A.a. 2020/2021

Sessione di Laurea Luglio 2021

Realizzazione del sistema di acquisizione di un monitor per la misura di fasci di particelle a bassa intensità

Relatore:

Prof. Gianni Coppa

Correlatore:

Dott. Marco Donetti

Candidato:

Davide Iervese

Abstract

La tesi presentata espone la realizzazione di un progetto finalizzato alla costruzione di un monitor per la misura di fasci di particelle aventi bassa intensità. L'idea è nata dalla collaborazione dell'Istituto Nazionale di Fisica Nucleare (INFN) della sezione di Roma e della Fondazione CNAO. La Fondazione CNAO è il luogo dove vengono svolti trattamenti con protoni e ioni di carbonio per la cura di tumori resistenti alla radioterapia. La tecnica utilizzata al CNAO è l'adroterapia, la quale necessita di un sincrotrone per generare fasci di particelle cariche che raggiungano energie molto elevate e di un sistema di somministrazione della dose utile per controllare i fasci ionici, da guidare per distribuire la dose: il Dose Delivery System (DDS). Quest'ultimo è costituito da due rivelatori per il monitoraggio del fascio (BOX1 e BOX2), ossia camere di ionizzazione a piastre parallele con gli anodi segmentati in diversi modi. Le camere si interfacciano con un sistema di controllo la cui intelligenza è governata da cinque FPGA. Il progetto del monitor per la misurazione di fasci a bassa intensità prevede la costruzione, da parte dell'INFN, di un nuovo monitor che verrà inserito nel DDS al posto della BOX1 e della BOX2. L'attività svolta in prima persona al CNAO, descritta nella seguente tesi, è dedicata alla realizzazione del sistema di acquisizione dati del nuovo monitor e alla sua integrazione nel DDS. Il monitor è costituito da due layer di fibre scintillanti, entrambi con 128 fibre. L'informazione viene acquisita utilizzando una scheda elettronica programmabile, la DT5550W / A55CIT4, che contiene un'FPGA e sulla quale è possibile montare due matrici di SiPM (Fotomoltiplicatori al Silicio), ognuna delle quali contiene 64 SiPM.

Il lavoro da me svolto presso la Fondazione CNAO è consistito, in primo luogo, nell'implementare una struttura dell'acquisizione dei dati molto simile a quella utilizzata nel DDS, andando a programmare l'FPGA della DT5550W / A55CIT4. Il software utilizzato per programmare la scheda, SCI-Compiler, permette l'implementazione grafica user-friendly del firmware con la quale programmare l'FPGA. Lo step successivo è basato sulla progettazione, costruzione e test di una scheda elettronica, il cui fine è trasformare il formato dei dati in uscita dalla DT5550W / A55CIT4 da seriale a parallelo: passo necessario per integrare l'acquisizione dei dati con l'elettronica del DDS. La scheda è denominata DD Low Int. Infine, per mettere in comunicazione il sistema costituito dall'insieme delle due schede DT5550W / A55CIT4 e DD Low Int con il DDS, è essenziale la programmazione di un'FPGA del DDS. Per questo compito viene utilizzato il software LabVIEW, tramite il quale è generata anche un'interfaccia per la gestione e la visualizzazione dei dati. Al momento del test finale, non essendo il monitor ancora ultimato, è effettuata un'acquisizione di dati simulati dal firmware della DT5550W / A55CIT4. Dall'esito finale del test, è possibile affermare che è verificato con successo il corretto funzionamento della catena elettronica di acquisizione del rivelatore di fasci di particelle a bassa intensità. Il sistema completo sarà in grado di rilevare fasci di particelle con intensità molto basse che attraversano il monitor a differenza della BOX1 e BOX2, le quali, essendo utilizzate per le terapie dove i fasci hanno intensità molto più alte, non sono in grado di acquisire con la precisione richiesta fasci a basse intensità.

Abstract

The thesis presented exposes the realization of a project aimed at the construction of a monitor for the measurement of low intensity particle beams. The idea was born from the collaboration of the National Institute of Nuclear Physics (INFN) of the Rome section and the CNAO Foundation. The CNAO Foundation is the place where treatments with protons and carbon ions are carried out for the cure of tumors resistant to radiotherapy. The technique used at CNAO is hadrontherapy, which requires a synchrotron to generate beams of charged particles that reach very high energies and a dose administration system useful for controlling the ion beams and guiding them to distribute the dose: the Dose Delivery System (DDS). The latter consists of two detectors for monitoring the beam (BOX1 and BOX2), that is ionization chambers with parallel plates with the anodes segmented in different ways. The cameras interface with a control system whose intelligence is governed by five FPGAs. The project of the monitor for the measurement of low intensity beams involves the construction, by the INFN, of a new monitor that will be inserted in the DDS instead of BOX1 and BOX2. The activity performed in person at CNAO, described in the following thesis, is dedicated to the realization of the data acquisition system of the new monitor and its integration into the DDS. The monitor consists of two layers of scintillating fibers, both with 128 fibers. The information is acquired using a programmable electronic board, the DT5550W / A55CIT4, which contains an FPGA and on which it is possible to mount two SiPM matrices (Silicon Photomultipliers), each of which contains 64 SiPM.

The work I did at the CNAO Foundation consisted, first of all, in implementing a data acquisition structure very similar to that used in the DDS, programming the FPGA of the DT5550W / A55CIT4. The software used to program the board, SCI-Compiler, allows the user-friendly graphic implementation of the firmware with which to program the FPGA. The next step is based on the design, construction and test of an electronic board, the purpose of which is to transform the format of the data output from the DT5550W / A55CIT4 from serial to parallel: a necessary step to integrate data acquisition with the DDS electronics. The board is named DD Low Int. Finally, programming a DDS FPGA is essential to put the system, consisting of the two boards DT5550W / A55CIT4 and DD Low Int, in communication with the DDS. For this task, the LabVIEW software is used, through which an interface for data management and visualization is also generated. At the time of the final test, as the monitor is not yet completed, simulated data is acquired from the firmware of the DT5550W / A55CIT4. From the final result of the test, it can be stated that the correct functioning of the electronic acquisition chain of the low intensity particle beam detector has been successfully verified. The complete system will be able to detect particle beams with very low intensities that cross the monitor unlike BOX1 and BOX2, which, being used for therapies where the beams have much higher intensities, are not able to acquire with the accuracy required low intensity beams.

Indice

Introduzione	1
Capitolo 1. Adroterapia	1
1.1 Interazione radiazione-materia.....	1
1.1.1 Bersaglio sottile.....	3
1.1.2 Bersaglio spesso.....	3
1.2 Interazioni fotoni-materia.....	5
1.2.1 Scattering.....	5
1.2.2 Assorbimento.....	6
1.3 Interazioni particelle cariche-materia.....	8
1.3.1 Scattering Coulombiano.....	9
1.3.2 Stopping Power e profondità di penetrazione.....	10
Capitolo 2. Cenni di progettazione di circuiti elettronici	14
2.1 Interoperabilità tra porte logiche.....	16
2.1.1 Confronto porte logiche TTL – CMOS.....	17
2.2 Porte open-collector.....	21
2.2.1 Pilotaggio di un carico generico.....	22
2.3 Problemi di alimentazione e condensatori di by-pass.....	23
2.4 Linee di trasmissione.....	27
2.4.1 Modello a costanti concentrate.....	28
2.4.2 Propagazione delle onde in una linea di trasmissione.....	28
2.4.3 Linea di trasmissione senza perdite.....	30
2.4.4 Coefficiente di riflessione in tensione.....	30
2.4.5 Impedenza d'ingresso in linea senza perdite.....	32
Capitolo 3. FPGA	34
3.1 Architettura dell'FPGA.....	34
3.2 Risorse hardware di base.....	35
3.2.1 Blocchi logici.....	35
3.2.2 Blocchi I/O.....	36

3.2.3 Risorse di interconnessione.....	36
3.3 Blocchi hardware specializzati.....	38
3.3.1 Blocco di gestione del clock.....	38
3.3.2 Blocchi di memoria.....	39
3.4 Vantaggi delle FPGA.....	40
Capitolo 4. Dose Delivery System.....	41
4.1 Rivelatori di fascio.....	42
4.2 Elettronica delle camere di ionizzazione.....	44
4.2.1 Tera06.....	44
4.2.2 Tera07.....	47
4.3 Sistema di controllo.....	51
4.3.1 Fast Control.....	53
4.3.2 Slow Control.....	54
4.4 Funzioni e operazioni del DDS.....	54
4.4.1 Off - trattamento.....	54
4.4.2 Pre – trattamento.....	55
4.4.3 Erogazione del fascio.....	55
4.4.4 Post – trattamento.....	56
4.4.5 Gestione degli interlock.....	56
4.4.6 Pubblicazione dei dati.....	56
Capitolo 5. Rivelatore per monitor a bassa intensità.....	58
5.1 Fibre scintillanti.....	58
5.1.1 Gli scintillatori.....	58
5.1.2 Fibre scintillanti.....	59
5.2 Fotomoltiplicatori.....	60
5.2.1 SiPM.....	61
5.3 Monitor a bassa intensità.....	63
Capitolo 6. Scheda di acquisizione (CAEN – DT5550W / A55CIT4).....	68
6.1 DT5550W.....	68
6.2 A55CIT4.....	72

Capitolo 7. SCI-Compiler	77
7.1 Descrizione GUI del software.....	78
7.1.1 Toolbar “Home”.....	79
7.1.2 Toolbar “Tools Box”.....	80
7.2 Programmazione con SCI-Compiler.....	81
7.2.1 Compilazione del firmware.....	83
7.2.2 Programmazione dell’FPGA.....	85
7.2.3 Come testare il firmware: Resource Explorer.....	85
7.2.4 Aggiornamento permanente del firmware.....	86
Capitolo 8. Programmazione scheda DT5550W/ A55CIT4	89
8.1 Generazione Latch e ResetD.....	92
8.2 Clk128.....	96
8.3 Acquisizione.....	100
8.4 Citiroc.....	103
8.5 SelIndirizzi.....	105
8.6 Serializzatore.....	108
8.7 Top.....	111
Capitolo 9. DD Low Int	119
9.1 Shift Register Serial In Parallel Out (SIPO).....	119
9.2 Schema elettrico.....	122
9.3 PCB DD Low Int.....	133
9.4 Scheda finale.....	134
Capitolo 10. Test di acquisizione con il DDS	142
10.1 Programma LabVIEW.....	143
10.1.1 Block Diagram codice LabVIEW FPGA.....	143
10.1.2 Block Diagram codice LabVIEW PC host.....	148
10.2 Front Panel dei due codici LabVIEW.....	149
10.3 Test di acquisizione dati.....	150
Conclusion	III

Bibliografia.....V

Introduzione

La tesi in esame ha come oggetto la spiegazione e descrizione del lavoro volto a realizzare un monitor per la misurazione di fasci di particelle a bassa intensità. L'attività, svolta in prima persona, è stata effettuata presso la Fondazione CNAO (Centro Nazionale di Adroterapia Oncologica) e si inserisce all'interno di un progetto di collaborazione con l'Istituto Nazionale di Fisica Nucleare (INFN) della sezione di Roma. La Fondazione CNAO è l'unico centro in Italia (uno dei sei al mondo) in grado di erogare trattamenti di adroterapia mediante l'impiego di protoni e ioni carbonio. L'adroterapia utilizza fasci di particelle cariche per un maggior controllo della dose di energia rilasciata al paziente richiesta per il trattamento del tumore, rispetto alla radioterapia convenzionale, lasciando allo stesso tempo quasi del tutto indenni gli organi e i tessuti sani circostanti. Il controllo della dose rilasciata al paziente è strettamente legato alla capacità di controllare con precisione le proprietà del fascio, in termini di intensità, profilo e localizzazione, e al corretto posizionamento dello stesso paziente durante tutta la terapia. Questo delicato compito viene svolto al CNAO dal Dose Delivery System (DDS), un sistema intelligente basato su FPGA (Field Programmable Gate Array), costituito da due monitor, che serve a tenere sotto controllo il fascio di particelle e ad intervenire per la sicurezza del paziente in caso di situazioni anomale. Il team dell'INFN ha pensato di realizzare un nuovo monitor da integrare con il DDS, in grado di rilevare fasci con intensità molto bassa. I monitor presenti nel DDS, infatti, facendo parte di un sistema che serve a controllare dei fasci per adroterapia con intensità terapeutiche, non riescono a misurare fasci con intensità molto bassa (inferiore a 1 milione di particelle al secondo). Il DDS modificato sarà in grado di gestire erogazioni di fasci nella Sala Sperimentale del CNAO nei casi in cui il numero di particelle richiesto sia basso.

Per questo motivo, l'INFN di Roma si è posto come obiettivo la realizzazione di un monitor costituito da fibre scintillanti accoppiate con SiPM (Fotomoltiplicatori al Silicio), in modo da avere una sensibilità di misura molto più alta. I SiPM sono contenuti in una scheda elettronica programmabile (DT5550W / A55CIT4), prodotta dalla CAEN S.p.A., che andrà a costituire il sistema di acquisizione dati, elaborando e gestendo gli impulsi elettrici provenienti dai SiPM. Il sistema di misura dovrà essere in grado di misurare la posizione delle particelle quando attraversano il monitor e di contare il numero di particelle passanti. L'originalità e la grandezza del sistema risiedono, quindi, nella sua capacità di rilevare singole particelle.

Il lavoro, dunque, è stato svolto su due fronti. Il team dell'INFN si è occupato della costruzione del monitor, in particolare degli aspetti meccanici e funzionali; la parte ad esso dedicata, per questa ragione, sarà introdotta brevemente e se ne illustreranno le principali basi, in quanto parte integrante dell'intero progetto.

L'obiettivo del mio lavoro svolto presso il CNAO, presentato nella tesi, è lo sviluppo dell'interfaccia tra il sistema di acquisizione del monitor con il DDS. Inizialmente, si dovrà programmare la scheda DT5550W / A55CIT4, che contiene una FPGA programmabile Xilinx XC7K160T attraverso un software, SCI-Compiler, che permette l'implementazione grafica user-friendly del firmware con la quale programmare l'FPGA stessa. L'interfaccia dovrà, poi, essere conforme alla struttura del protocollo di comunicazione già esistente del DDS e rispettare alcuni vincoli provenienti dall'elettronica presente in esso. In aggiunta, sarà necessario progettare,

assemblare e testare un'ulteriore scheda elettronica, il cui fine è modificare il protocollo di trasferimento dei dati dalla scheda DT5550W / A55CIT4, passando da una trasmissione di tipo seriale ad una parallela per avere compatibilità con l'elettronica del DDS. Dopo la progettazione firmware della DT5550W / A55CIT4 e la costruzione della seconda scheda, si procederà con la programmazione di un'FPGA del DDS, attraverso il software LabVIEW, per permettere la sua comunicazione con il sistema di acquisizione e poter effettuare dei test per verificare la correttezza della catena di acquisizione dati.

Adroterapia

L'adroterapia è la tecnica utilizzata presso la Fondazione CNAO per il trattamento e la cura di tumori spesso inoperabili o resistenti ai tradizionali trattamenti radioterapici, che dal 2017 è entrata a far parte dei Livelli Essenziali di Assistenza previsti dal Sistema Sanitario Nazionale. A differenza della radioterapia tradizionale, che si basa sull'utilizzo di raggi X o elettroni, l'adroterapia prevede l'uso di protoni e ioni carbonio. Queste particelle atomiche (definite "adroni", da cui deriva il nome della terapia stessa) hanno il vantaggio di essere più pesanti e dotate di maggior energia rispetto agli elettroni e, di conseguenza, di essere ancora più efficaci nel distruggere le cellule tumorali. Si evince, dunque, come questa tecnica possa essere utilizzata per irradiare i tessuti tumorali con alte dosi e migliore selettività spaziale 3D rispetto alla radioterapia tradizionale. L'uso di fasci di protoni e di fasci di carbonio necessitano di acceleratori ad alta energia, come i ciclotroni o sincrotroni, che si adattano per generare protoni con energie dell'ordine di 250 MeV o più. La radioterapia tradizionale, invece, viene eseguita utilizzando fotoni (raggi X) e fasci di elettroni energetici che distruggono i tessuti malati ma anche quelli sani. L'innovativa terapia adronica utilizza principalmente protoni energetici e, negli ultimi anni, ioni di carbonio, per raggiungere il tumore e danneggiare irreversibilmente le cellule malate, salvando molte più cellule sane. La localizzazione spaziale 3D del tumore, ora eseguita utilizzando diverse tecniche diagnostiche, consente di definire la forma, la dimensione, la profondità, la massa e la densità del tumore. Nel caso della terapia adronica, questi dati consentono di scegliere esattamente l'energia da dare agli adroni, in modo che il picco di Bragg possa essere rilasciato nel sito tumorale.

Per capire cosa sia il picco di Bragg, che è alla base della tecnica adroterapica, occorre prima presentare alcune conoscenze di fisica, utili a comprendere meglio il fenomeno; in particolare, le interazioni tra fotoni e materia e quelle tra particelle cariche e materia.

1.1) Interazione radiazione-materia

Considerando un fascio monocromatico di particelle, ossia con particelle aventi tutte la stessa velocità e direzione, definiamo le seguenti grandezze:

- n : densità delle particelle, data dal numero delle particelle (N) diviso il volume (V) attraversato da esse;
- v : velocità delle particelle;
- $I = nv$, intensità del fascio di particelle (dimensionalmente pari a $m^{-2} s^{-1}$);

Lo scopo è calcolare il numero di particelle che attraversano una superficie S , perpendicolarmente ad essa, ad una velocità v , in un certo intervallo di tempo Δt . Nella Fig. 1.1, la superficie S è la base del cilindro. Quest'ultimo rappresenta il fascio di particelle che attraversano la superficie S nell'intervallo di tempo Δt . Le particelle al di fuori del cilindro non riescono a raggiungere la superficie nell'intervallo Δt .

$$N = nV = nSl = nSv\Delta t = IS\Delta t \quad (1.1)$$

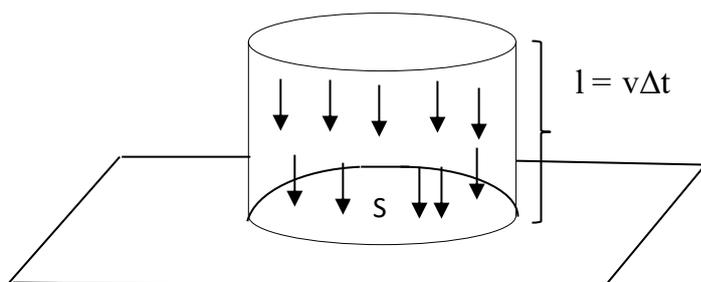


Fig. 1.1. Fascio di fotoni che attraversa una superficie S.

Nel caso in cui il fascio non fosse perpendicolare alla superficie, il numero di particelle sarebbe pari a $IS\Delta t\cos\theta$, dove θ è l'angolo compreso tra la direzione del fascio e la perpendicolare alla superficie.

Vedendo la (1.1), è stata definita l'intensità di un fascio di particelle come il numero di particelle che attraversano una superficie S nell'intervallo di tempo Δt .

Si considera, ora, un fascio di particelle che vada ad interagire con un singolo nucleo, dove, con il termine nucleo, si intende l'insieme di protoni e neutroni che costituiscono un determinato atomo. Al fine di conoscere quante particelle interagiscano con il nucleo in un certo intervallo di tempo Δt , bisogna definire un range di interazione, all'interno del quale si verifica un'interazione dal punto di vista energetico tra le particelle del fascio e il nucleo. Nella Fig.1.2, che mostra l'esempio di interazione tra una singola particella e un nucleo, si vede come il range di interazione è dato dalla circonferenza di raggio r_0 attorno al nucleo. Affinché ci sia interazione, la particella deve passare abbastanza vicina al nucleo. In Fig. 1.2, il fotone γ passa a una distanza dal nucleo pari a b , che è detto parametro d'urto. Se $b > r_0$, non ha luogo l'interazione, mentre se $b < r_0$ l'interazione energetica è presente e si può parlare di collisione della particella con il nucleo.

Il numero di interazioni nell'intervallo Δt è uguale alla densità delle particelle del fascio per il volume del cilindro costituito da particelle con $b < r_0$: $N_{\text{int}} = nV = n\pi r_0^2 v\Delta t = I \pi r_0^2 \Delta t$, dove definiamo πr_0^2 come la sezione d'urto σ . Questa grandezza dipende dalla velocità e dal tipo di particella, ma anche dalle caratteristiche del nucleo (numero atomico e numero di massa).

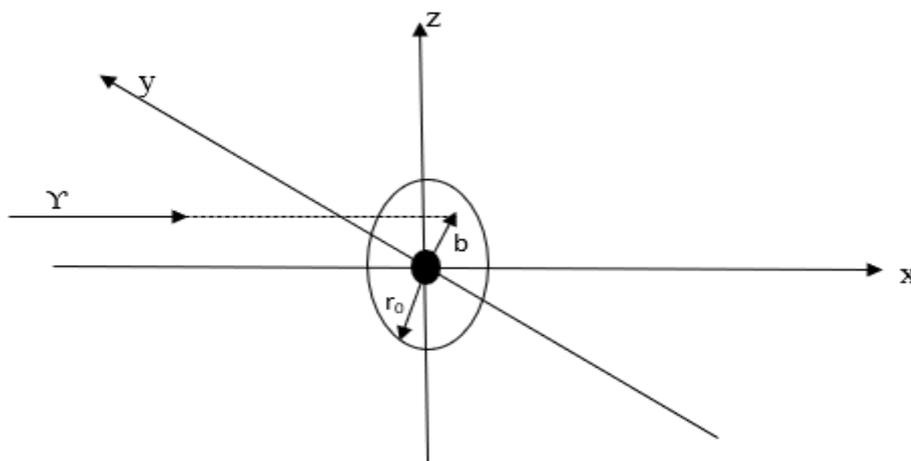


Fig.1.2. Interazione tra un fotone e un nucleo.

1.1.1) Bersaglio sottile

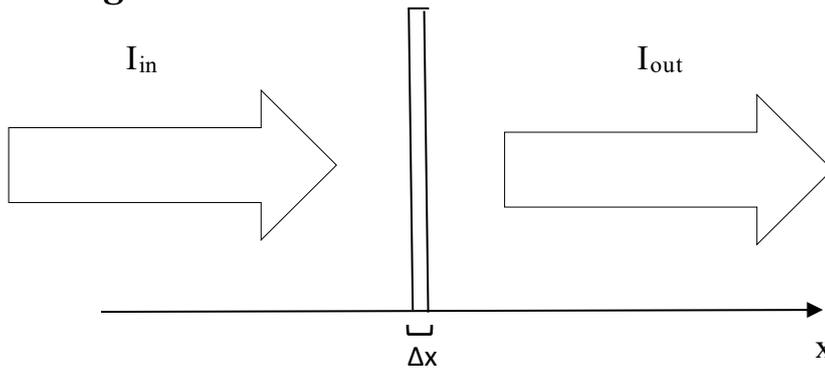


Fig. 1.3. Fascio di particelle che attraversa un bersaglio di spessore Δx .

In Fig.1.3 viene mostrato un insieme di particelle di intensità I_{in} che attraversano un bersaglio sottile, cioè con uno spessore piccolo Δx . Consideriamo sempre la superficie S del bersaglio perpendicolare alla direzione delle particelle. Si vede come, superato il bersaglio, il fascio avrà una intensità $I_{out} \neq I_{in}$, a causa di fenomeni che si verificano quando il fascio attraversa il bersaglio. Le particelle di intensità I_{out} sono quelle che hanno la stessa direzione di quelle entranti nel bersaglio.

Definiamo, nell'intervallo di tempo Δt :

- il numero di particelle in ingresso nella superficie S , $N_{in} = I_{in} S \Delta t$;
- il numero di interazioni, $N_{int} = I_{in} \sigma N_b \Delta t = I_{in} \sigma n_b S \Delta x \Delta t$. N_b è il numero di nuclei del bersaglio, quindi n_b è la densità dei nuclei del bersaglio.
- Il numero di particelle uscenti dalla superficie S , $N_{out} = I_{out} S \Delta t$;

Scrivendo l'equazione:

$$N_{out} = N_{in} - N_{int}; \quad (1.2)$$

e sostituendo con le uguaglianze scritte sopra, si ottiene:

$$I_{out} = I_{in} (1 - \sigma n_b \Delta x); \quad (1.3)$$

Un bersaglio si definisce sottile quando sono poche le particelle di un fascio che, attraversandolo, subiscono fenomeni di scattering e assorbimento. In termini matematici vuol dire che: $\Delta I = I_{in} - I_{out} \ll I_{in}$, cioè $\frac{I_{in} - I_{out}}{I_{in}} \ll 1$.

E' possibile allora scrivere, considerando la (1.3), che $\sigma n_b \Delta x \ll 1$; quindi, affinché un bersaglio possa essere considerato sottile, il suo spessore deve rispettare la condizione seguente: $\Delta x \ll \frac{1}{\sigma n_b}$.

1.1.2) Bersaglio spesso

Un bersaglio si definisce spesso se la condizione $\Delta x \ll \frac{1}{\sigma n_b}$ non è rispettata. Nel caso in Fig.1.4, $\Delta x = L$ che non è $\ll \frac{1}{\sigma n_b}$. I_{in} corrisponde all'intensità del fascio in $x = 0$, mentre I_{out} sarà l'intensità del fascio in $x = L$. Quest'ultima sarà minore di I_{in} dato che ci sarà una variazione negativa dell'intensità del fascio attraverso il bersaglio.

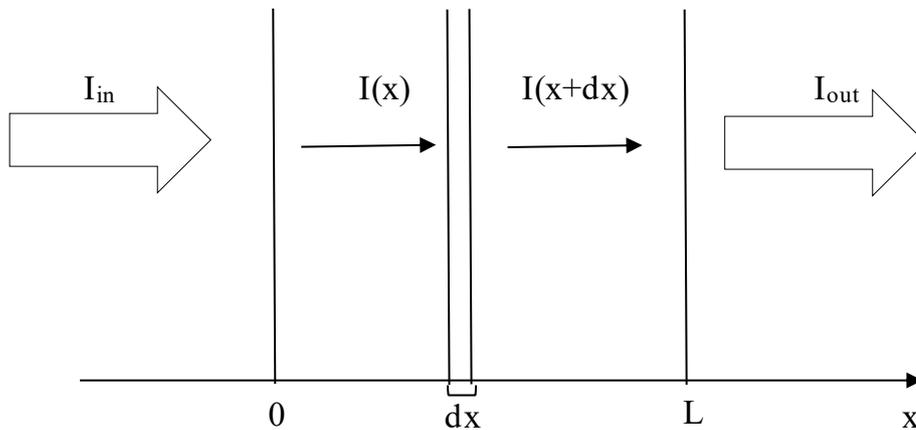


Fig. 1.4. Fascio di particelle che attraversa un bersaglio di spessore L .

Riscrivendo la (1.3) per il caso in esame, si ottiene:

$$I(x + dx) = I(x) - I(x)\sigma n_b dx ;$$

da questa si può scrivere:

$$\frac{I(x+dx) - I(x)}{dx} = -\sigma n_b I(x) ;$$

e quindi:

$$\frac{dI}{dx} = -\sigma n_b I ; \tag{1.4}$$

Dove σn_b è uguale al coefficiente di attenuazione μ . Risolvendo l'equazione differenziale (1.4), avendo posto come condizione iniziale $I(x = 0) = I_0$, si ottiene:

$$I(x) = I_0 e^{-\mu x} ; \tag{1.5}$$

quindi:

$$I_{out} = I_{in} e^{-\mu L}. \tag{1.6}$$

L'andamento dell'intensità del fascio in funzione dello spazio x sarà:

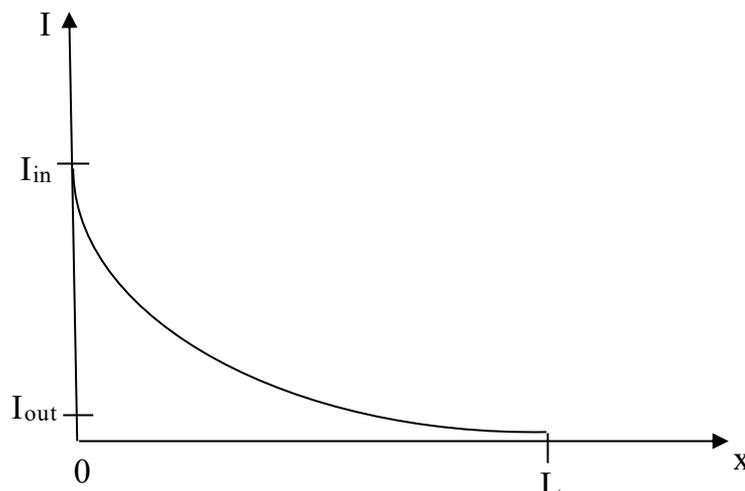


Fig. 1.5. Andamento esponenziale decrescente dell'intensità del fascio di particelle che attraversa il bersaglio.

Tuttavia, il fenomeno appena trattato è stato semplificato dal fatto che il coefficiente di attenuazione μ è stato considerato costante. Quest'ultimo, in realtà, dipende da x , in quanto la sezione d'urto e la densità dei nuclei del bersaglio (da cui μ dipende) sono dipendenti da x . Perciò, la (1.6) si trasforma in:

$$I_{\text{out}} = I_{\text{in}} e^{-\int_0^L \mu(x) dx} . \quad (1.7)$$

Detto ciò, si può parlare di due tipi di interazioni, cioè quelle tra fotoni e materia e quelle tra particelle cariche e materia, soffermandosi, in particolare, sulla seconda, che costituisce l'oggetto di interesse per l'adroterapia.

1.2) Interazioni fotoni-materia

I fotoni sono quanti di energia associati al campo elettromagnetico. La radiazione elettromagnetica, infatti, è costituita da particelle elementari, i fotoni, ognuno con energia pari ad $h\nu$ (h = costante di Planck; ν = frequenza di vibrazione del fotone), con massa pari a zero e carica nulla. Essi interagiscono con nuclei o con elettroni. Gli elettroni, in base alla cinetica di interazione con il fotone, possono essere considerati nella collisione come particelle singole libere, oppure come parte di un atomo, quindi energeticamente legati ad esso.

In generale, un elettrone è sempre legato ad un atomo, dato che gli elettroni liberi interagiscono con altre particelle, oppure vengono immediatamente catturati da atomi, esistendo da liberi nella materia per pochissimo tempo. Se un fotone colpisce un elettrone, che fa parte di un atomo, con un'energia dell'ordine di grandezza dell'energia di ionizzazione dell'atomo, questo può essere considerato legato all'atomo. Se, invece, il fotone ha un'energia molto maggiore rispetto all'energia di ionizzazione dell'atomo, allora l'elettrone può essere considerato come una particella libera nello spazio, dato che la sua energia di legame con l'atomo sarà trascurabile rispetto all'energia con la quale il fotone lo colpisce. Inoltre, detta m_e la massa dell'elettrone, se il fotone che colpisce l'elettrone ha un'energia molto minore rispetto a $m_e c^2$, (dove c è la velocità della luce) allora l'elettrone non raggiungerà velocità relativistiche e potrà essere trattato come una particella che risponde alla fisica classica. Se il fotone ha un'energia dell'ordine, o maggiore, di $m_e c^2$, in seguito alla collisione, l'elettrone avrà un'energia molto simile a quella del fotone che lo ha colpito e dovrà, dunque, essere trattato con la fisica relativistica. Nella Fig.1.6 viene riassunto quanto detto graficamente.

Le interazioni tra fotoni ed elettroni si possono associare a due tipi di fenomeni: scattering e assorbimento.

1.2.1) Scattering

Nello scattering si hanno un fotone e un elettrone sia prima che dopo la collisione. Ciò che cambia dopo la collisione è che la traiettoria del fotone viene deviata. Anche l'elettrone, solitamente considerato fermo avendo energia molto minore rispetto a quella del fotone incidente, avrà una nuova traiettoria; inoltre, il momento del fotone diminuirà a causa della collisione, nella quale parte della sua energia viene trasferita all'elettrone. Con il termine 'momento' di una particella si fa riferimento alla sua quantità di moto. La variazione del momento del fotone e, quindi, dell'energia,

Elettrone legato all'atomo;	Elettrone libero;	Elettrone libero;
particella classica	particella classica	particella relativistica

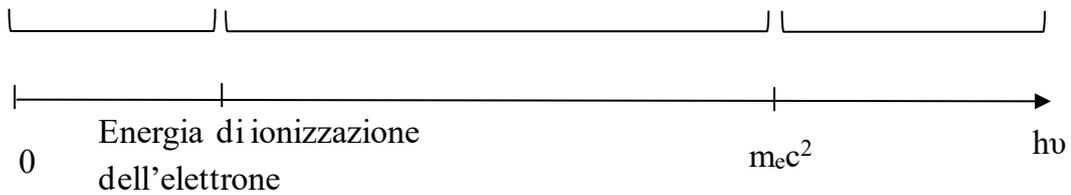


Fig. 1.6. Classificazione dell'elettrone in base all'energia di collisione del fotone.

implica anche una variazione della lunghezza d'onda di quest'ultimo. In base alle energie che vi sono in gioco, è possibile definire diversi tipi di scattering:

- **Scattering di Compton:** interazione tra fotone ed elettrone in cui l'energia $h\nu$ del fotone è dell'ordine, o molto maggiore, di $m_e c^2$. Questo è il caso in cui l'elettrone, dopo la collisione, acquisisce velocità relativistiche.
- **Scattering di Thomson:** interazione tra fotone ed elettrone in cui l'energia $h\nu$ del fotone è molto minore di $m_e c^2$, ma maggiore dell'energia di ionizzazione dell'atomo a cui l'elettrone è legato. In questo caso, dopo la collisione, l'elettrone non raggiunge velocità relativistiche, e viene considerato come una particella classica.
- **Scattering di Rayleigh:** interazione tra fotone ed elettrone in cui l'energia $h\nu$ del fotone è minore o uguale all'energia di ionizzazione dell'atomo. In questo caso l'energia del fotone non cambia dopo l'interazione, quindi il fotone, dopo aver interagito con l'elettrone, avrà la stessa lunghezza d'onda che aveva inizialmente.

In generale, il fenomeno dello scattering ha la stessa natura nei tre casi; cambiano, unicamente, le energie coinvolte e la sezione d'urto dell'interazione. Quest'ultima, in particolare, per frequenze di vibrazione basse del fotone, ovvero quando $h\nu \ll m_e c^2$, non varia e rimane pressoché costante. In corrispondenza di energie dell'ordine, o maggiori, di $m_e c^2$, la sezione d'urto diminuisce. Questo comportamento è mostrato in Fig.1.7.

1.2.2) Assorbimento

L'assorbimento si differenzia dallo scattering per il fatto che il fotone scompare dopo l'interazione. Questo fenomeno può avvenire solo con un elettrone che è legato ad un atomo, e non con un elettrone libero. Un esempio è la produzione di una coppia elettrone-positrone, che consiste nell'interazione tra un fotone e un atomo o un elettrone, a seguito della quale il fotone scompare. L'atomo o l'elettrone avranno un momento diverso rispetto a quello prima dell'interazione ed un elettrone e un positrone saranno generati con un determinato momento. Affinché questo fenomeno si realizzi, l'energia del fotone incidente deve superare un certo valore di soglia; al di sotto di questo la sezione d'urto è nulla e l'evento non si verifica.

Un altro fenomeno di assorbimento è l'effetto fotoelettrico. In questo caso, un elettrone viene scalzato dall'orbitale dell'atomo in cui si trova, a seguito dell'interazione con il fotone incidente.

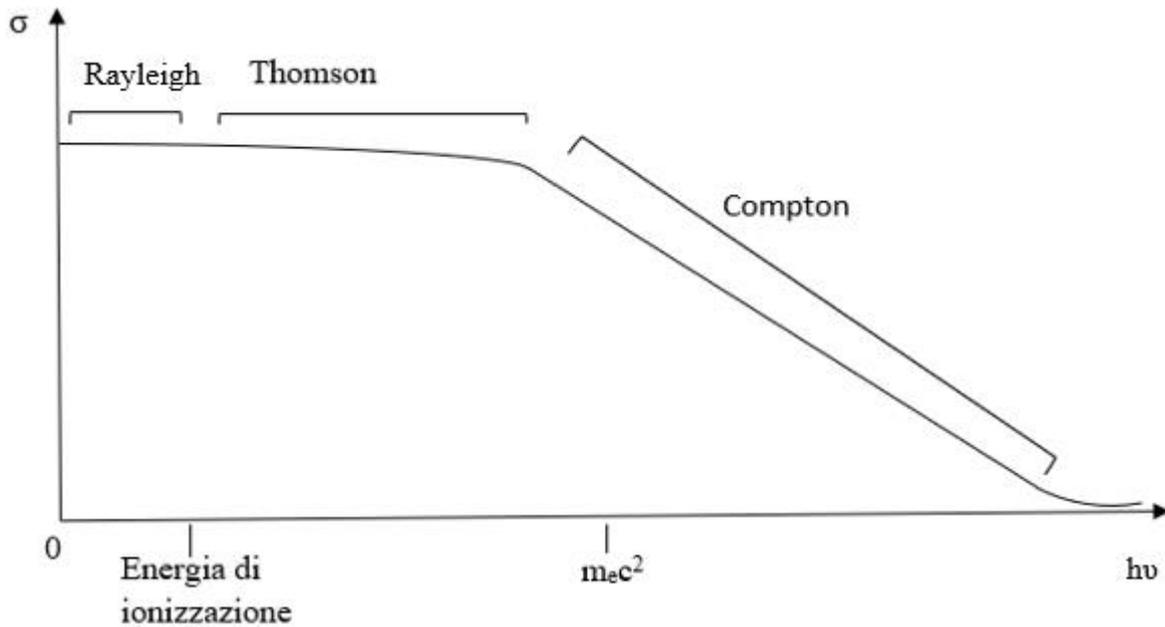


Fig. 1.7. Sezione d'urto in funzione dei vari tipi di scattering descritti.

L'energia dell'elettrone emesso dipende esclusivamente dalla frequenza del fotone e, quindi, dalla sua energia. L'elettrone lascia il suo orbitale se l'energia incidente è maggiore rispetto a quella che tiene legato l'elettrone all'atomo; elettroni che si trovano in orbitali più vicini al nucleo dell'atomo necessitano di un'energia maggiore per essere allontanati. Trattandosi di un fenomeno di assorbimento, il fotone scompare dopo aver spinto fuori l'elettrone dal suo orbitale; tuttavia, dall'effetto fotoelettrico, possono nascere ulteriori fenomeni dovuti al fatto che, essendo stato espulso un elettrone da un orbitale interno dell'atomo, quest'ultimo diviene instabile e cerca di ritornare ad una situazione di stabilità. Per raggiungere nuovamente la stabilità, un elettrone, proveniente da un orbitale ad energia maggiore di quello in cui è presente la vacanza elettronica, va a ricoprire quest'ultima. Si possono generare, quindi, due effetti:

- La fluorescenza: viene emesso un fotone a causa del passaggio di un elettrone da un orbitale a energia maggiore ad uno ad energia inferiore; di conseguenza, il fotone avrà energia pari alla differenza di quella tra i due orbitali.
- L'effetto Auger: dovuto sempre al fatto che un elettrone passa da un orbitale a energia maggiore ad uno a energia minore. In questo caso, la differenza energetica tra gli orbitali non viene emessa sotto forma di un fotone, ma viene trasferita ad un altro elettrone del guscio più esterno che viene estratto dall'atomo.

Esistono tanti altri tipi di interazioni tra fotoni ed elettroni sui quali non occorre soffermarsi, in quanto non necessari all'oggetto del capitolo. In generale, bisogna tener conto che un fascio di fotoni che attraversa la materia va incontro ad una serie di fenomeni di scattering ed assorbimento, che hanno una certa sezione d'urto. Le sezioni d'urto si sommano per determinare la sezione d'urto totale dell'interazione σ_{TOT} . L'intensità del fascio di fotoni, quindi, varia secondo la legge già descritta nell'equazione (1.5): $I_\gamma = I_{\gamma 0} e^{-\mu x}$, con $\mu = \sigma_{TOT} n$ (n = densità dei nuclei del volume attraversato dal fascio); μ , oltre a dipendere dalla variabile spaziale, è anche legata all'energia dei fotoni incidenti, a causa della dipendenza della sezione d'urto da essa, come mostrato anche in Fig. 1.7.

1.3) Interazioni particelle cariche-materia

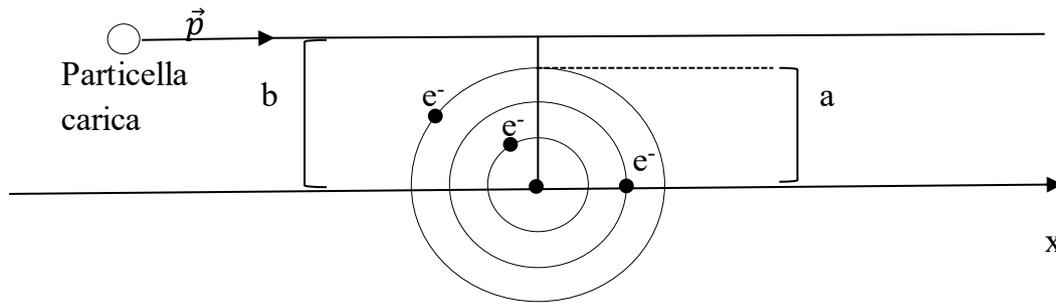


Fig. 1.8. Rappresentazione generica di un'interazione tra una particella carica e un atomo.

Quando particelle cariche, come elettroni o ioni, viaggiano nella materia, si verificano delle interazioni, dette interazioni di Coulomb, che possono coinvolgere direttamente i nuclei, generando fenomeni di scattering tra la particella carica e il nucleo o con gli elettroni degli atomi, causando o il loro passaggio ad uno stato ad energia maggiore o la ionizzazione dell'atomo, estraendo l'elettrone. Le interazioni tra particelle cariche sono governate dalla forza elettrostatica, o forza di Coulomb. Ricordiamo che questa è data dalla formula $F = \frac{q_1 q_2}{4\pi\epsilon_0 r^2}$, dove q_1 e q_2 sono le cariche delle particelle che interagiscono, ϵ_0 è la costante dielettrica del vuoto e r è la distanza tra le cariche.

In Fig. 1.8, vediamo una particella carica generica con un determinato momento \vec{p} che passa vicina ad un atomo che ha una serie di orbitali nei quali sono presenti degli elettroni. Nella figura si vedono due parametri, che sono il raggio dell'atomo a e il parametro d'impatto b , già introdotto nel Par. 1.1. Questi sono utili per definire diversi tipi di collisioni in base ai loro valori:

- Se $b \ll a$ si ha un processo radiativo: processo in cui un atomo o un nucleo compie una transizione da uno stato eccitato o ad uno a energia inferiore o allo stato fondamentale, emettendo un fotone. Per esempio, un elettrone che passa molto vicino a un nucleo perde una certa quantità di energia, che viene utilizzata per emettere un fotone; è il processo alla base della genesi della radiazione di Bremsstrahlung per l'emissione dei raggi X.
- Se $b \gg a$ si ha una collisione leggera: la particella carica interagisce con l'atomo, considerato nella sua interezza, e non con i singoli elettroni. L'energia di interazione è bassa dato che la forza di Coulomb dipende dal quadrato della distanza tra la particella e l'atomo, perciò, più b è grande, più l'energia della collisione è piccola.
- Se $b \leq a$ si ha una collisione forte: la particella carica interagisce con i singoli elettroni e, per il discorso analogo fatto nel caso precedente, l'energia coinvolta nell'interazione sarà elevata. In questa interazione, bisogna anche considerare il tipo di particelle cariche in questione. Se si considera un'interazione forte tra un elettrone che viaggia nello spazio e un elettrone dell'atomo, ci sarà una deviazione della traiettoria del primo elettrone, oltre a una perdita energetica, perché le due masse sono uguali tra di loro. Se l'interazione fosse tra un ione che viaggia nello spazio e un elettrone atomico, allora l'effetto della collisione sullo ione, in termini di deviazione della traiettoria, non si vedrebbe, a causa della differenza di massa. La massa di uno ione, infatti, è circa tre ordini di grandezza superiore a quella dell'elettrone; quindi, la forza di Coulomb nelle due interazioni descritte è sempre la stessa, cambiano solo le masse in gioco, che generano degli effetti diversi.

1.3.1) Scattering Coulombiano

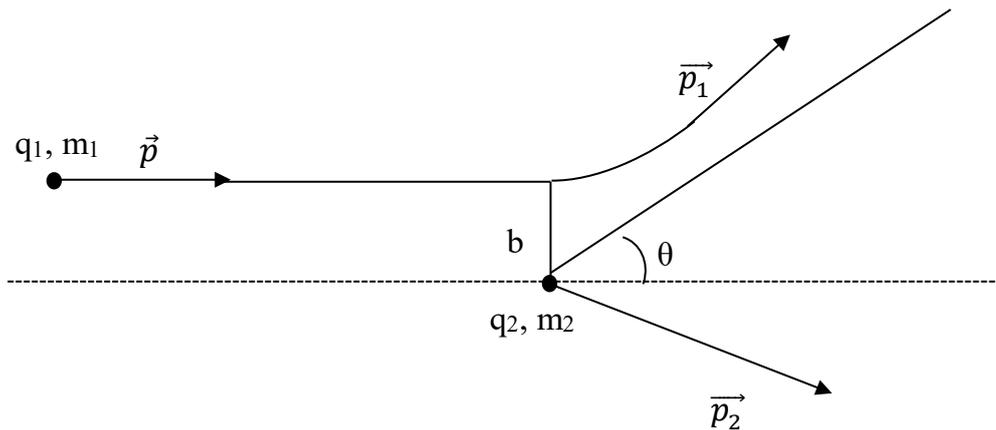


Fig.1. 9. Scattering di Coulomb tra due particelle cariche.

Fenomeno analogo a quello già descritto per i fotoni, che, però, coinvolge due particelle cariche. In questa interazione il momento totale e l'energia si conservano. Considerando per semplicità solo il momento, l'equazione che descrive la conservazione del momento, relativo alla Fig.1.9, è:

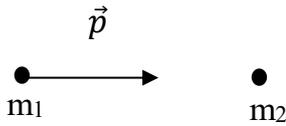
$$\vec{p} = \vec{p}_1 + \vec{p}_2 ; \quad (1.8)$$

dove \vec{p} è il momento della particella con carica q_1 e massa m_1 prima della collisione. \vec{p}_1 e \vec{p}_2 sono, rispettivamente, il momento della particella con carica q_1 e massa m_1 dopo la collisione e il momento della particella con carica q_2 e massa m_2 dopo la collisione che, prima, era considerata ferma. Come il momento si distribuisce tra le due particelle dopo l'urto è determinato dalla dinamica dell'interazione, la quale dipende molto dal parametro d'impatto b .

Infatti, la massima energia elettrostatica è uguale a $E = \frac{q_1 q_2}{4\pi \epsilon_0 b}$, che si ottiene quando la distanza tra le due particelle è pari al parametro d'impatto. In base a b cambiano le dinamiche di scattering:

- Se b è molto grande, allora l'energia elettrostatica massima scritta sopra sarà molto minore rispetto all'energia cinetica della particella q_1 prima dell'impatto (ϵ_1). In questo caso, l'angolo θ che si vede anche in Fig. 1.9, che sarebbe l'angolo tra la direzione della traiettoria di q_1 prima e dopo la collisione, è molto piccolo. Il momento \vec{p}_1 sarà, all'incirca, uguale a \vec{p} e \vec{p}_2 sarà circa uguale a zero.
- Se b è piccolo, l'energia elettrostatica massima è dell'ordine dell'energia cinetica ϵ_1 . L'angolo θ , ora, è più grande e parte del momento che aveva la particella q_1 viene ceduto alla particella q_2 .
- Un caso un po' più particolare è quello in cui $b = 0$. In questo caso, si parla di collisione frontale e l'angolo θ sarà pari a π . In questa interazione, la particella q_1 arriva ad un punto in cui la sua forza elettrostatica con q_2 è uguale alla sua energia cinetica e , e quindi, ritorna indietro. La collisione frontale fa sì che i momenti prima e dopo l'urto siano sempre nella stessa direzione, come mostrato in Fig.1.10. Si può nuovamente riprendere il discorso relativo alle masse delle due particelle che collidono: se la particella di massa m_1 è uno ione

Prima della collisione



Dopo la collisione

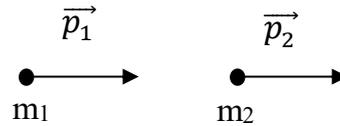


Fig. 1.10. Direzione dei momenti delle due particelle prima e dopo la collisione frontale.

che impatta la particella di massa m_2 , che è un elettrone, con una collisione frontale, la prima non tornerà indietro, avendo una massa molto più grande della seconda. Essa spingerà in avanti m_2 , fornendogli un momento che ha la sua stessa direzione. Al contrario, se m_1 è un elettrone e m_2 è uno ione, la prima tornerà indietro, in seguito alla collisione frontale ed avrà un momento sempre uguale alla direzione che aveva prima dell'impatto, ma con verso opposto. Inoltre, se le due particelle che si scontrano hanno la stessa identica massa, allora la particella che collide perderà completamente tutta la sua energia cinetica, dandola alla particella che subisce l'urto.

In generale, la collisione frontale è la situazione in cui c'è il massimo scambio di energia tra le due particelle.

Una considerazione importante da fare riguardo le interazioni Coulombiane è che la particella carica perde una certa quantità di energia, la quale è funzione dell'energia della particella stessa e dal parametro d'impatto.

1.3.2) Stopping Power e profondità di penetrazione

Se si andasse a considerare un target di spessore Δx e una particella carica che si muove verso di esso, si potrebbero avere due situazioni: nella prima, la particella non interagisce con il target e, quindi, la sua energia rimane la stessa; nella seconda, invece, la particella può interagire con il target e, quindi, uscire da esso con un'energia pari all'energia che aveva inizialmente meno la perdita data dall'interazione. Si considera una probabilità d'interazione pari a $p = \mu(\epsilon)\Delta x = n\sigma(\epsilon)\Delta x$. La probabilità di non interazione sarà $1-p$. Definendo $\Delta\epsilon$ come la perdita di energia causata dall'interazione, si può scrivere la seguente equazione:

$$\epsilon(x + \Delta x) = (1 - p)\epsilon(x) + p(\epsilon(x) - \Delta\epsilon) = \epsilon(x) - p\Delta\epsilon. \quad (1.9)$$

Sostituendo nella (1.9) il valore di p scritto prima, si ottiene:

$$\epsilon(x + \Delta x) - \epsilon(x) = -\mu(\epsilon)\Delta x\Delta\epsilon;$$

da cui si arriva all'equazione finale:

$$\frac{d\epsilon}{dx} = -\mu(\epsilon)\Delta\epsilon. \quad (1.10)$$

Si può, quindi, definire $\mu(\epsilon)\Delta\epsilon = S(\epsilon)$ come "Stopping Power". Da questa equazione si deduce che la variazione di energia della particella carica, in funzione dello spazio attraversato nel target, diminuisce di un fattore pari allo Stopping Power, dipendente dall'energia della particella stessa.

Nel caso in cui si conoscesse la funzione che descrive lo Stopping Power, sarebbe possibile risolvere l'equazione differenziale: $\frac{d\varepsilon}{dx} = -S(\varepsilon)$, ponendo come condizione iniziale $\varepsilon(x = 0) = \varepsilon_0$. In questo modo si troverebbe la profondità di penetrazione R della particella nel materiale, ovvero quanto la particella penetri nel materiale prima di fermarsi a causa della perdita totale della sua energia:

$$R = \int_0^{\varepsilon_0} \frac{d\varepsilon}{S(\varepsilon)}. \quad (1.11)$$

Bohr ha studiato una possibile formulazione per lo Stopping Power, andando a considerare l'interazione tra uno ione ed un elettrone atomico, facendo alcune approssimazioni:

- l'elettrone è a riposo prima dell'interazione, cioè ha energia cinetica nulla;
- l'elettrone acquisisce velocità ma non cambia la sua posizione, in quanto il tempo di interazione è talmente breve che lo spostamento è trascurabile;
- la traiettoria dello ione è una linea dritta senza deflessioni;
- la velocità dello ione è costante.

Come si vede dalla Fig. 1.11, la formula di Bohr non coincide esattamente con i dati sperimentali; tuttavia, l'andamento è simile. Alcuni anni dopo, Bethe si è avvicinato ancora di più all'andamento reale. Da questi grafici, si può affermare che lo Stopping Power per gli ioni che si muovono nella materia è basso quando questi hanno alta energia. Se le particelle pesanti, prendendo in esempio i protoni, hanno un'energia iniziale elevata, avranno, inizialmente, uno Stopping Power basso e, quindi, si muoveranno facilmente nella materia. Più questi si muovono in essa, più, pian piano, perdono energia. La perdita di energia è accompagnata da un aumento dello Stopping Power, fino ad arrivare in un range di energia in cui nella curva è presente un picco, ossia il picco di Bragg; lì i protoni perdono tutta la loro energia cinetica, fermando la loro corsa. Come si può vedere dalla Fig. 1.12, lo Stopping Power rappresentato in funzione dello spazio presenta sempre un picco; questo, però, si verifica a valori di x tanto più elevati, quanto maggiore è l'energia iniziale dei protoni.

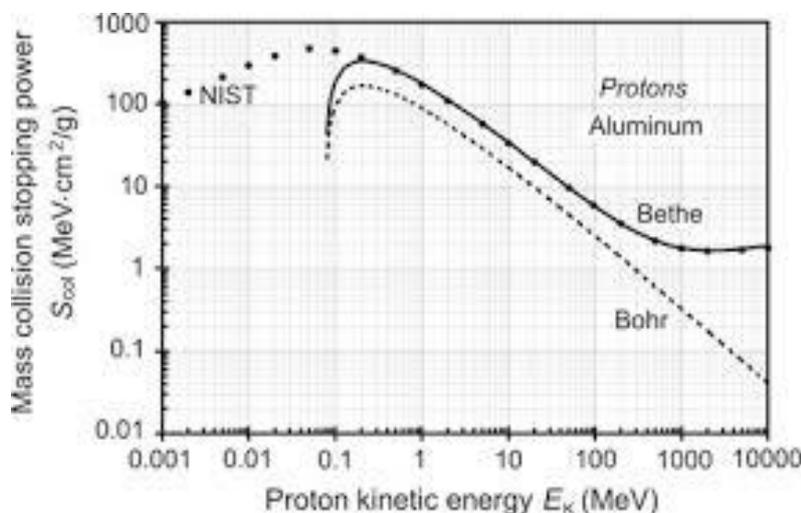


Fig. 1.11. Confronto degli Stopping Power dati dalle formule di Bohr e Bethe e dai dati sperimentali.

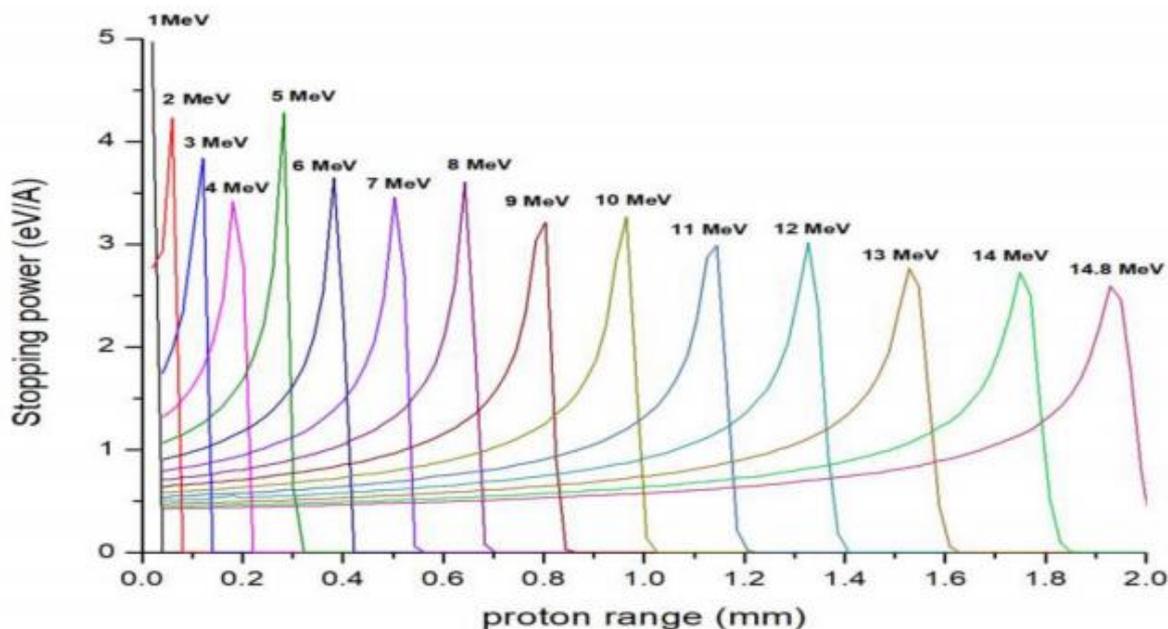


Fig. 1.12. Serie di picchi di Bragg relativi a protoni ad energia crescente.

L'adroterapia sfrutta questa proprietà che hanno le particelle pesanti, come ioni o protoni, per rilasciare un'alta quantità di energia in corrispondenza di siti tumorali, in modo da portare alla morte delle cellule cancerogene. Nell'adroterapia le particelle cariche vengono accelerate fino ad arrivare ad energie di 250 MeV ed oltre. Divengono così delle radiazioni ionizzanti, le quali hanno degli effetti dannosi sulle cellule.

Le radiazioni ionizzanti possono essere di due tipi: dirette, se consegnano la loro energia direttamente alla materia attraverso tante interazioni di Coulomb lungo il loro cammino; indirette, se causate da particelle non cariche, come atomi neutri, neutroni o fotoni, che, prima, trasferiscono la loro energia alle particelle cariche della materia incontrate nel loro cammino, le quali, a loro volta, rilasciando l'energia, come descritto prima, ionizzano la materia.

L'effetto della radiazione sulla cellula dipende, in particolare, dalla dose del fascio, dal tipo di radiazione e dal tipo di cellula. La dose è un parametro importante in questo campo; essa è definita come l'energia assorbita dalla materia, considerata in un determinato volume V , divisa la massa del volume. Per energia assorbita si intende la somma dell'energia di fotoni e particelle cariche che entrano nel volume V , a cui si sottrae la somma dell'energia di fotoni e particelle cariche che escono dal volume V , sottraendo anche eventuali variazioni positive di massa dovute a fenomeni di assorbimento, come la produzione di una coppia elettrone-positrone. In poche parole, è l'energia totale rilasciata nel volume V . La dose si misura in Gray (Gy), dove $1 \text{ Gy} = 1 \text{ J/Kg}$.

Nella Fig. 1.13, è rappresentata la dose depositata in funzione della profondità del tessuto per diverse tipologie di radiazioni. Il rilascio di energia dei raggi X a 4 MeV e del fascio di elettroni mostrano un decadimento esponenziale della dose dalla superficie più esterna della porzione di tessuto verso i tessuti più interni. In particolare, si può vedere come gli elettroni esauriscano dopo pochi centimetri tutta la loro energia. Questo indica che il tentativo di rilascio di alte dosi negli strati

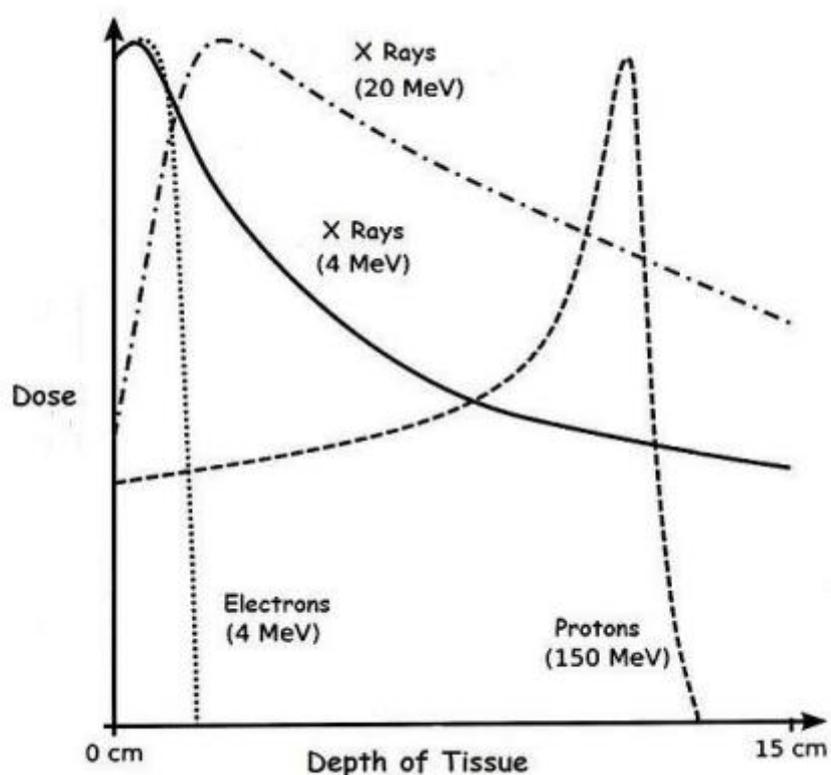


Fig. 1.13. Confronto della dose rilasciata da diversi tipi di particelle

profondi del tumore produca assorbimenti delle stesse nei tessuti sani più in superficie; quindi, terapie con elettroni e raggi X a quelle energie non sono consigliati per tumori che si trovano ad una certa profondità. Utilizzando un fascio di raggi X a 20 MeV, il risultato è migliore rispetto ai due fasci precedenti: in particolare, si vede che viene rilasciata meno dose sulla superficie del tessuto. Tuttavia, anch'esso, dopo alcuni centimetri, rilascia il suo massimo quantitativo di dose, che da lì in poi decresce sempre di più; di conseguenza, il rilascio di dose maggiore non arriva molto in profondità. Se viene mandato un fascio di protoni a 150 MeV, si nota come il picco di Bragg sia situato più in profondità rispetto ai fasci precedenti. Anche i protoni rilasciano una certa dose nei tessuti più superficiali, ma molto minore rispetto agli esempi visti precedentemente. In base alla profondità in cui è situato il tumore, l'energia con la quale i protoni vengono fatti subentrare nella materia viene cambiata, agendo a livello del sincrotrone, per avere il picco di Bragg precisamente dove desiderato. Inoltre, l'uso dei raggi X produce il rilascio di una certa quantità di dose anche negli strati più profondi dietro il sito del tumore. I protoni e i fasci di ioni pesanti, invece, mostrano una dose massima sul picco di Bragg; dopodiché l'energia rilasciata si esaurisce. Essi, quindi, massimizzano la dose nel sito tumorale, riducono significativamente la dose nei tessuti sani superficiali e azzerano la dose negli strati più profondi dietro il sito del tumore.

Cenni di progettazione di circuiti elettronici

Una parte del progetto sul monitor per basse intensità è dedicata alla realizzazione della scheda elettronica DD Low Int (Dose Delivery Low Intensity) che costituirà l'interfaccia tra il sistema Dose Delivery del CNAO e la scheda CAEN - DT5550W / A55CIT4. La trattazione che segue andrà, infatti, ad esaminare alcuni argomenti e problematiche riguardanti la progettazione elettronica, che sono state affrontate nella progettazione reale della scheda e la cui analisi sarà utile in seguito.

I circuiti elettronici sono dei circuiti elettrici costituiti da componenti elettronici, dedicati all'elaborazione di segnali elettrici che portano delle informazioni. L'informazione è rappresentata sottoforma di parole binarie che avranno una certa lunghezza, ad esempio 1 byte. La parola binaria può essere trasferita in due modi:

- Forma seriale
- Forma parallela

Essendo un'informazione binaria, l'uscita di un blocco digitale può assumere tensioni che, se sono comprese in un certo intervallo, vengono loro associate lo 0 logico, altrimenti l'1 logico. Una parola in forma seriale è un'informazione sequenziale con bit che sono presenti in tempi successivi su un'unica linea. Per interpretare correttamente la parola, è necessario sapere quando andare a leggerla, utilizzando un riferimento temporale. E' presente, quindi, un segnale che dice quando leggere la linea digitale, cioè il segnale di clock, che serve a temporizzare tutte le azioni di un sistema digitale.

Con il clock si va a leggere il segnale dati in corrispondenza dei fronti di salita o di discesa del clock. Nell'esempio in Fig. 2.1 viene mostrato un segnale dati che viene letto in corrispondenza del fronte di discesa del segnale del clock ottenendo la parola digitale 101100101000. In più, si possono affiancare al clock altri segnali che codificano l'attivazione o l'inattivazione di una linea, formando un protocollo di comunicazione proprio. Ad esempio, è possibile utilizzare un segnale che, quando la linea dati seriale è inattiva, ha valore 1 e, prima di inviare la parola, passa a livello logico 0 per un certo intervallo di tempo, ad esempio un ciclo di clock. Questo consente di sapere quando inizia l'invio della parola, ovvero quando la linea è attivata. Quindi, vedendo la transizione da 1 a 0, si capisce che c'è stata l'attivazione della linea seriale e dal ciclo di clock successivo si comincia a campionare il segnale dati.

Nella forma parallela, invece, si hanno tante linee quanti sono i bit che si intendono trasferire, come

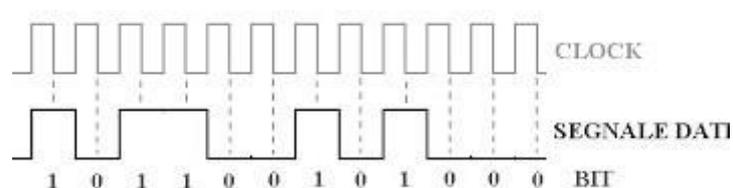


Figura 2.1. Trasmissione seriale di un segnale dati.

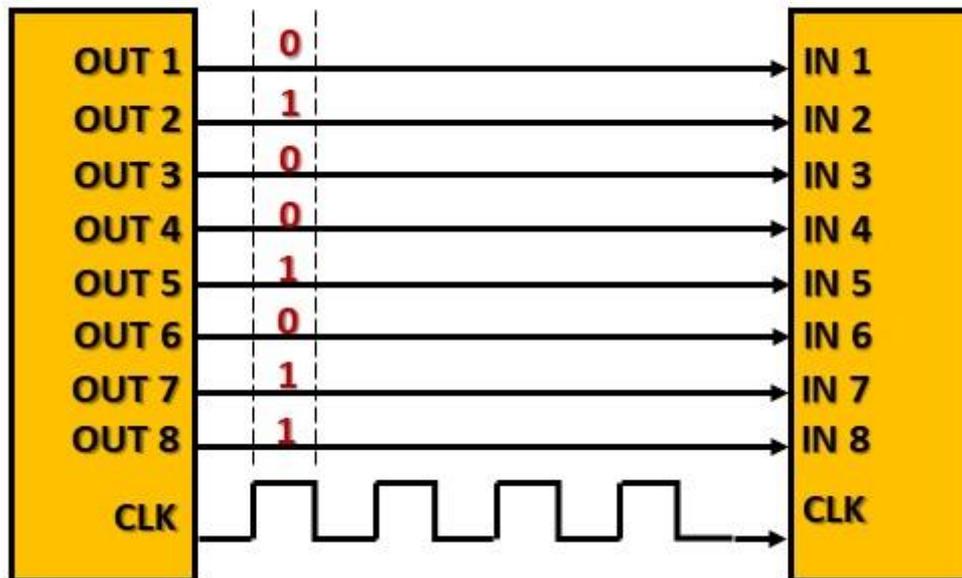


Fig.2.2. Protocollo di comunicazione parallelo.

mostrato in Fig. 2.2. Occorre sempre un clock che dica quando andare a leggere le linee. Il grosso vantaggio della forma di trasferimento dati parallela è quello di consentire la lettura contemporanea di più linee, permettendo, per esempio, di trasferire parole di 8 bit in un solo colpo di clock. Nella linea seriale viene trasferito un bit della parola digitale per colpo di clock; dunque, la velocità di trasmissione dipende dalla frequenza del clock. Tuttavia, il protocollo seriale ha il vantaggio, rispetto a quello parallelo, di risparmiare spazio in termini di hardware. In circuiti stampati si andrebbero a ridurre gli integrati da utilizzare, così come nelle comunicazioni a lunga distanza sarebbero necessari molti meno cavi; quindi, utilizzando una forma di comunicazione seriale, c'è un risparmio economico. In generale, il protocollo parallelo è più veloce ma meno versatile.

In elettronica digitale gli stati logici sono rappresentati mediante grandezze elettriche, generalmente tensioni elettriche. Quasi sempre si usa la logica positiva, dove il livello logico alto corrisponde a 1 e il livello logico basso corrisponde a 0. L'elettronica digitale si basa sul collegamento di più moduli digitali, tenendo conto delle relative alimentazioni. Con moduli digitali si intendono componenti digitali che, messi insieme, svolgono una certa funzione; hanno ingressi e uscite e devono essere alimentati (V_{AL} e GND); in realtà, possono non avere ingressi, come per esempio un oggetto che sfrutta un oscillatore interno, ma devono sempre avere delle uscite.

In sede di progettazione di un circuito elettronico diventa molto importante il collegamento delle uscite di un modulo digitale ad ingressi di altri moduli digitali. Infatti, se si hanno due moduli digitali in comunicazione tra loro, le linee di uscita del modulo pilotante devono essere ben interpretate dal modulo pilotato. Il modulo digitale pilotante e pilotato devono avere livelli logici/elettrici comparabili; ovvero una tensione che per un modulo è interpretata come alta deve essere interpretata come tale anche dall'altro.

2.1) Interoperabilità tra porte logiche

Idealmente, affinché due moduli digitali comunichino correttamente, basterebbe fissare una soglia della tensione di ingresso del modulo pilotato pari a $\frac{V_{AL}}{2}$, e, se la tensione di uscita del modulo pilotante fosse maggiore, allora si parlerebbe di livello logico alto, altrimenti basso. Tuttavia, la tensione di soglia è definibile solamente in senso statistico ma non si può sapere il suo valore preciso; inoltre, non è possibile garantire neanche un valore di soglia preciso, in relazione alla soglia del modulo pilotato, della tensione di uscita del modulo pilotante, in quanto dipende da come la porta logica è caricata, ossia da quanti siano gli input a cui l'uscita della porta si collega. Il costruttore del componente, allora, garantisce che, quando il livello logico è alto, in una condizione di carico massimo che non deve essere superata, la tensione di uscita sia sempre maggiore di V_{OH} . V_{OH} è il minimo valore della tensione di uscita della porta logica garantito dal costruttore quando si ha livello logico alto. Allo stesso modo si definisce V_{OL} come il massimo valore di tensione in uscita garantito quando si ha livello logico basso.

Analogamente, vengono definiti i valori limite V_{IH} e V_{IL} , dove il primo indica la minima tensione di ingresso che viene associata ad uno stato logico alto per l'ingresso ed il secondo la massima tensione di ingresso che viene associata ad uno stato logico basso per l'ingresso. Quindi, se $V_{input} > V_{IH}$, si ha livello logico alto, se $V_{input} < V_{IL}$, si ha livello logico basso. La tensione di soglia di ingresso si trova all'incirca nell'intorno della metà tra V_{IH} e V_{IL} ma, essendo incerta la sua posizione precisa, il costruttore definisce questi intervalli di riconoscimento dello stato logico: tra V_{AL} e V_{IH} c'è livello logico alto, tra V_{IL} e GND c'è livello logico basso. La compatibilità tra porte logiche è garantita se: $V_{OH} > V_{IH}$ e $V_{OL} < V_{IL}$.

In Fig. 2.3 è mostrato graficamente quanto detto precedentemente con dei valori di tensione di esempio. Bisogna, inoltre, dire che V_{OH} e V_{OL} sono dipendenti dal carico massimo, dato che esiste un limite al numero di porte a cui l'uscita della porta può collegarsi. Se questo venisse superato, la porta logica di uscita non riuscirebbe più a fornire corrente alle porte logiche d'ingresso collegate, causando l'abbassamento della tensione al di sotto dei livelli definiti. V_{IH} e V_{IL} , invece, sono indipendenti dal carico della porta.

I rumori e i disturbi possono modificare le tensioni presentate in uscita. La differenza tra V_{OH} e V_{IH}

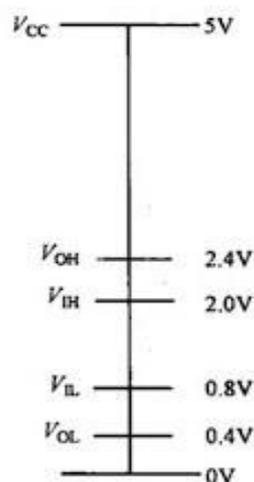


Fig. 2.3. Interoperabilità tra porte logiche.

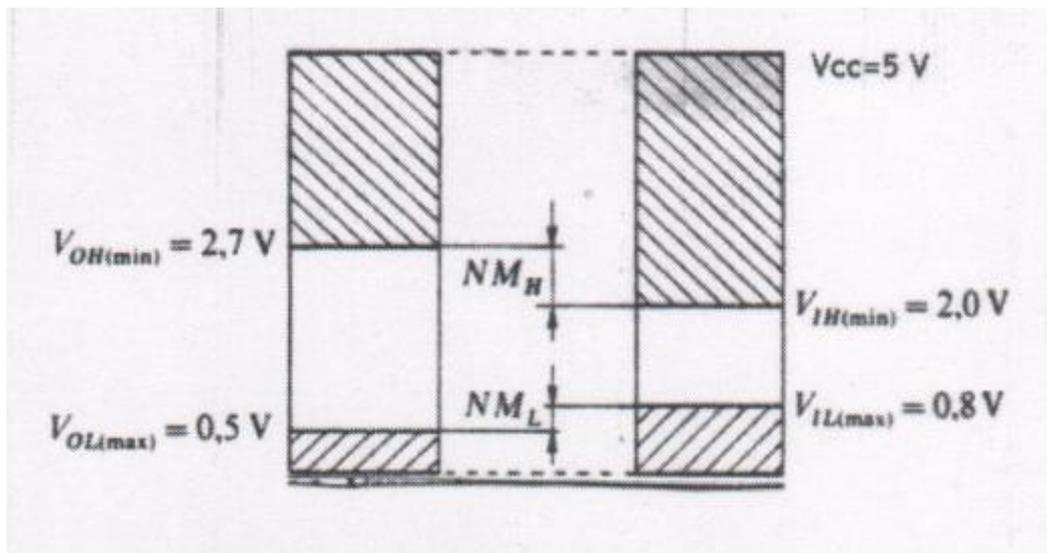


Fig. 2.4. NM_H e NM_L indicano, rispettivamente, il margine di rumore per il livello logico alto e basso.

e tra V_{OL} e V_{IL} è detta margine di rumore NM (Noise Margin) e garantisce che lo stato logico venga interpretato correttamente anche in presenza di disturbi. Se, per esempio, si ha un determinato margine di rumore $V_{OH} - V_{IH}$, si può decidere di caricare ancora un po' la porta logica, abbassando ulteriormente V_{OH} e rispettando sempre la condizione $V_{OH} > V_{IH}$. In questo modo, però, si ha un maggior rischio che un segnale molto disturbato possa creare dei problemi di comunicazione tra le porte. È il progettista che sceglie un determinato compromesso tra quanto caricare la porta logica e quanto margine di rumore avere. In generale, quanto maggiore è il margine di rumore, tanto maggiore è il range all'interno del quale il segnale può variare, senza che il circuito logico venga indotto in errore. I margini d'errore da valutare sono due: quello a livello logico alto e quello a livello logico basso, come si vede in Fig. 2.4. In fase di progettazione si considera quello minore in modo da mettersi nella condizione più restrittiva.

2.1.1) Confronto porte logiche TTL – CMOS

Le tecnologie di porte logiche più comunemente utilizzate sono la TTL (Transistor-Transistor Logic) e la CMOS (Complementary Metal-Oxide Semiconductor). La famiglia 74xx, appartenente alla tecnologia TTL, alimentata a 5 V, ha le seguenti caratteristiche in termini di livelli logici in ingresso e in uscita (Fig. 2.5):

- V_{OL} variabile tra 0,4 V e 0,5 V;
- V_{IL} pari a 0,8 V;
- V_{OH} variabile tra 2,4 V e 2,7 V;
- V_{IH} pari a 2 V;

Questi valori forniscono un margine di errore di almeno 0,4 V – 0,7 V per il livello logico alto e di 0,3 V-0,4 V per il livello logico basso, in caso di comunicazione tra porte logiche di questo stesso tipo [1].

Nella tecnologia CMOS, famiglia 4000, i valori in input minimi tali da essere interpretati come livello logico alto variano tra 3 V e 3,5 V. Si capisce, quindi, come sia più difficoltoso mettere in comunicazione porte logiche TTL in output con porte CMOS in input. I componenti CMOS, infatti,

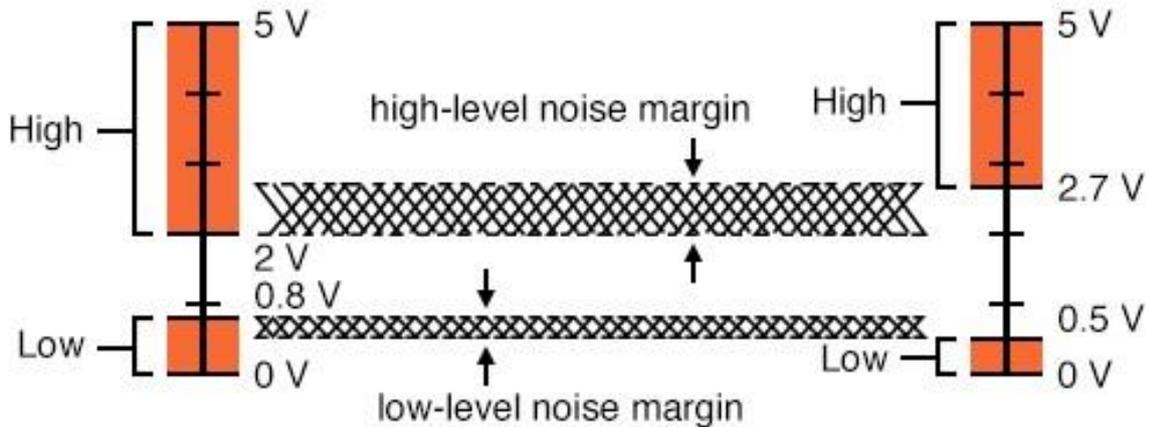


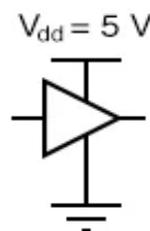
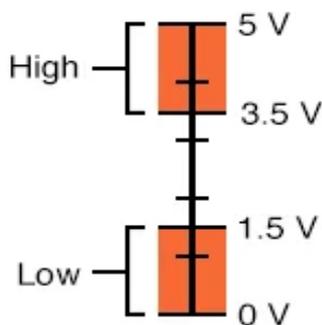
Fig. 2.5. Livelli logici e margini di errore tra porte logiche TTL.

hanno specifiche molto diverse da quelli TTL. Considerando una porta logica CMOS alimentata a 5 V, si ha un range di tensione accettabile in ingresso per il livello logico basso tra 0 V e 1,5 V, mentre per il livello logico alto il range varia tra 3,5 V e 5 V. Riguardo, invece, i valori di tensione di uscita, il range è tra 0 V e 0,05 V per il livello logico basso e tra 4,95 V e 5 V per il livello logico alto (Fig. 2.6).

Si deduce che le porte CMOS abbiano un maggior margine di rumore rispetto alle TTL: 1,45 V sia per il margine del livello logico basso che alto contro un massimo di 0,7 V per le TTL. In pratica, le porte CMOS possono tollerare oltre il doppio del rumore sovrapposto al segnale in ingresso nelle linee di input, prima che venga mal interpretato, rispetto alla tecnologia TTL. Il margine di rumore diventa ancora più grande se si lavora a tensioni di alimentazione più alte: le porte CMOS possono essere alimentate fino a tensioni di 15 V (alcune arrivano anche a 18 V), a differenza delle TTL che possono essere alimentate solo a 5 V. Con $V_{AL} = 15$ V, si hanno i seguenti valori (Fig. 2.7):

- $V_{IH} = 11$ V;
- $V_{IL} = 4$ V;
- $V_{OH} = 14,95$ V;
- $V_{OL} = 0,05$ V [1].

Acceptable CMOS Gate
Input Signal Levels



Acceptable CMOS Gate
Output Signal Levels

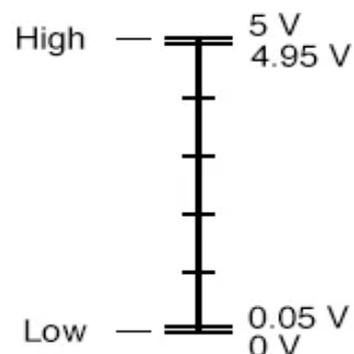


Fig. 2.6. Livelli logici porta logica CMOS alimentata a 5 V.

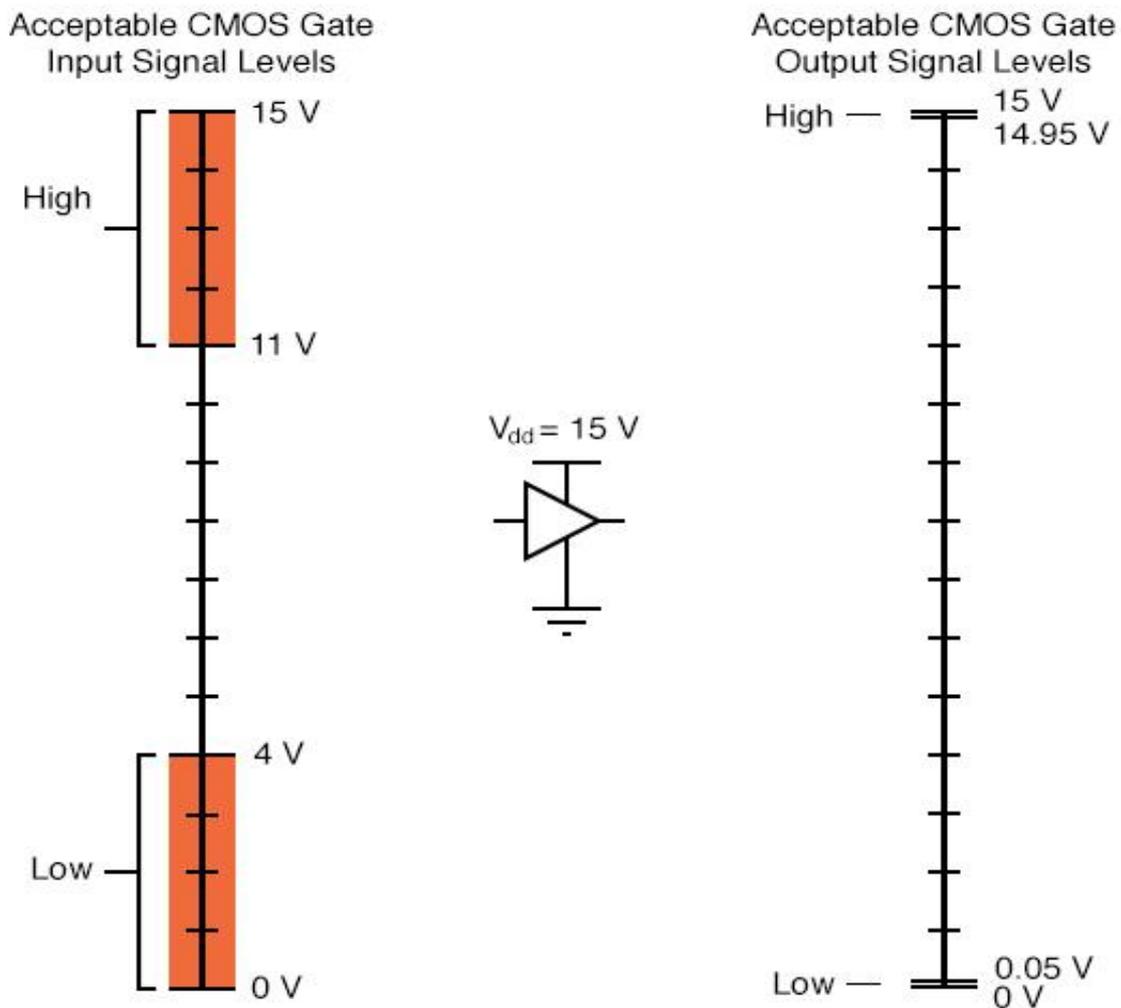


Fig. 2.7. Livelli logici portalogica CMOS alimentata a 15 V.

Questi diversi livelli logici, per le due tecnologie in esame, possono recare problemi quando i componenti TTL e CMOS vengono usati nello stesso sistema. Tuttavia, si nota come l'unica situazione di incompatibilità si abbia quando è necessario collegare un output TTL in ingresso ad un componente CMOS. Considerando entrambe le porte alimentate a 5 V, se la TTL esce a livello logico basso tra 0 V e 0,5 V, quest'ultimo sarà ben interpretato dalla CMOS che vuole un intervallo per quel livello logico tra 0 V e 1,5 V. Il problema nasce quando la TTL esce a livello logico alto. In tal caso, la tensione in uscita è nell'intervallo compreso tra 5 V e 2,7 V, mentre la CMOS, affinché interpreti un livello logico alto in ingresso, si aspetta un intervallo compreso tra 5 V e 3,5 V. Questo può creare un'interpretazione errata dei livelli logici.

Per risolvere il problema, si va ad aumentare il livello del segnale in uscita dalla porta TTL utilizzando un resistore di pull-up, come mostrato in Fig. 2.8.

Questo non basta se la porta CMOS, con la quale si interfaccia la TTL, è alimentata ad una tensione superiore rispetto ai 5 V richiesti dalla TTL, per esempio 10 V. In quel caso, ancora una volta, non si hanno problemi per la corretta interpretazione del livello logico basso; mentre, l'intervallo 2,7 V – 5 V dell'uscita TTL, non combacia con quello 7 V – 10 V dell'ingresso CMOS. Il problema si

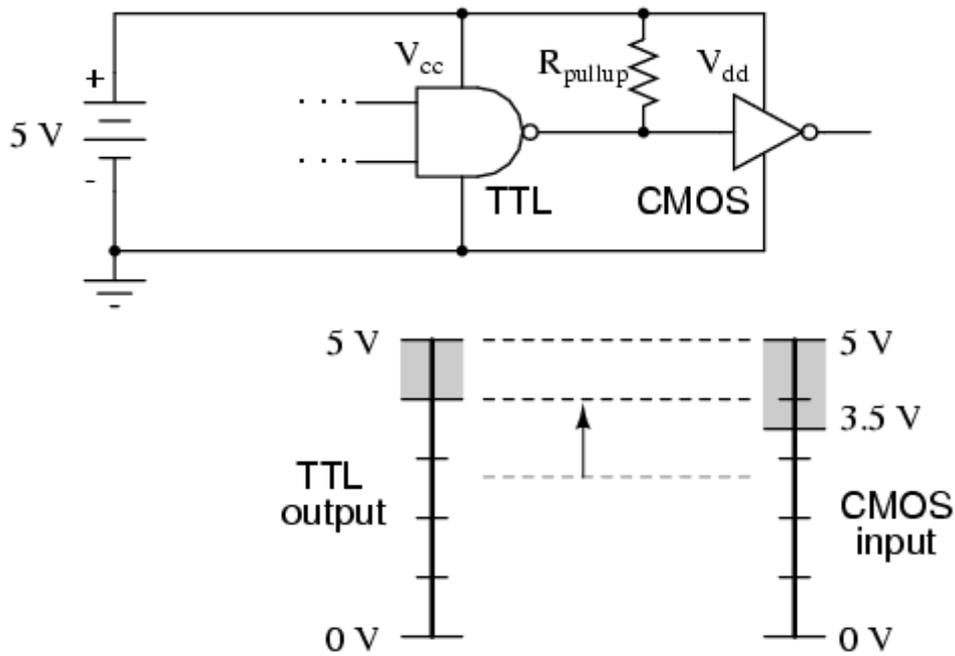


Fig. 2.8. Resistore di pull-up inserito tra le due porte logiche.

risolve usando, invece che uno stadio di uscita totem-pole della porta, un'uscita open-collector con un resistore di pull-up collegato alla linea di alimentazione di 10 V, per aumentare il livello del segnale di uscita e renderlo compatibile col range d'ingresso CMOS [1]. L'esempio appena descritto è mostrato in Fig.2.9. Tuttavia, una spiegazione relativa alle porte open-collector e del ruolo del resistore di pull-up verrà data nel paragrafo successivo.

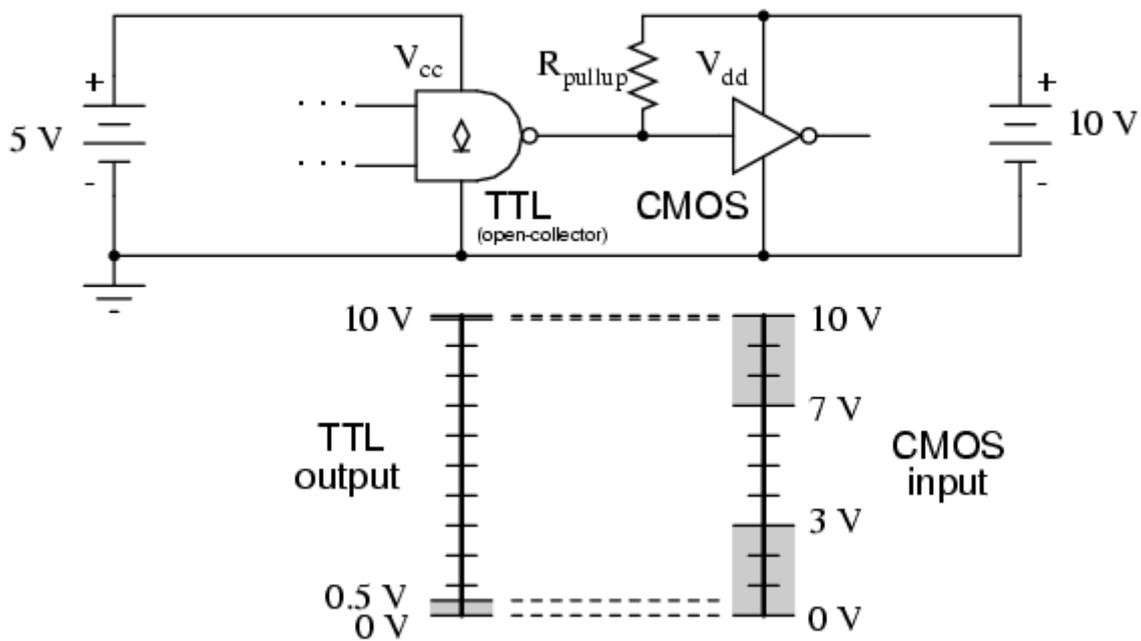


Fig. 2.9. Porta d'uscita open-collector TTL.

2.2) Porte open-collector

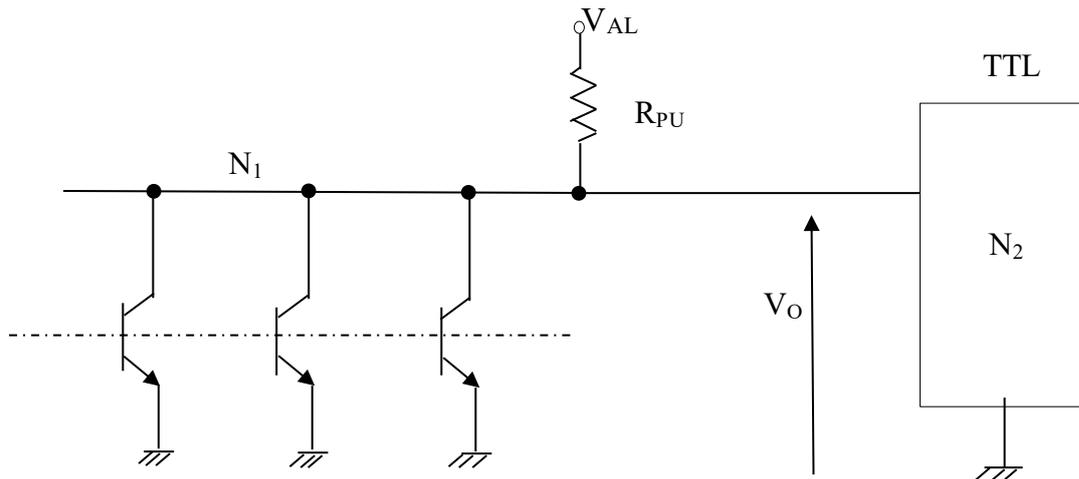


Fig. 2.10. N_1 porte open-collector che pilotano N_2 porte logiche TTL.

Le porte open-collector sono utilizzate per pilotare carichi che richiedono una certa potenza o anche per collegarsi a circuiti o chip che sono alimentati a tensioni diverse rispetto al dispositivo logico di pilotaggio. Nell'esempio rappresentato in Fig.2.10, ci sono N_1 porte open-collector collegate allo stesso resistore R_{PU} che pilotano N_2 porte TTL. V_O è la tensione d'uscita dell'insieme delle porte open-collector che deve essere compatibile con V_{IL} e V_{IH} delle porte pilotate. Si devono, quindi, fare delle scelte di progetto, tali per cui V_O sia ben interpretata dai livelli logici in ingresso al blocco TTL, ovvero tali che $V_{OH} > V_{IH}$ e $V_{OL} < V_{IL}$. Solitamente, la configurazione che si vede in Fig. 2.10 viene utilizzata per implementare la logica cablata: wired-AND o wired-OR. Nel primo caso, se tutte le porte forniscono un'uscita a livello logico alto, l'uscita comune a tutte le porte open-collector è a livello logico alto e, quindi, il comportamento è come quello di una porta logica AND. Nel secondo caso, se almeno un'uscita tra quelle porte va a livello logico basso, l'uscita comune a tutte le porte open-collector viene forzata a livello logico basso, comportandosi come una porta logica OR.

Una porta open-collector è un tipo di porta logica che ha come uscita fisica il collettore del transistor a giunzione bipolare (BJT = Bipolar Junction Transistor). Il transistor viene fatto lavorare in interdizione e in saturazione, per le quali si hanno, rispettivamente, il livello logico alto e basso in uscita. Il BJT in zona di interdizione ha entrambe le giunzioni B-E e B-C polarizzate inversamente. La corrente di collettore I_C che va dal collettore all'emettitore è molto bassa, solitamente di centinaia di μA , ed è detta corrente inversa di saturazione (I_{COsat}); in questa situazione il transistor si comporta come un interruttore aperto. In zona di saturazione, le due giunzioni sono polarizzate direttamente e la corrente I_C non dipende più dalla corrente di base I_B , ma dalla maglia collettore-emettitore. E' come se tra il collettore e l'emettitore ci fosse un cortocircuito, anche se una piccola tensione permane sempre che è all'incirca di qualche centinaia di mV, indicata come V_{CEsat} . In questa situazione il transistor si comporta come un interruttore chiuso. Si può definire, dunque, I_{OH} la corrente di uscita dalla porta open-collector, che coincide con I_{COsat} , quando l'uscita è a livello logico alto, ossia quando il transistor è in interdizione. Si definisce, poi, I_{OL} la corrente che entra nel collettore della porta quando il livello logico è basso, ossia il transistor è in saturazione. Questa è la massima corrente che può scorrere nel transistor quando si trova nella suddetta condizione.

Uno dei passi più importanti nell'utilizzo di porte open-collector è il dimensionamento del resistore di pull-up. Questo, come si è visto nel Par. 2.1.1, ha il compito di permettere una corretta interpretazione degli stati logici tra porte differenti, come TTL e CMOS e tra porte alimentate a tensioni differenti. Un'altra funzione fondamentale è quella di proteggere i transistor delle porte open-collector perché limita la corrente che scorre in ciascuno di essi nel momento in cui passano in saturazione. Infatti, la corrente che scorre nel collettore in quella situazione è: $I_C = \frac{V_{AL}}{R_{PU}}$. (In realtà, a V_{AL} andrebbe sottratta V_{CEsat} che, però, è molto più piccola rispetto alla tensione di alimentazione, quindi la si trascura). Questa corrente I_C deve essere minore di I_{OL} , ossia la corrente massima che può scorrere nel transistor saturo senza danneggiarlo. Si vede l'importanza del valore di R_{PU} che, quanto più è basso, tanto più alza il valore di I_C . Converrebbe, quindi, assegnare a R_{PU} un valore più alto possibile; tuttavia, esiste sempre un limite superiore da rispettare dato che, quando i transistor vanno in interdizione dove assorbono una certa corrente, ovvero I_{COsat} , se R_{PU} fosse troppo alto, si avrebbe un abbassamento troppo grande della tensione di uscita sulla porta V_O . Essa è data, infatti, dalla seguente equazione: $V_O = V_{AL} - R_{PU} I_{COsat}$. Quindi, R_{PU} va dimensionato, tenendo conto delle due seguenti condizioni:

- $I_C < I_{OL} = I_{Cmax}$;
- $V_{OH} > V_{IH}$.

2.2.1) Pilotaggio di un carico generico

In Fig. 2.11 è mostrato un esempio di pilotaggio di un carico generico R_L . Il carico, durante il suo funzionamento, richiede una certa tensione V_L e, di conseguenza, assorbe una certa corrente I_L . Dalla configurazione mostrata in figura è possibile scrivere:

$$V_{AL} = V_L + I_L R_{PU} \quad (2.1)$$

Questo è il caso in cui il transistor è interdetto e assorbe una certa I_{COsat} che viene trascurata, essendo molto bassa rispetto alla corrente assorbita dal carico, solitamente sempre di due o tre ordini di grandezza superiore a I_{COsat} . Quando il transistor passa in saturazione, sul carico si ha la tensione V_{CEsat} dell'ordine di qualche centinaio di mV; in tal caso il carico non lavora perché necessita della tensione V_L ai suoi capi e, di conseguenza, non assorbe la corrente I_L . Inoltre, come è stato detto in precedenza, in quest'ultima situazione è presente la corrente di collettore I_C che non deve superare un certo valore massimo per quella porta logica open-collector.

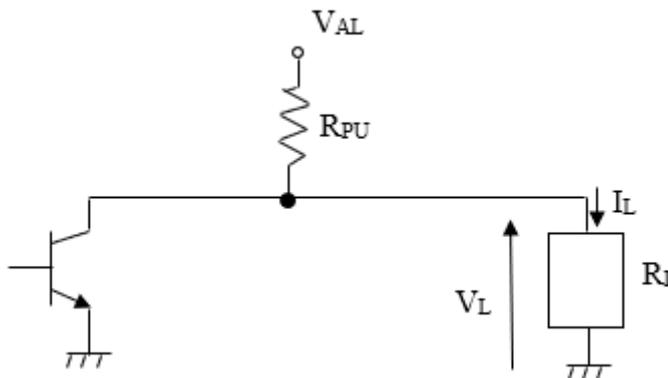


Fig. 2.11. Pilotaggio di un carico generico con porta open-collector.

Per dimensionare il valore del resistore di pull-up si considera la formula inversa della (2.1), tale per cui: $R_{PU} = \frac{V_{AL}-V_L}{I_L}$. Dato che V_L e I_L sono fissati in base al carico in questione, il parametro fondamentale per dimensionare R_{PU} è V_{AL} . Una volta fissato V_{AL} si avrà un certo valore della resistenza che deve anche soddisfare $I_C = \frac{V_{AL}}{R_{PU}} \leq I_{Cmax}$ (come detto nel Par.2.2, a V_{AL} andrebbe sottratto il valore della tensione V_{CEsat} che, però, non si considera, in quanto trascurabile rispetto al valore di V_{AL} stesso). Sulla base di questo, si evince che R_{PU} può assumere un intervallo di valori: $\frac{V_{AL}}{I_{Cmax}} \leq R_{PU} \leq \frac{V_{AL}-V_L}{I_L}$.

Bisogna tener presente anche il discorso relativo al consumo in termini di potenza del dispositivo. Ci sono due potenze da considerare:

- La potenza erogata al carico, quando il transistor è interdetto: $P_{eRpu} = I_L^2 R_{PU}$;
- La potenza dissipata dal dispositivo, quando il transistor è saturo: $P_{dRpu} = I_C^2 R_{PU}$.

Si nota come V_{AL} , oltre ad essere un parametro importante per la resistenza R_{PU} , è anche rilevante per la potenza dissipata quando il transistor è saturo poiché, in questo caso, esiste una dipendenza quadratica tra la potenza dissipata e V_{AL} ; quindi, con una tensione di alimentazione troppo grande, si può avere un consumo del dispositivo troppo elevato.

Considerando che il carico venga pilotato con un'onda quadra avente un certo duty cycle $D.C. = \frac{T_1}{T} \cdot 100$, dove T è il periodo dell'onda quadra e T_1 è il tempo in cui l'onda quadra è a livello logico alto; è possibile esprimere la potenza totale dissipata sul resistore di pull-up:

$$P_{Rpu} = [I_L^2 R_{PU} \frac{D.C.}{100} + I_C^2 R_{PU} (1 - \frac{D.C.}{100})] \quad (2.2)$$

La potenza dissipata quando il carico è spento, solitamente, è maggiore di quella dissipata quando il carico è acceso, dato che il transistor in saturazione dissipa una maggior potenza rispetto a quando è in interdizione. Quindi, se il D.C. è piccolo, la situazione è sfavorevole perché, per ogni ciclo dell'onda quadra, si rimane per più tempo a livello logico basso, consumando di più. E' meglio avere un D.C. elevato per dissipare meno potenza sul resistore R_{PU} .

2.3) Problemi di alimentazione e condensatori di by-pass

Esistono dei problemi legati all'alimentazione di circuiti integrati che possono portare al malfunzionamento dell'intero circuito elettrico, generando disturbi e interferenze. Questi problemi sono legati, in particolare, alla lunghezza e alla larghezza dei collegamenti elettrici che ci sono tra i vari chip, comprendendo anche le piste di alimentazione e di ground.

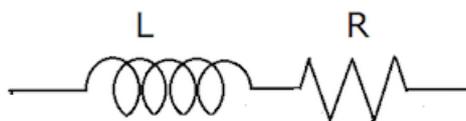


Fig. 2.12. Conduttore rettilineo.

Un conduttore non ideale, che ha una certa lunghezza l , diametro d e sezione S , possiede una certa resistenza R e un coefficiente di autoinduttanza L . Queste non idealità generano delle cadute di potenziale lungo la linea e lo fanno in modo diverso a seconda della natura del componente. Viene illustrato un esempio numerico. Si consideri che il conduttore abbia $l = 0,1$ mm e $S = 1$ mm². Il conduttore viene ipotizzato in rame, quindi ha una resistività $\rho = 0,018 \Omega \frac{\text{mm}^2}{\text{mm}}$. La formula della resistenza è: $R = \rho \frac{l}{S}$. La resistenza di quel conduttore, con le caratteristiche enunciate, è, all'incirca, pari a 1,8 m Ω . Se in quel conduttore scorresse una corrente di 50 mA, applicando la legge di Ohm, si avrebbe su di esso una caduta di potenziale di 100 μ V. Se quel conduttore fosse una pista che porta una tensione di alimentazione $V_{AL} = 5$ V ad un pin di un integrato, lungo la pista si perderebbero 100 μ V che, su 5 V, sarebbe un valore trascurabile. In generale, la resistenza di un conduttore, per lunghezze non esageratamente grandi, non crea molti problemi. Crea, invece, più problemi l'induttanza L nel momento in cui si ha a che fare con regimi variabili, mentre, in continua, si comporta come un cortocircuito. Il suo valore dipende molto dalle caratteristiche del conduttore come per la resistenza; si hanno, però, formule diverse, soprattutto in base alla forma geometrica del conduttore. Per semplicità, viene illustrata una formula relativa ad un conduttore cilindrico rettilineo:

$$L = 2l \left[\log_{10} \left(\frac{2l}{d} \right) - 0,75 \right] \quad (2.3)$$

Questa formula restituisce il valore di L in nH, esprimendo lunghezza e diametro in cm. Considerando un filo conduttore con $l = 10$ cm e $d = 0,1$ cm, si ottiene un coefficiente di autoinduttanza $L = 31$ nH. Si deduce che, affinché si abbia un minore valore di L , sia meglio utilizzare piste corte e larghe.

In Fig. 2.13 è rappresentata una situazione in cui si ha un generatore di tensione V_{AL} che va ad alimentare una porta logica idealmente rappresentata, nell'immagine, come un generatore di corrente, in quanto la porta assorbe corrente quando ai suoi capi vi è una tensione e che la manda a livello logico alto. Se il conduttore, che sarebbe il filo compreso tra i due puntini neri in figura, fosse ideale, la porta d'ingresso raffigurata avrebbe ai suoi capi la tensione di alimentazione senza alcun tipo di perdite. In realtà, una linea conduttrice avrà una certa resistenza R al passaggio di corrente e un'induttanza parassita L , la quale può essere considerata uniformemente distribuita su tutta la linea. Viene effettuato, ancora, un esempio numerico per capire al meglio le conseguenze generate sul circuito elettrico, in particolare dal parametro parassita L . Infatti, come si è già visto, la

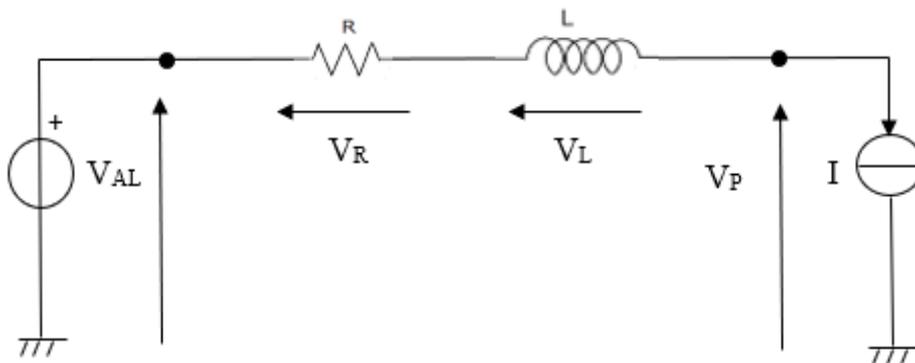


Fig. 2.13. Porta logica alimentata dalla tensione V_{AL} attraverso un conduttore rettilineo.

resistenza R genera una piccola caduta di tensione costante sui 5 V di alimentazione, che non crea, però, grandi problemi. Si va a considerare una situazione peggiore rispetto a quella precedente, in cui è stata calcolata $L = 31 \text{ nH}$, immaginando che, nella linea che collega l'alimentazione alla porta logica in Fig. 2.13, L sia uguale a 65 nH . La corrente assorbita dalla porta logica ha la forma di un'onda quadra che, non essendo ideale, ha un certo tempo di salita e di discesa. Si consideri un'onda quadra di 50 mA con tempo di salita/discesa pari a $\frac{dI}{dt} = 10 \frac{\text{mA}}{\text{ns}}$. A questo punto è possibile calcolare la caduta di potenziale sulla linea dovuta all'induttanza che si genera in corrispondenza della variazione di corrente da 0 mA a 50 mA e viceversa.

$$V_L = L \frac{dI}{dt} = 65 \text{ nH} \cdot 10 \frac{\text{mA}}{\text{ns}} = 650 \text{ mV}.$$

Questo vuol dire che, in corrispondenza del fronte di salita della corrente, l'induttanza genera un abbassamento della tensione prelevata dall'alimentazione di 650 mV sulla porta. Lo stesso fenomeno accade quando si verifica il fronte di discesa della corrente, dove l'induttanza genera un aumento di tensione di 650 mV . Poiché l'induttanza si oppone al passaggio di corrente, per un verso della corrente si ha una diminuzione di tensione, mentre, per l'altro, si ha un aumento. Quanto detto è rappresentato in Fig. 2.14: la tensione passa da 5 V a $4,35 \text{ V}$ nei nanosecondi del fronte di salita di corrente, si ristabilizza a 5 V perché, nei nanosecondi successivi, la corrente non varia e poi sale a $5,65 \text{ V}$ nei nanosecondi in cui vi è il fronte di discesa della corrente, ritornando, nuovamente, a 5 V . Questi picchi/ buchi di tensione sono dei veri e propri disturbi che causano:

- disturbi sulle porte logiche legate a quella che subisce la variazione di tensione, generando possibili problemi di compatibilità tra le porte o riduzioni del margine di rumore;
- la variazione di tensione è distribuita su tutta la linea e, se lungo questa ci sono altri circuiti integrati, questi subiscono il disturbo;
- le variazioni di tensione comportano una variazione di corrente lungo la linea, creando un campo magnetico che si concatena con le piste adiacenti, generando rumore nel circuito.

Il problema appena descritto può essere risolto inserendo dei condensatori, detti di by-pass, vicini al morsetto di alimentazione della porta logica. Se si inserisse un condensatore con capacità abbastanza grande vicino alla porta, quando c'è la richiesta di corrente, sarebbe il condensatore a

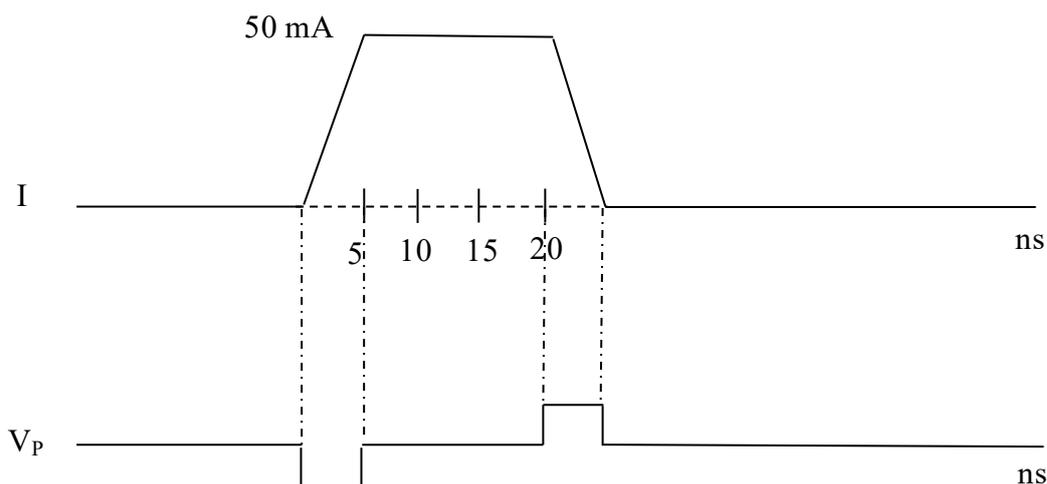


Fig. 2.14. Buco e picco di tensione in seguito alla variazione di corrente sulla linea.

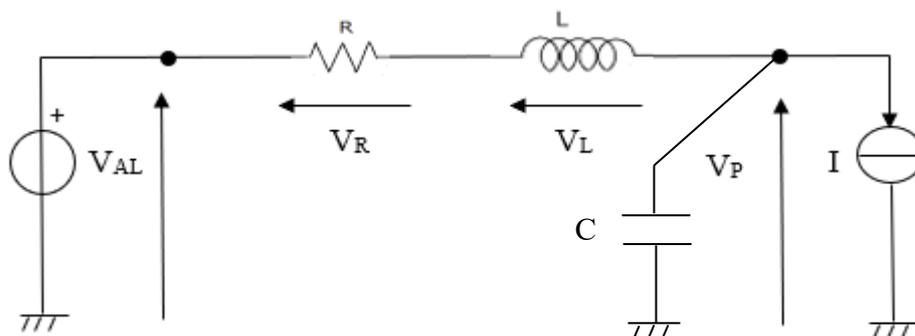


Fig. 2.15. Condensatore di by-pass C in corrispondenza della porta.

fornire la carica necessaria e non il generatore di tensione V_{AL} . In questo modo, la corrente non passerebbe attraverso L e non si presenterebbero i disturbi elettrici descritti. Nella Fig. 2.15 è mostrato il condensatore di by-pass C inserito nel circuito della Fig. 2.13.

La differenza di potenziale che si genera ai capi del condensatore è $\Delta V = \frac{\Delta Q}{C}$, dove Q è la carica e C è la capacità del condensatore. Scegliendo dei condensatori con capacità abbastanza grande, si hanno delle cadute di potenziale piccole in corrispondenza dei pin di alimentazione dell'integrato, dove i condensatori sono stati inseriti. Tuttavia, bisogna sempre considerare un limite superiore al valore della capacità, di cui si parlerà in seguito. Prima, si consideri il seguente esempio numerico. Prendendo in considerazione, ancora una volta, l'impulso di corrente in Fig. 2.14, per ottenere quell'onda quadra di 50 mA con durata di 20 ns, occorre una carica $Q = I\Delta t = 1 \text{ nC}$. Il tempo, in realtà, non è 20 ns perché l'impulso non è perfettamente rettangolare, ma, per semplicità, si considera tale e, di conseguenza, si avrà una carica Q leggermente sovrastimata. Supponendo che il condensatore abbia una capacità di 100 nF, essendo alimentato a 5 V, disporrà di una carica $Q = CV = 500 \text{ nC}$. Quindi, la carica messa a disposizione dal condensatore è ampiamente sufficiente per erogare la corrente richiesta. Si può, di conseguenza, calcolare la caduta di potenziale che genera il componente sulla linea in seguito alla richiesta di 1 nC per generare l'impulso di corrente: $\Delta V = \frac{\Delta Q}{C} = 10 \text{ mV}$. Si può notare come questo valore sia migliore rispetto ai 650 mV dovuti all'induttanza parassita; inoltre, non si hanno più buchi e picchi di tensione come prima, ma delle variazioni negative e positive di tensione con un andamento esponenziale, come mostrato in Fig. 2.16.

Per quanto riguarda la scelta del valore di capacità del condensatore, teoricamente non esiste un limite superiore ma, più il valore è alto, meno sono i problemi di disturbo elettrico. Tuttavia, vi sono dei limiti pratici. Innanzitutto, il primo limite è spaziale, dato che, più si scelgono capacità grandi, più i condensatori sono fisicamente grandi, andando ad occupare molto spazio sul circuito; quindi, se si vogliono realizzare dei circuiti piccoli e leggeri, bisogna porsi dei limiti di spazio. Dopodiché, si ha il problema del costo, perché, a parità di tecnologia, un condensatore con una maggiore capacità costa di più rispetto ad uno con capacità minore. Questo è un vincolo importante di cui

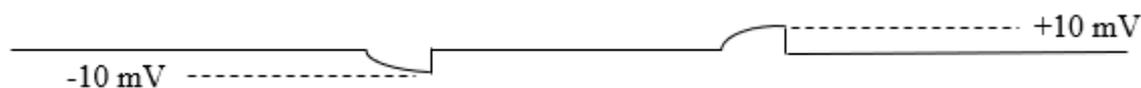


Fig. 2.16. Variazioni esponenziali di tensione sulla linea dovute al condensatore di by pass.

tener conto quando si producono dispositivi su larga scala. Infine, bisogna considerare il fatto che neanche i condensatori sono ideali: le due piastre sono collegate a due conduttori rettilinei che hanno un certo coefficiente di autoinduttanza. In continua e a basse frequenze queste induttanze non creano problemi, mentre, a frequenze più alte, si oppongono alle variazioni di corrente, non consentendo al condensatore di fornire la corrente necessaria alla porta ed opponendosi, quindi, alla scarica immediata che riesce a generare. Questa induttanza parassita pregiudica il funzionamento del condensatore e, a parità di tecnologia, maggiore è la capacità C e maggiore è l'induttanza parassita.

In sintesi, bisognerebbe utilizzare condensatori di by pass piccoli, poco costosi e con un bassissimo coefficiente di autoinduttanza. Si sfruttano, generalmente, i condensatori multistrato che hanno queste caratteristiche e arrivano a capacità di centinaia di nF. I condensatori di by pass vengono ormai utilizzati con valori di C compresi tra i 100 e 200 nF. Il valore tipico nei circuiti elettronici è 100 nF.

2.4) Linee di trasmissione

Una linea di trasmissione è essenzialmente una rete a due porte e ciascuna delle due porte è formata da due terminali, come mostrato in Fig. 2.17; una delle due è la porta di trasmissione e l'altra è la porta di ricezione [2]. La prima fornisce una tensione in uscita proveniente da una sorgente di qualsiasi tipo. Per la teoria dei circuiti, questa è rappresentabile come un circuito equivalente di Thevenin, formato da un generatore di tensione in serie con una resistenza equivalente del generatore. Il circuito collegato alla porta di ricezione della linea è detto carico, rappresentato da una resistenza di carico equivalente o da un'impedenza di carico nel caso di regime alternato. Una coppia di cavi costituisce effettivamente una linea di trasmissione ma il suo impatto sulle correnti e sulle tensioni dipende dalla lunghezza l della linea e dalla frequenza f del segnale emesso dal generatore. Il fattore che determina se sia possibile trascurare o meno gli effetti di una linea sul segnale trasmesso è $\frac{\omega l}{u}$, dove $\omega = 2\pi f$ è la pulsazione del segnale e u la sua velocità di propagazione. Sapendo che $u = f\lambda$, dove λ è la lunghezza d'onda del segnale, è possibile scrivere: $\frac{\omega l}{u} = \frac{2\pi f l}{u} = \frac{2\pi l}{\lambda}$. Quando il rapporto l/λ è molto piccolo si possono trascurare gli effetti dovuti alla linea di trasmissione ma quando $l/\lambda \geq 0,01$ può essere necessario tener conto non solo dello sfasamento associato al ritardo ma anche della presenza di segnali riflessi dal carico verso il generatore. Può anche essere necessario considerare gli effetti della dispersione e della perdita di potenza sulla linea [2].

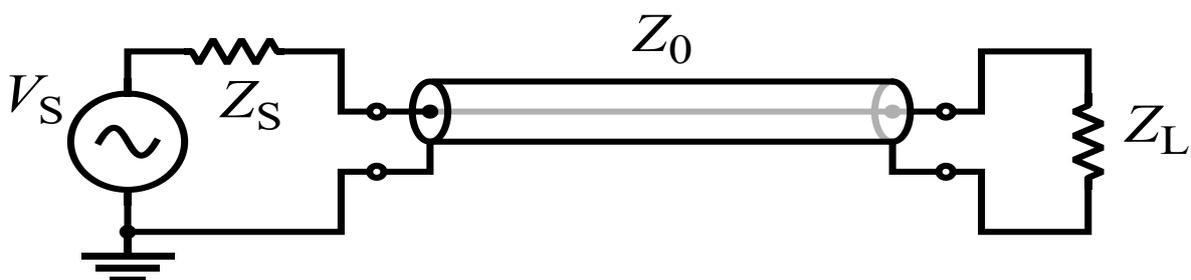


Fig. 2.17. Linea di trasmissione.

2.4.1) Modello a costanti concentrate

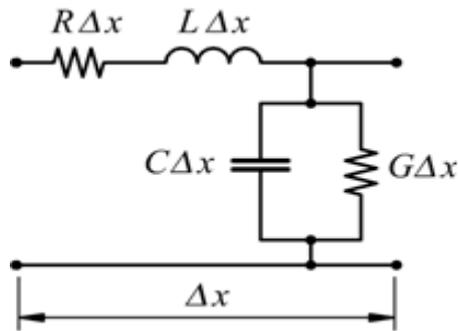


Fig. 2.18. Tratto infinitesimo Δx di una linea di trasmissione rappresentato con il modello a costanti concentrate.

Una qualsiasi linea di trasmissione si può rappresentare in configurazione a fili paralleli, a prescindere dalla sua forma specifica [2]. La linea è orientata lungo un asse, ad esempio x , che si suddivide in tratti infinitesimi di lunghezza Δx , dove ciascun tratto è rappresentato mediante un circuito equivalente, come illustrato in Fig.2.18. Questa rappresentazione è detta modello circuitale a costanti concentrate; è costituita dai seguenti parametri:

- R : la resistenza collettiva di entrambi i conduttori per unità di lunghezza (Ω/m);
- L : l'induttanza complessiva di entrambi i conduttori per unità di lunghezza (H/m);
- G : la conduttanza del mezzo isolante per unità di lunghezza (S/m);
- C : la capacità dei due conduttori per unità di lunghezza (F/m);

Nel modello vi sono due elementi in serie (R e L) e due elementi in parallelo (G e C). Per dare una spiegazione fisica del modello, si consideri un breve tratto di linea coassiale: essa è formata da un conduttore interno e da un conduttore esterno cilindrico separati da un materiale con permittività elettrica ϵ , permeabilità magnetica μ e conducibilità σ [2]. Quando un generatore di tensione viene collegato ai due conduttori della porta di trasmissione della linea, in essi scorrono delle correnti. La resistenza di linea R rende conto della resistenza complessiva per unità di lunghezza del conduttore interno ed esterno. La stessa cosa vale per l'induttanza L . La conduttanza in parallelo per unità di lunghezza G rende conto del flusso di corrente tra il conduttore interno e quello esterno, reso possibile dalla conducibilità σ del materiale isolante [2]. G è posto in parallelo perché il flusso di corrente è diretto da un conduttore all'altro. Infine, bisogna esaminare il parametro C : quando due cariche uguali e opposte sono posizionate su due conduttori arbitrari non in contatto tra loro, separati da un dielettrico, tra le cariche sarà indotta una differenza di potenziale.

2.4.2) Propagazione delle onde in una linea di trasmissione

Le equazioni che descrivono la tensione e la corrente attraverso la linea di trasmissione in funzione del tempo t e dello spazio x sono dette equazioni del telegrafo:

$$-\frac{\partial v(x,t)}{\partial x} = Ri(x,t) + L \frac{\partial i(x,t)}{\partial t} \quad (2.4)$$

$$-\frac{\partial i(x,t)}{\partial x} = Gv(x,t) + C \frac{\partial v(x,t)}{\partial t} \quad (2.5)$$

In condizioni di regime sinusoidale la (2.4) e la (2.5) possono essere espresse nella forma fasoriale:

$$-\frac{d\tilde{V}(x)}{dx} = (R + j\omega L)\tilde{I}(x) \quad (2.6)$$

$$-\frac{d\tilde{I}(x)}{dx} = (G + j\omega C)\tilde{V}(x) \quad (2.7)$$

Da queste è possibile ricavare l'equazione d'onda per $\tilde{V}(x)$ e $\tilde{I}(x)$, andando a derivare entrambi i membri della (2.6) e sostituendo al termine $\frac{d\tilde{I}(x)}{dx}$, che si ottiene dopo la derivazione, la (2.7). Con questo procedimento si arriva all'equazione:

$$\frac{d^2\tilde{V}(x)}{dx^2} - \gamma^2\tilde{V}(x) = 0 \quad (2.8)$$

dove

$$\gamma = \sqrt{(R + j\omega L)(G + j\omega C)} \quad (2.9)$$

Applicando lo stesso ragionamento per la (2.7) si ottiene anche l'equazione d'onda per $\tilde{I}(x)$.

$$\frac{d^2\tilde{I}(x)}{dx^2} - \gamma^2\tilde{I}(x) = 0 \quad (2.10)$$

γ è la costante di propagazione complessa della linea di trasmissione ed è costituita da una parte reale α , detta costante di attenuazione della linea, espressa in Np/m, e da una parte immaginaria β , detta costante di fase della linea, espressa in rad/m. Quindi $\gamma = \alpha + j\beta$. Le equazioni d'onda (2.8) e (2.10) hanno come soluzioni delle onde progressive che hanno la forma seguente [2]:

$$\tilde{V}(x) = V_0^+ e^{-\gamma x} + V_0^- e^{\gamma x} \quad (2.11)$$

$$\tilde{I}(x) = I_0^+ e^{-\gamma x} + I_0^- e^{\gamma x} \quad (2.12)$$

dove il termine $e^{-\gamma x}$ rappresenta la propagazione dell'onda in direzione delle x positive e il termine $e^{\gamma x}$ rappresenta la propagazione dell'onda in direzione delle x negative. Si possono mettere in relazione le ampiezze dell'onda di corrente, I_0^+ e I_0^- , con le ampiezze dell'onda di tensione V_0^+ e V_0^- , sostituendo la (2.11) nella (2.6) e ricavando la corrente $\tilde{I}(x)$ che risulta essere [2]:

$$\tilde{I}(x) = \frac{\gamma}{R + j\omega L} (V_0^+ e^{-\gamma x} - V_0^- e^{\gamma x}) \quad (2.13)$$

Confrontando i termini dell'espressione con i termini corrispondenti della (2.12) si può concludere che:

$$\frac{V_0^+}{I_0^+} = Z_0 = -\frac{V_0^-}{I_0^-} \quad (2.14)$$

dove è possibile definire l'impedenza caratteristica della linea Z_0 :

$$Z_0 = \frac{R + j\omega L}{\gamma} = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \quad (2.15)$$

Z_0 è uguale al rapporto tra l'ampiezza in tensione e l'ampiezza in corrente, relative a ciascuna delle singole onde propaganti, con un segno meno in caso di propagazione nel verso delle x negative [2]. A questo punto la (2.13) si può riscrivere in funzione di Z_0 nella forma

$$\tilde{I}(x) = \frac{V_0^+}{Z_0} e^{-\gamma x} - \frac{V_0^-}{Z_0} e^{\gamma x} \quad (2.16)$$

2.4.3) Linea di trasmissione senza perdite

In molte situazioni la linea di trasmissione si può progettare in modo da minimizzare le perdite ohmiche, scegliendo conduttori con conducibilità molto elevata e materiali dielettrici interposti con conduttività trascurabile. Si ottiene, in tal modo, che R e G assumono valori molto piccoli, tali che $R \ll \omega L$ e $G \ll \omega C$ [2]. Ponendo $R = G = 0$ nella (2.9) si ottiene che:

$$\gamma = \alpha + j\beta = j\omega\sqrt{LC} \quad (2.17)$$

il che significa che $\alpha = 0$ e $\beta = \omega\sqrt{LC}$. Applicando la condizione di assenza di perdite anche alla (2.15), si ricava l'impedenza caratteristica

$$Z_0 = \sqrt{\frac{L}{C}} \quad (2.18)$$

2.4.4) Coefficiente di riflessione in tensione

Ponendo $\gamma = j\beta$ per la linea senza perdite, le espressioni della tensione complessiva e della corrente complessiva nella linea date dalla (2.11) e dalla (2.12) divengono [2]

$$\tilde{V}(x) = V_0^+ e^{-j\beta x} + V_0^- e^{j\beta x} \quad (2.19)$$

$$\tilde{I}(x) = \frac{V_0^+}{Z_0} e^{-j\beta x} - \frac{V_0^-}{Z_0} e^{j\beta x} \quad (2.20)$$

Queste espressioni contengono due incognite, V_0^+ e V_0^- , ossia le ampiezze in tensione rispettivamente dell'onda incidente e dell'onda riflessa. Per determinare V_0^+ e V_0^- si considera la linea di trasmissione senza perdite nel circuito completo, che comprende un circuito generatore ai terminali di ingresso e un carico Z_L ai terminali di uscita, come in Fig. 2.17. Sul carico, il fasore di tensione ai suoi capi \tilde{V}_L e il fasore di corrente che lo attraversa \tilde{I}_L sono legati tramite l'impedenza di carico Z_L come segue: [2]

$$Z_L = \frac{\tilde{V}_L}{\tilde{I}_L} \quad (2.21)$$

Per ragioni pratiche, si consideri il riferimento della coordinata spaziale x in modo che $x = 0$ corrisponda alla posizione del carico. In tal caso si scrive la tensione \tilde{V}_L come la tensione complessiva sulla linea $\tilde{V}(x)$ data dalla (2.19) e la corrente \tilde{I}_L come $\tilde{I}(x)$ data dalla (2.20), entrambe valutate in $x = 0$:

$$\tilde{V}_L = V_0^+ + V_0^- \quad (2.22)$$

$$\tilde{I}_L = \frac{V_0^+}{Z_0} - \frac{V_0^-}{Z_0} \quad (2.23)$$

Sostituendo queste espressioni nella (2.21) si ottiene:

$$Z_L = \left(\frac{V_0^+ + V_0^-}{V_0^+ - V_0^-} \right) Z_0 \quad (2.24)$$

Ricavando V_0^- si ha [2]

$$V_0^- = \left(\frac{Z_L - Z_0}{Z_L + Z_0} \right) V_0^+ \quad (2.25)$$

Il rapporto tra l'ampiezza dell'onda di tensione riflessa e l'ampiezza dell'onda di tensione incidente al carico è noto come coefficiente di riflessione in tensione Γ [2].

$$\Gamma = \frac{V_0^-}{V_0^+} = \frac{Z_L - Z_0}{Z_L + Z_0} \quad (2.26)$$

e ricordando la (2.14), il rapporto tra le ampiezze delle correnti è

$$\frac{I_0^-}{I_0^+} = -\frac{V_0^-}{V_0^+} = -\Gamma \quad (2.27)$$

Il coefficiente di riflessione in tensione Γ dipende dall'impedenza di carico Z_L e dall'impedenza caratteristica della linea Z_0 ; quest'ultima, per una linea senza perdite, è un numero reale, come mostrato nella (2.18). Z_L , in generale, è una quantità complessa, come per esempio nel caso di un circuito RL serie, per il quale $Z_L = R + j\omega L$. Dunque, anche Γ può essere complesso: [2]

$$\Gamma = |\Gamma| e^{j\theta} \quad (2.28)$$

Dove $|\Gamma|$ è il modulo di Γ e θ è il suo angolo di fase.

Un carico è adattato alla linea se $Z_L = Z_0$, perché in tal caso non si ha riflessione da parte del carico stesso. Infatti, in questa condizione, $\Gamma = 0$ e $V_0^- = 0$.

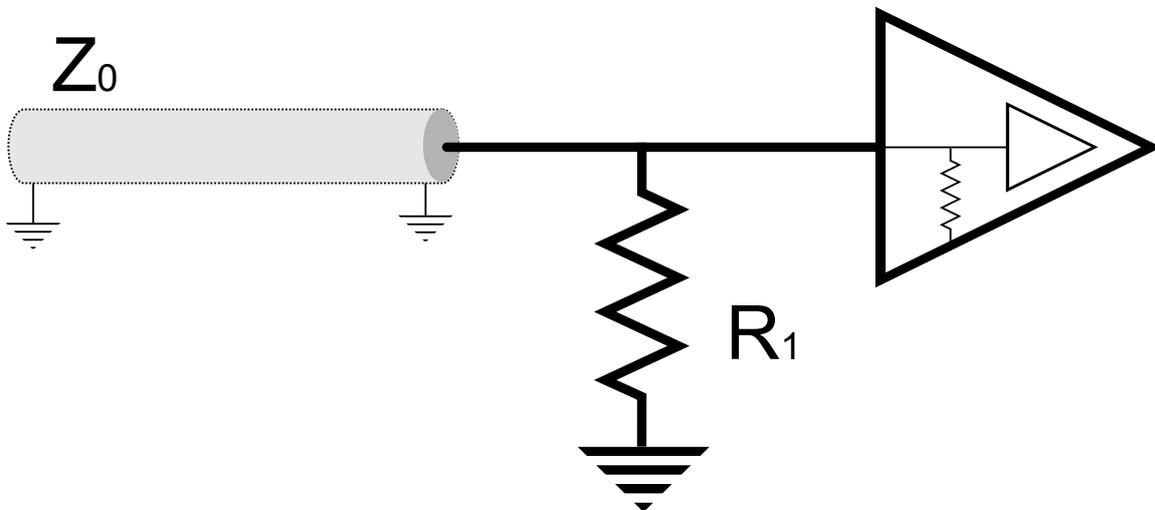


Fig. 2.19. Resistore di terminazione R_1 per l'adattamento della porta logica alla linea utilizzata con impedenza caratteristica Z_0 .

Si trova questo problema anche in sede di progettazione di circuiti elettronici. Nei casi in cui vi siano segnali digitali ad alta frequenza in uscita da una porta logica, che vengono mandati in ingresso ad un'altra porta attraverso cavi o connettori particolari, occorre inserire un resistore di terminazione, come R_1 nella Fig. 2.19, tra il pin di input della porta in ingresso e la tensione di riferimento, in modo tale che vi sia adattamento del carico alla linea utilizzata.

2.4.5) Impedenza d'ingresso in linea senza perdite

L'impedenza d'ingresso Z_{in} della linea è definita come il rapporto tra la tensione complessiva dell'onda incidente e di quella riflessa e la corrente complessiva in un arbitrario punto x della linea, diversamente dall'impedenza caratteristica della linea Z_0 , che dipende dalla tensione e dalla corrente di ciascuna delle due onde prese individualmente (incidente e riflessa) [2].

Dal punto di vista del circuito generatore, la linea di trasmissione può essere sostituita da un'impedenza pari a Z_{in} , come mostrato in Fig. 2.20. Il fasore di tensione ai capi di Z_{in} si può scrivere come:

$$\tilde{V}_i = \tilde{I}_i Z_{in} = \frac{\tilde{V}_g Z_{in}}{Z_g + Z_{in}} \quad (2.29)$$

Sostituendo la relazione $V_0^- = \Gamma V_0^+$ (ottenibile dalla 2.27) nella (2.19) si ottiene l'espressione

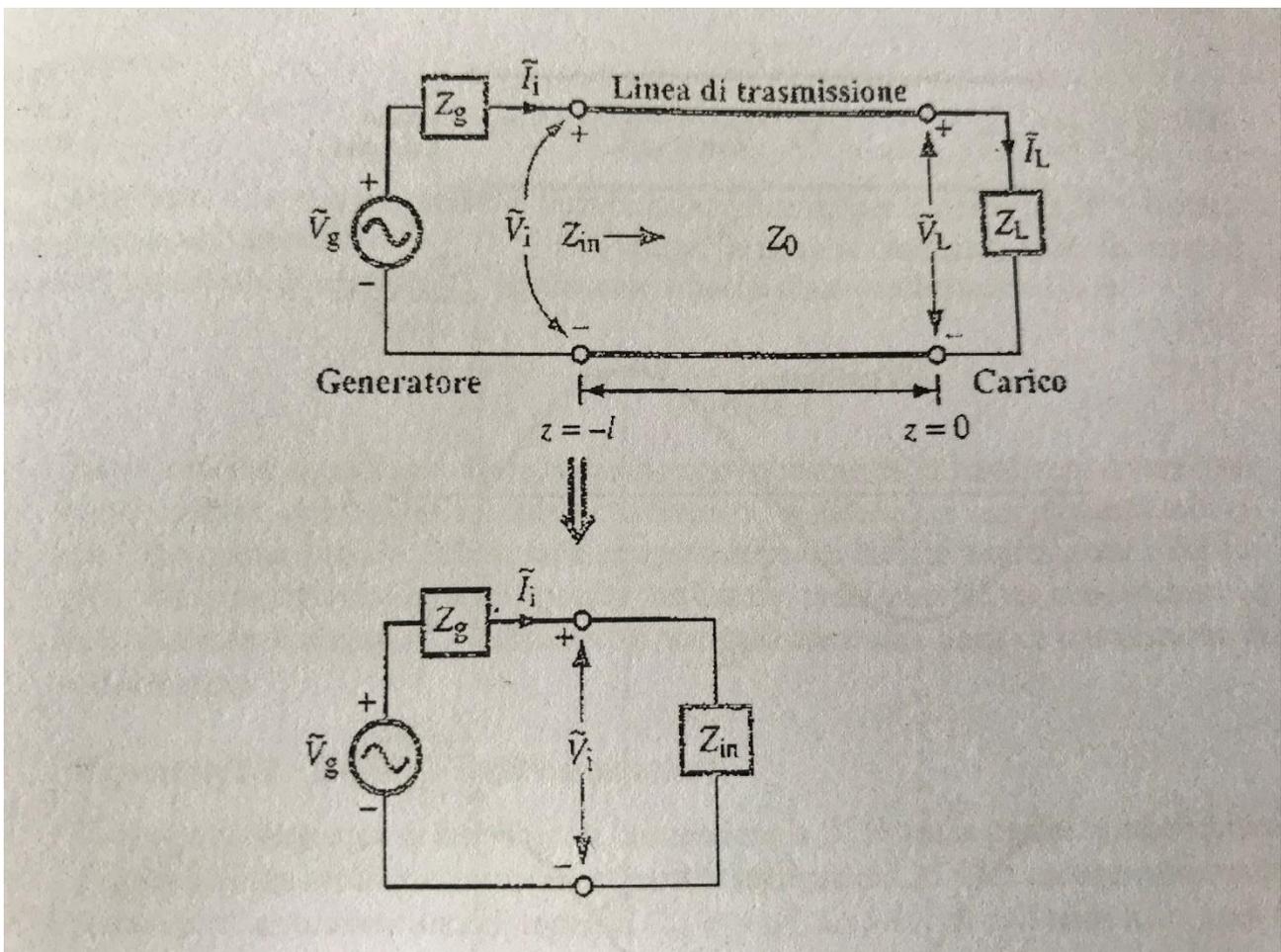


Fig. 2.20. Linea di trasmissione terminata sostituita con l'impedenza d'ingresso della linea Z_{in} [2].

$$\tilde{V}(x) = V_0^+ (e^{-j\beta x} + \Gamma e^{j\beta x}) \quad (2.30)$$

Dal punto di vista della linea di trasmissione, la tensione ai capi della sua sezione di ingresso è data dalla (2.30) calcolata in un punto generico x della linea pari a $-l$.

$$\tilde{V}_i = \tilde{V}(-l) = V_0^+ (e^{j\beta l} + \Gamma e^{-j\beta l}) \quad (2.31)$$

Uguagliando la (2.29) e la (2.31) e ricavando V_0^+ si ottiene:

$$V_0^+ = \left(\frac{\tilde{V}_g Z_{in}}{Z_g + Z_{in}} \right) \left(\frac{1}{e^{j\beta l} + \Gamma e^{-j\beta l}} \right) \quad (2.32)$$

FPGA

Le FPGA sono chip di silicio con porte logiche non connesse. È possibile definire la funzionalità dell'FPGA utilizzando un software per configurare le porte FPGA. Si sviluppano funzioni di computazione nel software che vengono compilate per generare un file di configurazione, chiamato bitstream, contenente informazioni su come le porte debbano essere collegate insieme [3]. Inoltre, le FPGA sono completamente riconfigurabili e assumono immediatamente una nuova "personalità" quando si imposta una diversa configurazione della circuiteria.

3.1) Architettura dell'FPGA

L'architettura di base della maggior parte delle FPGA è mostrata nella Fig. 3.1, basata su una matrice di blocchi hardware di base configurabili, detti blocchi logici (LB), circondati da blocchi input/output (IOB) che permettono all'FPGA di comunicare con l'esterno. L'insieme di tutti gli LB in un dato dispositivo viene solitamente definito come "logica distribuita". Oltre a quest'ultima, volta a supportare lo sviluppo di funzioni custom, le FPGA includono blocchi hardware specializzati e finalizzati all'implementazione di funzioni richieste in molte applicazioni pratiche. Esempi di queste risorse specifiche sono i blocchi di memoria, i blocchi di gestione del clock, i circuiti aritmetici, i serializzatori / deserializzatori, i ricetrasmittitori ed i microcontrollori [4]. Lo svantaggio principale di questa architettura a matrice riguarda il numero di IOB richiesti in un dato dispositivo dato che, più la logica è distribuita e specializzata, maggiore è il numero di IOB necessari e maggiori sono i costi. Un'altra limitazione di questa architettura deriva dal fatto che l'alimentazione e i pin di ground si trovano nella periferia dei dispositivi e, quindi, le cadute di tensione si verificano inevitabilmente quando le correnti fluiscono al / dal nucleo da / a questi pin.

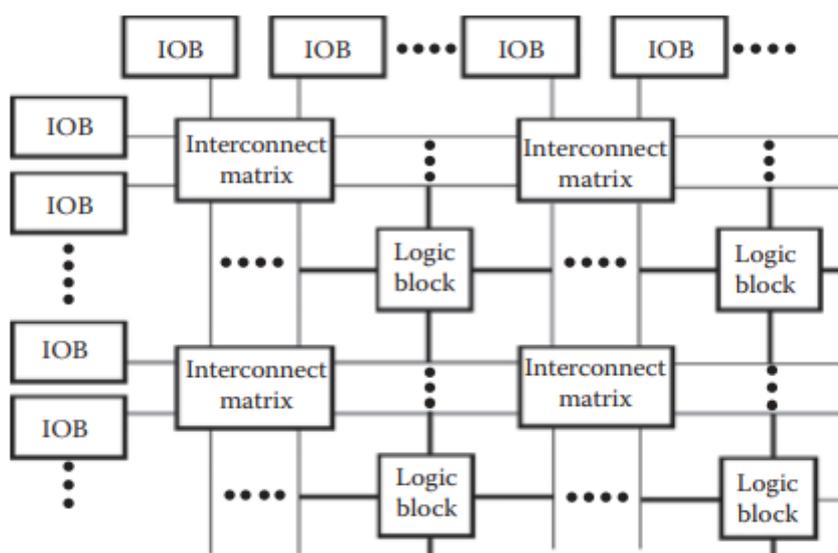


Fig. 3.1. Architettura di base FPGA [4].



Fig. 3.2. Architettura basata su colonne [4].

Una terza limitazione è che la capacità di scalare blocchi hardware specializzati dipende dalla quantità di logica distribuita disponibile nelle loro vicinanze [4]. Per mitigare queste limitazioni, i fornitori hanno sviluppato architetture FPGA basate su colonne (Xilinx 2006; Altera 2015b) come quella illustrata nella Fig. 3.2.

In primo luogo, in queste architetture, non c'è dipendenza del numero di IOB dalla quantità di logica distribuita e specializzata perché diversi tipi di risorse sono collocati in colonne dedicate e indipendenti. Questo significa che gli IOB si trovano nelle loro colonne corrispondenti e non solo nella periferia, e che il numero di IOB dipende solamente dal numero di pin I/O che il fornitore decide il dispositivo abbia. L'alimentazione e i pin di ground sono distribuiti in tutta l'area del chip, riducendo così al minimo i problemi di integrità del segnale [4].

3.2) Risorse hardware di base

Secondo la Fig. 3.1, le risorse hardware di base delle FPGA sono gli LB, IOB e risorse di interconnessione [4].

3.2.1) Blocchi logici

Gli LB hanno lo scopo di implementare funzioni custom combinatorie e sequenziali. Di conseguenza, sono costituiti principalmente da funzioni combinatorie, riconfigurabili ed elementi di memoria (flip-flop / latch). La parte combinatoria può essere implementata in diversi modi ma, oggi, le look-up table (LUT) sono gli elementi combinatori più utilizzati degli LB. Una LUT può essere pensata come un modo per memorizzare la tabella di verità della funzione combinatoria in modo tale che, quando si utilizzano gli ingressi di quella funzione come bit di indirizzo della LUT, il bit di memoria, che memorizza il valore della funzione per ogni particolare combinazione di input, può essere letto all'uscita della LUT. Le differenze tra LB di diversi fornitori (o diverse famiglie FPGA dello stesso fornitore) si riferiscono sostanzialmente al numero di ingressi delle

LUT (che definiscono il numero massimo di variabili logiche da cui può dipendere la funzione combinatoria implementata nella LUT), il numero di elementi di memoria e le capacità di configurazione degli LB [4].

3.2.2) Blocchi I/O

Gli IOB fungono da collegamenti tra i pin del dispositivo e le risorse interne. I loro elementi principali sono buffer bidirezionali programmabili e flip-flop per sincronizzare i segnali di I/O e di configurazione. IOB con diversi livelli di complessità sono disponibili nelle diverse famiglie di dispositivi FPGA correnti. Tuttavia, condividono tutti alcune caratteristiche comuni:

- I dati di input possono essere collegati direttamente alle risorse interne o passare attraverso un elemento di memoria. Allo stesso modo, i dati di output possono passare attraverso un elemento di memoria o bypassarlo.
- Gli elementi di memoria possono essere configurati come flip-flop o latch.
- I buffer bidirezionali supportano diversi livelli di tensione (1,2, 1,5, 1,8, 2,5, 3,0 e 3,3 V) e diversi standard I/O (single-ended, differenziale o con riferimento di tensione).

Gli IOB sono raggruppati in banchi che condividono risorse comuni e, di solito, dettagli di configurazione. Ciascun banco nell'FPGA può essere configurato per supportare uno standard di I/O diverso. Poiché ogni standard ha le proprie specifiche per tensioni, correnti, tipi di buffer e tipi di terminazione, la capacità di adattare la stessa FPGA per utilizzare simultaneamente diversi standard di I/O consente di collegarlo a circuiti che operano in condizioni elettriche diverse, senza necessità di circuiti di condizionamento esterni. Questo semplifica la progettazione di PCB, riduce i tempi di progettazione e, notevolmente, anche i costi.

- Controllo programmabile della corrente di uscita per alcuni standard di I/O.
- Controllo programmabile della velocità di variazione dell'uscita (salita e discesa).
- Resistori di pull-up e pull-down programmabili.
- Linee di ritardo programmabili.
- Tensione differenziale di uscita programmabile.

Fanno parte degli IOB anche i blocchi SerDes (Serializzatori e Deserializzatori). Questi sono circuiti di conversione seriale-parallelo (deserializzatore in input) e parallelo-seriale (serializzatore in output) che servono per interfacciare sistemi digitali con collegamenti che sfruttano la comunicazione seriale. Facilitano notevolmente l'implementazione di sistemi con requisiti di velocità di trasferimento dati elevata [4]. In alcune FPGA, i blocchi SerDes possono funzionare solo con segnali differenziali; ovvero, possono essere utilizzati solo quando gli IOB corrispondenti sono configurati per funzionare in uno standard di I/O differenziale. In altri dispositivi, possono funzionare sia con segnali single-ended che differenziali [4]. I blocchi SerDes possono supportare diverse modalità operative e lavorare a velocità di trasferimento dati differenti.

3.2.3) Risorse di interconnessione

Le risorse di interconnessione formano una maglia di linee dislocate su tutto il dispositivo, finalizzate a collegare gli ingressi e le uscite di tutti gli elementi funzionali dell'FPGA, come LB e IOB. Sono distribuiti tra le righe e le colonne di elementi funzionali, come mostrato nella Fig. 3.3.

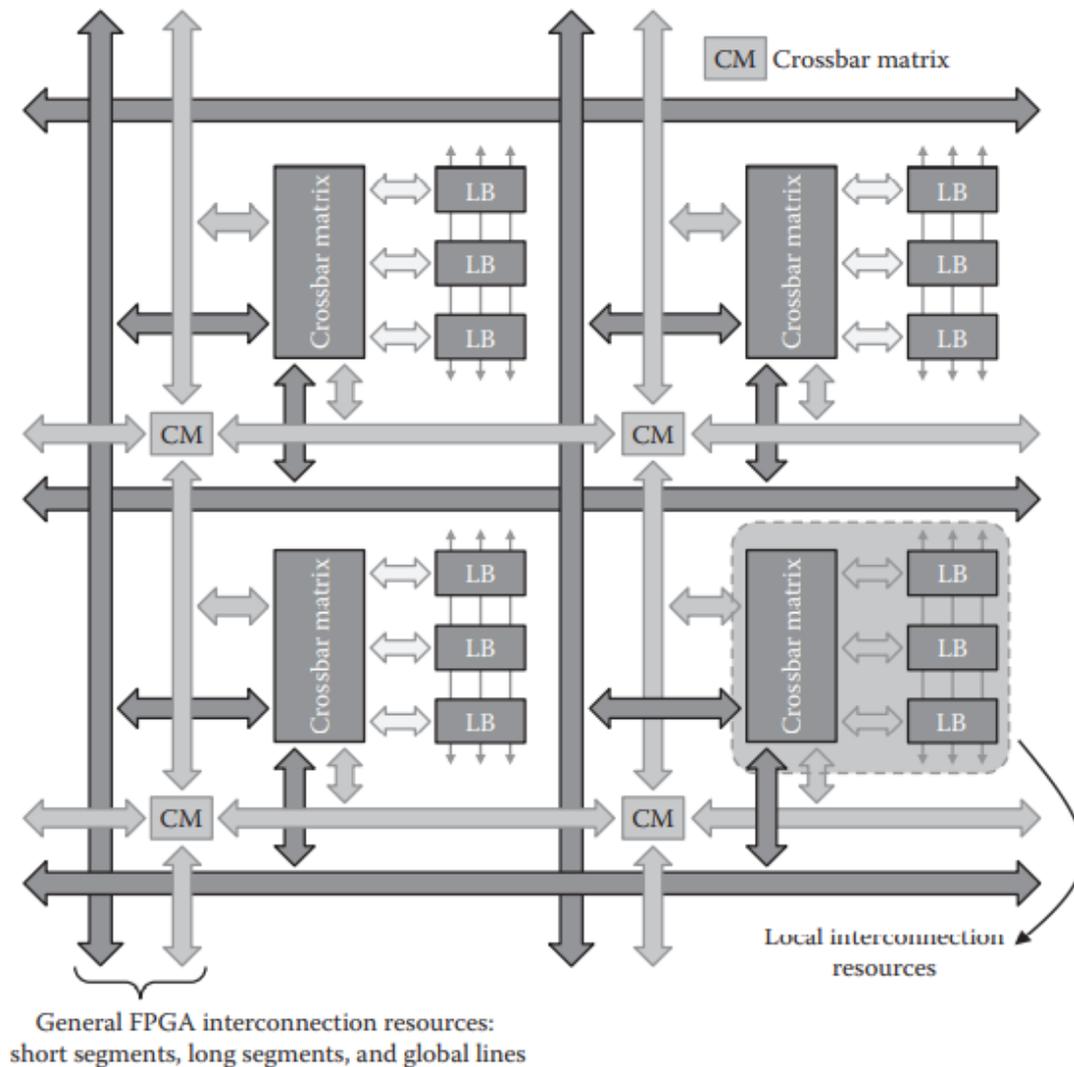


Fig.3.3. Risorse di interconnessione FPGA [4].

Le linee di interconnessione sono suddivise in segmenti, cercando di ottenere tempi di ritardo di propagazione dell'interconnessione minimi, in base alla posizione degli elementi da collegare. Sono presenti segmenti con lunghezze diverse, a seconda che siano destinati a collegare elementi vicini tra loro (segmenti corti) o in aree distanti del dispositivo (segmenti lunghi). Per segnali specifici che prevedono un elevato fan-out (il fan-out indica il numero di porte logiche che possono essere collegate all'uscita di una porta), come segnali di clock, set o reset, vengono utilizzate linee speciali, che coprono l'intero dispositivo o sue ampie regioni. Ad esempio, i segnali di clock hanno un'infrastruttura di interconnessione dedicata, composta da linee globali e gruppi di linee regionali, dove ciascun gruppo è associato a un'area specifica del dispositivo. Affinché si possa realizzare una particolare interconnessione, più segmenti sono collegati in serie, mediante le cosiddette “matrici crossbar”. Poiché gli LB sono le risorse più abbondanti e hanno un numero significativo di segnali di ingresso e uscita, di solito vengono collegati prima ad una “matrice crossbar” dedicata, che è una risorsa di interconnessione locale, condivisa da molti LB, e da questa alle risorse di interconnessione FPGA generali [4].

I ritardi di interconnessione sono un fattore critico nelle prestazioni delle FPGA, che dipendono dal tipo di risorse utilizzate, dal numero di matrici da attraversare e dalla distanza che deve essere percorsa dai segnali. Per questo motivo, il posizionamento dei blocchi funzionali di un dato circuito nell'FPGA è un fattore chiave per ottenere le migliori prestazioni possibili.

3.3) Blocchi hardware specializzati

Diversi tipi di blocchi hardware specializzati sono disponibili nella maggior parte delle FPGA attuali e il loro numero varia da un dispositivo all'altro. In realtà, il tipo e il numero di risorse hardware specializzate incluse in un dato dispositivo determinano il suo target applicativo [4].

3.3.1) Blocco di gestione del clock

La generazione, il controllo e la qualità dei segnali di clock sono tra i problemi più importanti da affrontare nella progettazione di sistemi digitali complessi e, in particolare, nel caso di sistemi che richiedono velocità di trasferimento dati molto elevate, dove la sincronizzazione tra le diverse parti del sistema è un problema critico. Per quanto riguarda la gestione del clock, le FPGA sono suddivise in regioni progettate per ridurre al minimo i ritardi di propagazione del clock al loro interno. A ciascuna regione viene assegnato un set di pin d'ingresso clock dedicati, insieme alle risorse per gestire e distribuire i segnali di clock, come mostrato nella Fig. 3.4.

Il numero di regioni clock dedicate variano a seconda delle dimensioni del dispositivo. Esistono anche linee di clock globali, così come altre linee di clock che collegano regioni di clock adiacenti. Al fine di ridurre i problemi associati ai segnali di clock e al numero di oscillatori esterni necessari, le FPGA includono i blocchi di gestione del clock (CMB) [4]. Questi CMB vengono utilizzati principalmente per la sintesi di frequenza (moltiplicazione o divisione della frequenza), riduzione dello skew (ossia il ritardo di ricezione di un bit rispetto al suo istante di trasmissione) e controllo del duty cycle dei segnali di clock. Ciascun CMB è associato a uno o più ingressi di clock dedicati e, nella maggior parte dei dispositivi, può anche prendere in ingresso un segnale di clock globale interno o l'uscita di un altro CMB (collegamento a catena). Le connessioni a catena consentono di aumentare il range dinamico per la sintesi di frequenza.

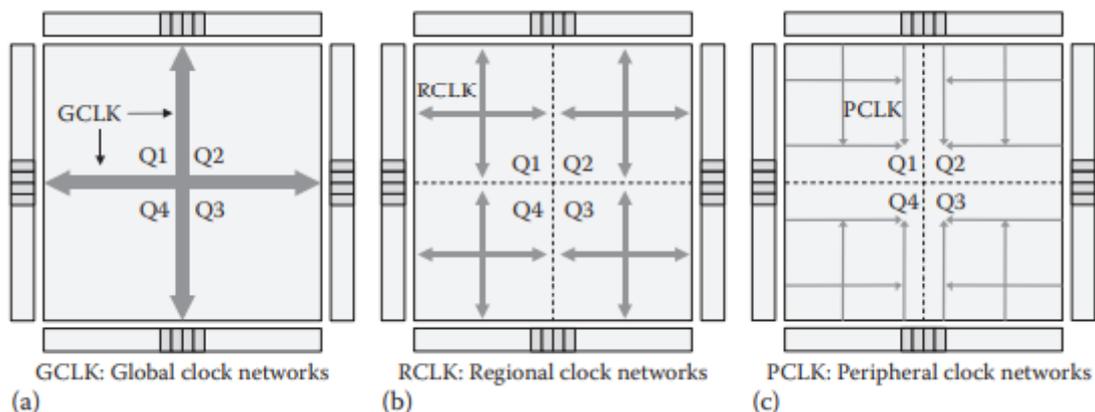


Fig. 3.4. Rete di clock a) globale, b) regionale, c) periferica [4].

Per ottimizzare le prestazioni, i CMB sono fisicamente posizionati vicino agli IOB e sono collegati ad essi con risorse dedicate. Nella maggior parte delle FPGA, i CMB vengono utilizzati anche per generare i segnali di clock utilizzati dai blocchi SerDes e dai ricetrasmittitori [4].

3.3.2) Blocchi di memoria

La maggior parte dei sistemi digitali richiede risorse per archiviare quantità significative di dati. Le memorie sono gli elementi principali incaricati di questo compito. I tempi di accesso alla memoria sono solitamente molto più lunghi dei ritardi di propagazione nei circuiti logici e, per questo motivo, i fornitori di FPGA hanno sempre prestato particolare attenzione all'ottimizzazione delle risorse logiche in modo da poter supportare, nel modo più efficiente possibile, l'implementazione delle memorie interne [4].

Poiché la logica combinatoria, le LUT e i flip-flop sono disponibili negli LB, le memorie interne possono essere costruite combinando le risorse di diversi LB ed ottenendo la cosiddetta memoria distribuita. Tuttavia, affinché la memoria distribuita sia più efficiente, agli LB possono essere fornite risorse aggiuntive rispetto a quelle destinate a supportare l'implementazione di funzioni logiche di uso generale, come ingressi e uscite di dati aggiuntivi, segnali di enable e segnali di clock. Questo implica che gli LB debbano essere più complessi. Man mano che le architetture FPGA si sono evolute per supportare l'implementazione di sistemi digitali sempre più complessi, le esigenze di memoria sono aumentate. Di conseguenza, i fornitori hanno deciso di inserire nei propri dispositivi dei blocchi di memoria dedicati, che utilizzano linee di interconnessione specifiche per ottimizzare i tempi di accesso. Queste sono particolarmente adatte per l'implementazione di memorie "profonde" (con un gran numero di posizioni) mentre la memoria distribuita è più adatta per memorie "ampie" (con molti bit per posizione) con poche posizioni, shift register o linee di ritardo. Nelle FPGA attuali, sia la memoria distribuita che i blocchi di memoria dedicati supportano configurazioni e modalità operative simili. La memoria dedicata è strutturata in elementi costitutivi di base di capacità fissa, che possono essere combinati per ottenere memorie più profonde (collegamento in serie) o più ampie (collegamento in parallelo). Le possibili combinazioni dipendono dal tipo di memoria di destinazione e dalla modalità di funzionamento. Le configurazioni più comuni sono:

- RAM single-port, in cui è possibile eseguire una sola operazione di lettura o scrittura alla volta ogni ciclo di clock;
- Simple RAM dual-port, in cui è possibile eseguire contemporaneamente un'operazione di lettura e una di scrittura;
- True RAM dual-port, dove è possibile eseguire due operazioni di scrittura, due operazioni di lettura o una di lettura e una di scrittura simultaneamente con frequenze diverse se necessario;
- ROM, in cui è possibile eseguire un'operazione di lettura in ogni ciclo di clock;
- Shift register;
- FIFO, sincrona, utilizzando un clock per entrambe le operazioni di lettura e scrittura o asincrona, utilizzando due clock indipendenti per le operazioni di lettura e scrittura [4].

Blocchi di memoria dedicati basati su celle SRAM possono essere trovati in tutte le FPGA attuali. In alcuni dispositivi sono disponibili anche memorie flash con capacità di lettura/scrittura. Il loro

grande vantaggio deriva dal fatto di essere non volatili e il loro principale svantaggio è che richiedono più segnali di controllo rispetto a quelli richiesti nelle SRAM, rendendo, quindi, più complesso il loro controllo da parte dell'FPGA. L'aggiunta di memorie nelle FPGA è facilitata da strumenti di progettazione software, che partizionano automaticamente i blocchi di memoria definiti dal progettista, assegnandoli ai blocchi di memoria disponibili nel dispositivo di destinazione, in base alle modalità operative specificate e ai vincoli di progettazione relativi all'area e velocità. Il contenuto della memoria può essere inizializzato anche con l'ausilio degli strumenti di progettazione, dove i valori da memorizzare inizialmente nelle memorie sono descritti con una sintassi predefinita, che consente di includere nel bitstream di configurazione il contenuto dei file di testo [4].

3.4) Vantaggi delle FPGA

Le FPGA sono riconfigurabili tramite software, rendendole molto flessibili. A differenza di un microprocessore o di un ASIC (Application Specific Integrated Circuit), è possibile modificare la funzionalità di una FPGA. Poiché l'esecuzione dell'FPGA è controllata dalle interconnessioni programmabili, gli utenti possono implementare attività realmente parallele all'interno dell'FPGA stessa, che verranno eseguite simultaneamente e indipendentemente l'una dall'altra [3]. I vantaggi dati da questa tecnologia sono i seguenti:

- **Flessibilità**: la tecnologia FPGA offre la flessibilità di testare un'idea o un concetto e verificarlo nell'hardware senza passare attraverso il lungo processo di fabbricazione dell'ASIC personalizzato. Poiché sono riconfigurabili, l'utilizzo di chip FPGA consente di tenere il passo con le esigenze future dell'applicazione in questione. Man mano che un prodotto o un sistema matura, è possibile apportare miglioramenti funzionali senza perdere tempo a riprogettare l'hardware o modificare il layout della scheda.
- **Prestazione**: sfruttando il parallelismo hardware, le FPGA superano la potenza di calcolo dei processori di segnali digitali, rompendo il paradigma dell'esecuzione sequenziale. Il controllo degli ingressi e delle uscite a livello hardware fornisce tempi di risposta più rapidi e funzionalità specializzate per soddisfare perfettamente i requisiti dell'applicazione.
- **Affidabilità**: le FPGA, non utilizzando sistemi operativi, riducono al minimo i problemi di affidabilità con una vera esecuzione parallela e hardware dedicato a ogni singola attività. Al contrario, i sistemi basati su processori spesso implicano diversi livelli di astrazione per aiutare a pianificare le attività e condividere le risorse tra più processi. I sistemi basati su processore possono continuamente essere soggetti al rischio che le attività critiche in termini di tempo si impediscano a vicenda.
- **Costo**: la spesa per la progettazione ASIC supera di gran lunga quella delle soluzioni hardware basate su FPGA. Il grande investimento iniziale per gli ASIC è facile da giustificare per gli OEM (Original Equipment Manufacturer) che spediscono migliaia di chip all'anno; molti utenti, però, necessitano di funzionalità hardware personalizzate per decine o centinaia di sistemi in fase di sviluppo. Poiché i requisiti di sistema spesso cambiano nel tempo, il costo per apportare modifiche incrementali ai progetti FPGA è trascurabile rispetto alla grande spesa di rigenerazione di un ASIC [3].

Dose Delivery System

La Fondazione CNAO, l'INFN e l'Università di Torino hanno progettato, costruito e messo in funzione un sistema di somministrazione della dose, il Dose Delivery System (DDS). Questo sistema serve per monitorare e guidare i fasci ionici, accelerati da un sincrotrone dedicato, e per distribuire la dose con una tecnica di scansione attiva 3D.

La tecnica di scansione attiva sfrutta le proprietà fisiche delle particelle cariche: un fascio di dimensioni millimetriche (pencil beam), proveniente direttamente dalla linea del fascio, viene condotto verso il paziente per ottenere piccole deposizioni di dose. Il tumore viene irradiato dalla sovrapposizione di una sequenza di fasci, chiamati spot, ciascuno dei quali rilascia un numero definito di particelle in una posizione definita con un fascio monoenergetico. Il volume target è segmentato in più strati, detti layer isoenergetici, ortogonali alla direzione del fascio. La tecnica di somministrazione della dose risultante è una sorta di scansione 3D del volume target. L'estensione sul piano trasversale è ottenuta per mezzo di due dipoli di scansione indipendenti per le deflessioni del fascio orizzontale e verticale, posti all'estremità della linea di estrazione e diversi metri a monte del paziente. Modificando l'energia del fascio ed eseguendo l'irradiazione layer per layer, un tumore di forma arbitraria può essere irradiato dalla sua parte più prossimale a quella più distale o viceversa [10]. Questa tecnica di irradiazione, possibile solo con particelle cariche, consente di raggiungere una conformazione di dose molto elevata nel tumore, riducendo al minimo il carico di radiazioni sui tessuti sani. Un'attenta pianificazione del trattamento assegna l'energia, il numero di particelle e la posizione trasversale ad ogni spot al fine di fornire la distribuzione ottimale della dose al paziente. L'immagine in Fig. 4.1 illustra una rappresentazione schematica della tecnica di scansione attiva.

Per le linee di fascio del CNAO, queste operazioni vengono gestite dal DDS. I componenti principali di questo sistema sono due rivelatori di monitoraggio del fascio indipendenti, denominati BOX1 e BOX2, che saranno descritti nel corso del capitolo, interfacciati con due sistemi di controllo che svolgono le funzioni di controllo rapido e controllo lento in tempo reale, presentati anch'essi successivamente, e collegati ai magneti di scansione e a un chopper. Il DDS è uno dei

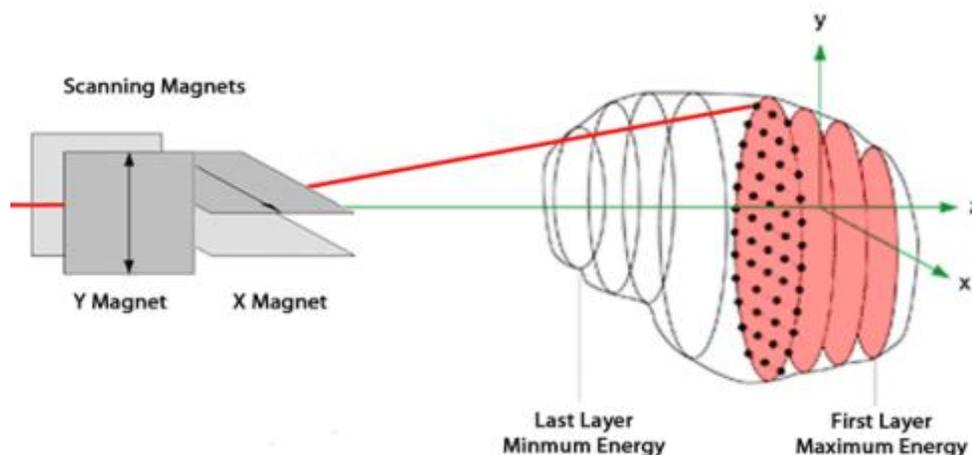


Fig. 4.1. Tecnica di scansione attiva.

pochi sistemi esistenti in tutto il mondo che lavorano con un fascio ionico per la radioterapia a scansione modulata [9]. Di seguito, verrà descritta in dettaglio la parte di rivelazione del fascio del DDS, ovvero la BOX1 e BOX2, seguita, infine, dal sistema di controllo basato su FPGA.

4.1) Rivelatori di fascio

Il metodo di irradiazione utilizzato presso il CNAO, dove, tramite una tecnica di scansione attiva, un pencil beam irradia il tumore con un gran numero di spot, richiede un sistema di monitoraggio del fascio preciso, veloce, stabile e uniforme, che consenta la misurazione online del numero di particelle erogate negli spot e della posizione degli spot nel campo di irradiazione. Vengono presentati i rivelatori di monitoraggio del fascio, sviluppati dal CNAO e dal gruppo dell'Istituto Nazionale di Fisica Nucleare e dell'Università di Torino, che sono attualmente installati nelle quattro linee di trattamento del CNAO [8].

I rivelatori sviluppati per il CNAO si basano su camere di ionizzazione a piastre parallele con un unico grande elettrodo (o anodo) o elettrodi segmentati in strisce (strip) o pixel. L'anodo raccoglie la carica di ionizzazione, dovuta al passaggio delle particelle, proporzionale all'energia depositata nel gas e, infine, alla dose erogata lungo i tre assi. Per esigenze di sicurezza e ridondanza, sono stati progettati due rivelatori indipendenti per ogni linea di trattamento: sono alloggiati in box separate, BOX1 e BOX2, e vengono letti ed elaborati in modo indipendente durante il trattamento. BOX1 contiene una camera integrale (INT1) con un anodo con area grande, seguita da due camere dove l'anodo è segmentato in strip (STPX e STPY). BOX2 contiene un'altra camera integrale (INT2), seguita da una camera in cui l'anodo è segmentato in pixel (PXC) [8]. Un disegno schematico dell'assemblaggio dei rivelatori è mostrato in Fig.4.2. S1 e S2 sono gli anodi segmentati in strip, P è l'anodo segmentato in pixel, C3 è l'anodo delle camere integrali, C2 e C4 sono i catodi. Le cornici D1, D2 e D3 sono necessarie per fornire lo spazio desiderato tra gli elettrodi mentre C1 e C5 vengono aggiunti per fornire un gap aggiuntivo per il gas su entrambi i lati. Tutti i catodi sono costituiti da una lamina di mylar di 12 μm di spessore con uno strato di alluminio di poche decine di nm su un lato, ad eccezione del catodo comune delle due camere strip che è due volte più spesso e alluminato su entrambi i lati [8]. Gli anodi delle camere integrali e strip sono costituiti da un foglio di kapton di 25 μm di spessore ricoperto da uno strato di alluminio di 17 μm inciso con tecnica PCB standard per ottenere 128 strip conduttive larghe 1,55 mm con un passo di 1,65 mm. L'anodo pixel è costituito da uno strato di kapton spesso 50 μm , su cui è stato depositato su entrambi i lati un film di rame spesso 20 μm , distribuito in una matrice di 32 x 32 pixel su un lato e delle piste, una per ogni

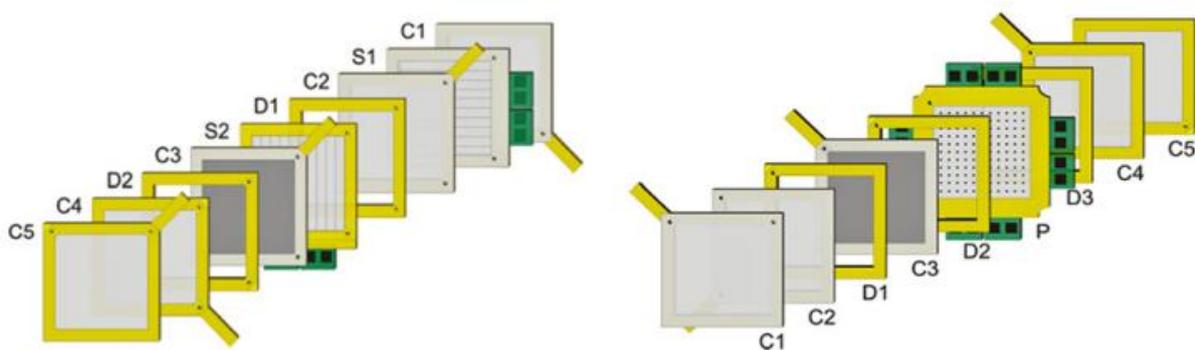


Fig.4.2. Disegno schematico del rivelatore BOX1 (sinistra) e BOX2 (destra) [8].

pixel, sull'altro lato, essendo il collegamento realizzato attraverso un foro conduttivo. L'area di ogni pixel conduttivo è $6,5 \times 6,5 \text{ mm}^2$ e il passo è $6,6 \text{ mm}$ [8]. Questi parametri spaziali elencati sono mostrati in Fig.4.3. Per evitare accumuli di carica e correnti di dispersione, un anello di protezione è posizionato intorno alle aree sensibili degli anodi e impostato al potenziale anodico. Gli elettrodi sono distesi e incollati su cornici in fibra di vetro che forniscono anche supporto meccanico per le schede di lettura. Ulteriori cornici vengono aggiunte tra gli elettrodi per fornire un gap di gas di 5 mm per ciascuna camera. Per riempire la camera di gas è necessaria una sovrappressione all'interno del pacchetto di cornici. La combinazione di una vasta area e di lamine sottili può comportare un cambiamento considerevole del gap di gas. Per evitare questo effetto, vengono aggiunte due camere vuote, una su ciascun lato delle camere attive, dove il gas scorre alla stessa pressione. Le camere vuote sono state realizzate con una cornice che sostiene una lamina di mylar alluminata di $12 \mu\text{m}$ di spessore. Per ridurre il rumore elettrico indotto, queste lamine sono messe al potenziale di terra. Un raccordo per il gas su ciascuna camera vuota fornisce l'ingresso e l'uscita del gas. Infine, il percorso del gas attraverso la camera attiva è stato vincolato tramite scanalature e fori lavorati sulle cornici. Le cornici sono impilate una sull'altra e sono allineate tramite perni meccanici di alta precisione [8].

L'involucro esterno della scatola è sigillato per far funzionare le camere con i gas; nella parte anteriore e in quella posteriore vengono utilizzate due finestre costituite da una sottile lamina di mylar alluminato per garantire la tenuta dei gas, minimizzando il materiale lungo il percorso del fascio. La dimensione di ogni box è $[50 \times 70 \times 10] \text{ cm}^3$. L'area sensibile totale è $[240 \times 240] \text{ mm}^2$ per le camere integrali e $[211 \times 211] \text{ mm}^2$ per le camere strip e pixel. Al fine di massimizzare l'efficienza di raccolta, le camere sono azionate con azoto (N_2) e con un campo elettrico di circa 800 V/cm , generato polarizzando i catodi a -400 V e gli anodi alla tensione di riferimento dell'ingresso dell'elettronica di front-end (2.05 V). Si è osservato che in queste condizioni operative l'efficienza

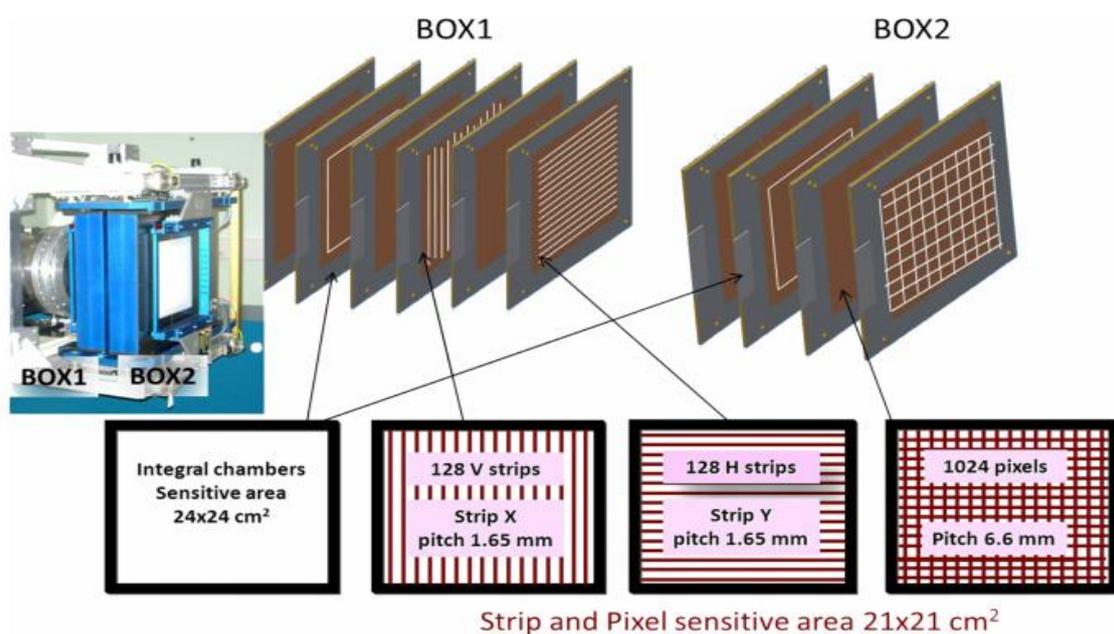


Fig. 4.3. Elettrodi dei rivelatori in dettaglio (BOX1 e BOX2) [9].

di raccolta è circa del 100% [8]. In ogni rivelatore, i dati dalle camere integrali e dalle camere strip e pixel vengono acquisiti indipendentemente. Al CNAO le camere integrali leggono ad una frequenza di 1 MHz mentre la frequenza di lettura delle camere strip e pixel è, rispettivamente, di 20 e 10 kHz. Il DDS, quindi, misura a una frequenza fissa il numero di particelle erogate, la posizione trasversale del fascio e la sua dimensione, mediante queste cinque camere di ionizzazione a piastre parallele riempite di azoto. Nella BOX1, INT1 esegue la misura del flusso del fascio. La STPX, che ha orientazione verticale, e la STPY orientata orizzontalmente, forniscono la misura della posizione e dell'ampiezza del fascio lungo le due direzioni ortogonali. Nella BOX2 vi è INT2 che misura la fluenza del fascio, mentre PXC misura posizione e larghezza. Il materiale interposto dai rivelatori del fascio è distribuito approssimativamente in 20 cm e costituisce uno dei principali contributi alla dispersione laterale del fascio. Per minimizzare l'effetto, le box vengono installate vicine al paziente a circa 70 cm dall'isocentro [9]. L'isocentro è il punto nello spazio che viene identificato come il punto in cui viene centrato il volume tumorale durante un trattamento. La carica raccolta nelle camere di ionizzazione dipende dalla pressione e dalla temperatura del gas, nonché dall'alta tensione che c'è nello spazio dove si trova il gas, ossia il gap. Queste grandezze sono monitorate da un set di trasduttori installati in ciascuna box e controllati da un controller di interfaccia periferica (PIC). I valori vengono periodicamente letti e verificati dal PIC rispetto ai valori preimpostati. Adeguate procedure di interlock vengono attivate ogni volta che uno qualsiasi dei valori è al di fuori dell'intervallo previsto.

4.2) Elettronica delle camere di ionizzazione

L'elettronica di lettura dei rivelatori si basa su due ASIC ad ampio range dinamico a 64 canali: il TERA06 per le camere strip e pixel e il TERA07 per le camere integrali. Il circuito è costruito su un chip VLSI (Very Large Scale Integration), che indica un'elevata integrazione di transistor all'interno di un singolo chip, e si basa sull'idea di contare il numero di volte in cui un condensatore viene caricato dalla corrente di ingresso e scaricato dal circuito [5].

4.2.1) Tera06

L'ASIC, denominato Tera06, è stato progettato con tecnologia CMOS da 0,8 μm , incapsulato in un package PGA 144, ed è stato sviluppato per leggere in modo indipendente la corrente da un massimo di 64 pixel o strip e convertirla in frequenza. Questi ASIC equipaggiano le intere piastre parallele dei rivelatori di ionizzazione strip e pixel, che sono utilizzati al CNAO, per la misura della posizione e dell'intensità dei fasci terapeutici [7].

Viene fornita l'alta tensione alla piastra catodica che genera il campo elettrico, E , tra le piastre catodica e anodica. Dall'altro lato della camera, ciascuna strip o pixel anodico è direttamente accoppiato a uno degli ingressi del canale. Le coppie di ioni create nel gas dal fascio carico che attraversa il volume del gas sono costrette dal campo elettrico a spostarsi verso gli elettrodi. Questo risulterà in una corrente con un'intensità proporzionale alla frazione di flusso del fascio che attraversa l'area anodica [5]. Il chip permette di leggere in parallelo la corrente fino a 64 canali e misurare la carica corrispondente convertendola in conteggi. La conversione si basa sul principio dell'integratore di carica. Per avere un ampio range dinamico, l'architettura ASIC si basa su un convertitore corrente-frequenza, seguito da un contatore digitale. Utilizzando questa architettura, la

corrente può essere misurata contando il numero di impulsi che escono dal convertitore in un certo tempo [7].

La Fig. 4.4 mostra il convertitore corrente-frequenza. La corrente di ingresso è integrata su un condensatore (C_{int}) da 600 fF tramite un amplificatore operazionale di transconduttanza (OTA). L'uscita integrata (V_A) è una rampa di tensione che viene confrontata con una tensione di soglia (V_{th}) da un comparatore sincrono. Quando $V_A > V_{th}$ il comparatore attiva il generatore di impulsi che manda un impulso ad un circuito di sottrazione. Una quantità fissa di carica, detta charge quantum (Q_c), viene sottratta dal condensatore C_{int} , traducendosi in una forte diminuzione della tensione attraverso C_{int} , che è proporzionale alla carica iniettata dal circuito di sottrazione. Il circuito di sottrazione funziona come segue: il charge quantum si ottiene grazie all' impulso di tensione inviato dal generatore di impulsi al condensatore da 200 fF (chiamato condensatore di sottrazione, C_{sub}) incluso nel blocco del circuito di sottrazione (Fig. 4.5). Nel fronte di salita e di discesa di questo impulso, due cariche uguali di segno opposto vengono iniettate. Nel fronte di salita, che inietterebbe una corrente con lo stesso segno della corrente in ingresso, si va a cortocircuitare il circuito di sottrazione a massa tramite l'interruttore I_2 . D'altra parte, all'arrivo del fronte di discesa, l'interruttore I_1 è chiuso, I_2 è aperto e la carica viene opportunamente sottratta. La linea di ritardo è stata regolata per fornire la corretta sincronizzazione. In questo modo, la carica e la scarica di C_{int} sono operazioni analogiche che vengono eseguite in parallelo senza tempi morti [5]. Dopo questa sottrazione di carica, l'uscita dell'OTA (V_A) aumenta di nuovo e il processo viene ripetuto [7]. Per ogni sottrazione di carica viene inviato anche un impulso al contatore digitale. Pertanto, la frequenza dell'impulso è proporzionale alla corrente di ingresso, secondo la seguente relazione: $f = I_{in}/Q_c$, dove I_{in} è la corrente di ingresso. Dunque, il numero di impulsi generati nel tempo di misurazione moltiplicato per il charge quantum fornisce il valore della carica totale letta all'ingresso. Il charge quantum è definito dalla relazione $Q_c = C_{sub}V_{sub}$, dove V_{sub} è definita come la

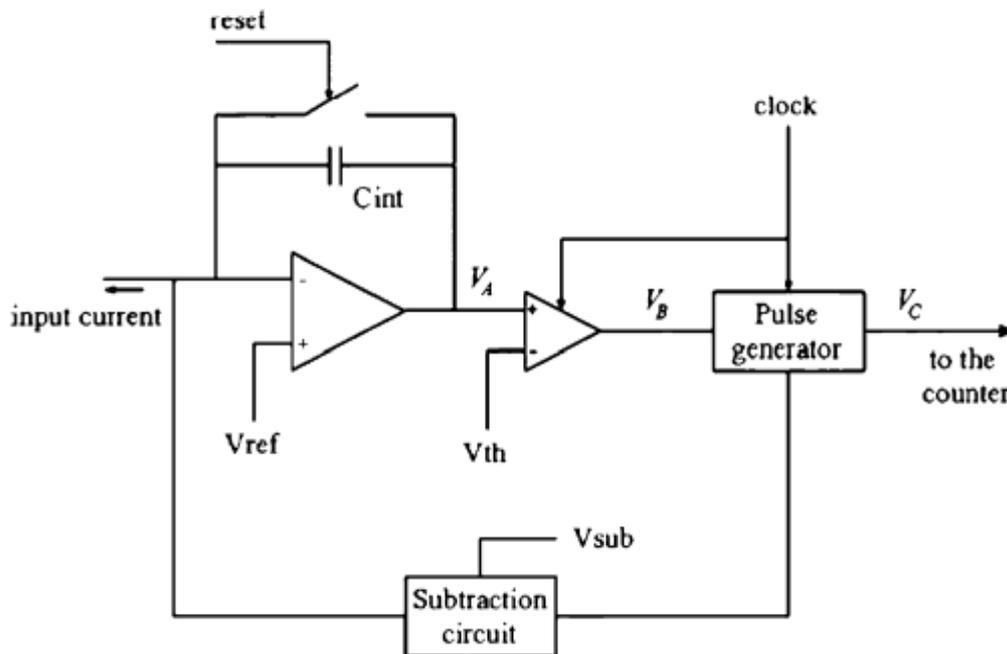


Fig. 4.4. Schema del convertitore corrente-frequenza [7].

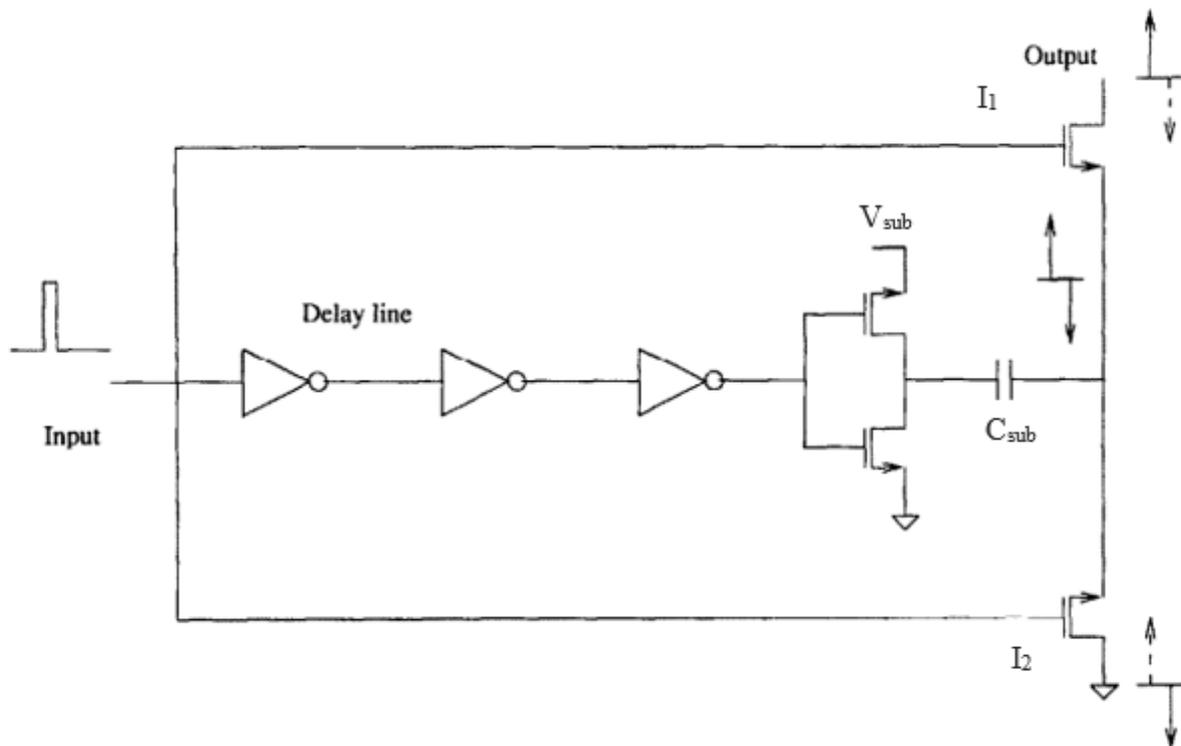


Fig. 4.5. Schema del circuito di sottrazione [5].

differenza tra due tensioni di riferimento ($V_{\text{pulse}+}$ e $V_{\text{pulse}-}$), che sono impostate esternamente in un intervallo compreso tra 0,5 e 4,5 V. Una variazione di tensione V_{sub} ai capi del condensatore di sottrazione C_{sub} corrisponde a una variazione di tensione di $\Delta V = \frac{V_{\text{sub}} C_{\text{sub}}}{C_{\text{int}}}$ ai capi del condensatore di integrazione C_{int} [7].

La variazione di tensione ΔV , applicata all'ingresso dell'OTA, deve essere in grado di ridurre l'ampiezza della sua tensione d'uscita, resettando lo stato del comparatore. Il charge quantum minimo è fissato dalla risoluzione del comparatore ed è limitato a 100 fC in questo design. Lo schema generale dell'ASIC è mostrato in Fig.4.6. I seguenti segnali possono essere impostati esternamente e sono comuni a tutti i canali:

- tensione di riferimento dell'OTA (V_{ref});
- soglie del comparatore V_{th} , $V_{\text{pulse}+}$ e $V_{\text{pulse}-}$.

I contatori possono essere azzerati tramite un ingresso digitale asincrono comune, reset. La lettura dei contatori può essere effettuata indipendentemente rispetto a qualsiasi altra operazione. Usando il latch, la configurazione dei bit effettiva di ciascun contatore a 16 bit viene memorizzata in un registro a 16 bit. Questa operazione viene eseguita in parallelo per tutti i 64 canali. Qualsiasi canale può essere acquisito indirizzandolo tramite sei ingressi digitali, che sono le linee di selezione del canale. Il registro appropriato viene indirizzato al segnale del bus di uscita a 16 bit tramite il Multiplexer (MUX). Vale la pena ricordare quanto segue: l'operazione di latch viene eseguita contemporaneamente per tutti i canali, in modo che i conteggi vengano copiati nei registri quando le transizioni di conteggio sono completate; sebbene l'acquisizione debba essere effettuata canale per canale, i contatori vengono bloccati contemporaneamente con un latch nei registri (dunque i dati si

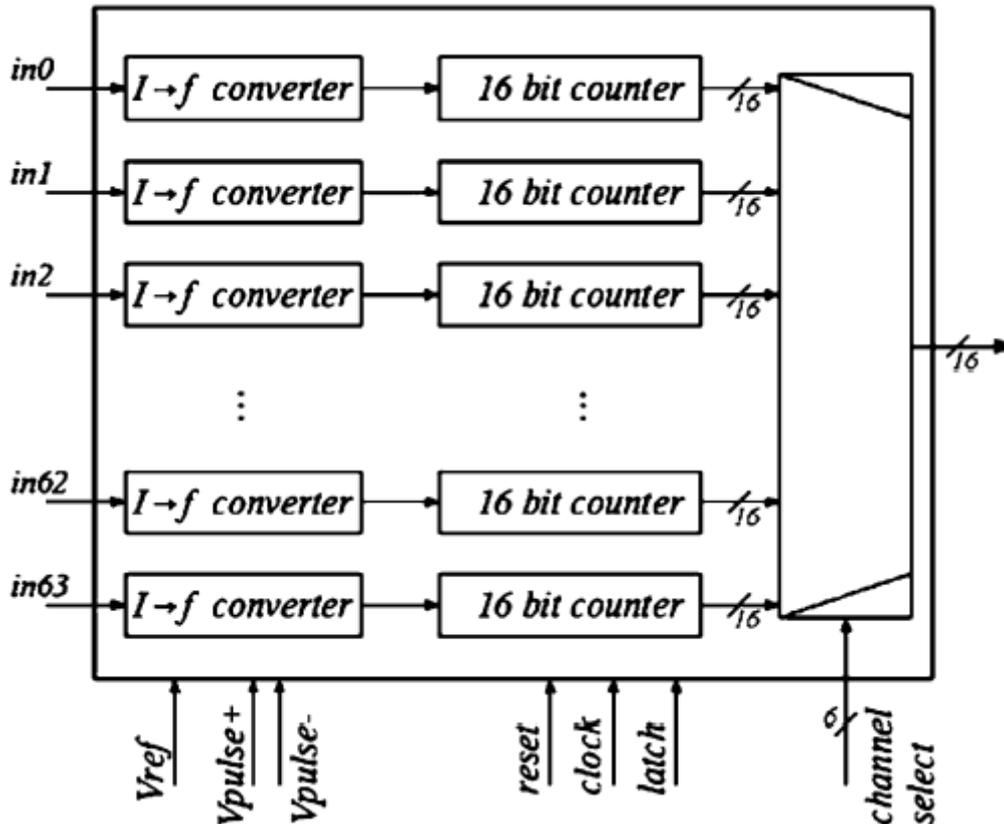


Fig. 4.6. Architettura del chip Tera06 [7].

riferiscono allo stesso istante). Essendo l'acquisizione indipendente dal conteggio degli impulsi, l'acquisizione dei dati non interrompe l'attività del contatore, quindi non c'è tempo morto dovuto alla lettura dei dati. Il chip funziona con una frequenza di clock di 20 MHz e la frequenza massima dei contatori è un quarto della frequenza di clock, ossia 5 MHz [7].

4.2.2) Tera07

Il chip Tera06 ha alcune limitazioni come contatori a 16 bit, corrente di ingresso unipolare, 5 MHz come frequenza di conteggio massima ed è progettato con tecnologia CMOS da 0,8 μm . La versione successiva, denominata TERA07, migliora le prestazioni. Essendo stato progettato con tecnologia CMOS 0,35 μm , può accettare ingressi di entrambe le polarità e i contatori sono stati estesi a 32 bit, con la capacità di contare sia avanti che indietro, ossia con un ordine di conteggio crescente oppure decrescente. Il chip funziona con una frequenza di clock di 100 MHz e una frequenza di conversione massima di 20 MHz, ossia un quinto della frequenza di clock. Il limite di velocità di conteggio più alto fornisce un contributo sostanziale all'incremento del range dinamico del circuito, che è di circa 10^4 sia per le correnti di ingresso positive che per quelle negative. La conversione dell'ingresso da unipolare a bipolare ha il vantaggio di poter ottenere una migliore precisione nella misura del regime di bassa corrente (pA). Infatti, con la presente versione, la corrente di dispersione, il cui valore assoluto e la cui polarità variano da canale a canale, può essere misurata con precisione e sottratta dall'ingresso [6].

Lo schema di un canale è mostrato in Fig.4.7. Il funzionamento è sempre lo stesso visto finora; vi è una differenza dovuta al fatto che in questa architettura si usino contatori sincroni digitali up/down.

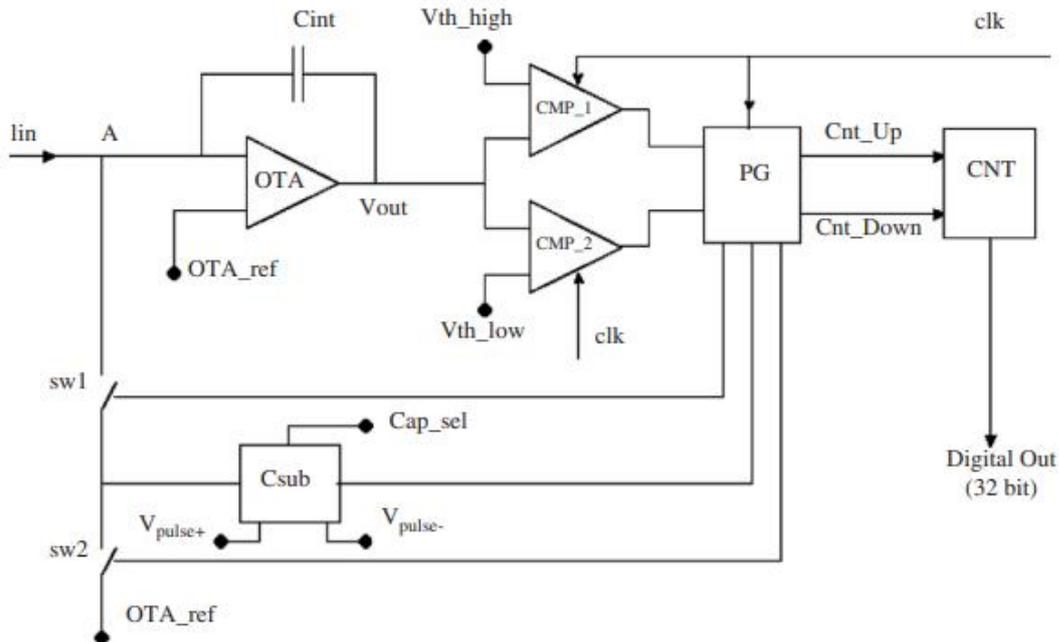


Fig. 4.7. Schema di un canale del Tera07 [6].

La corrente di ingresso è integrata su un condensatore C_{int} da 600 fF tramite un OTA. La tensione di uscita dell'integratore aumenta quando la corrente di ingresso è negativa (cioè esce dal chip) e diminuisce nel caso inverso. Questa tensione viene confrontata con due soglie fissate da due comparatori sincroni, CMP_1 e CMP_2. La tensione di soglia dei due comparatori è fissata esternamente: V_{th_high} è la tensione di soglia del comparatore CMP_1 funzionante per correnti di ingresso negative, mentre V_{th_low} si riferisce a CMP_2, attivo per correnti di ingresso positive [6]. Ogni volta che la tensione di ingresso del comparatore attraversa la soglia, il relativo comparatore imposta un livello all'ingresso del generatore di impulsi (PG). Facendo riferimento alla Fig.4.7, PG invia al condensatore C_{sub} un impulso positivo di 20 ns. L'ampiezza dell'impulso di tensione ΔV_{pulse} è definita come la differenza tra due tensioni di riferimento, V_{pulse+} e V_{pulse-} , che sono impostate esternamente nell'intervallo tra 0,5 e 3,3 V. Tre condensatori paralleli, rispettivamente di 50, 100 e 200 fF, possono essere aggiunti indipendentemente per ottenere la capacità totale C_{sub} , per la quale è possibile selezionare qualsiasi valore compreso tra 50 e 350 fF a passi di 50 fF. Le risposte in uscita di C_{sub} a un impulso di tensione sono due segnali di corrente di segno opposto, δ_+ e δ_- con una carica associata Q_+ (Q_-) = ΔV_{pulse} ($-\Delta V_{pulse}$) $\times C_{sub}$ [6]. La temporizzazione del primo segnale, δ_+ , è determinata dal fronte di salita dell'impulso di tensione, mentre δ_- corrisponde al fronte di discesa. Uno schema delle forme d'onda è mostrato in Fig.4.8. Agendo sulla temporizzazione dei comandi sugli interruttori sw1 e sw2, Q_+ o Q_- può essere indirizzato o all'ingresso dell'OTA o all'OTA_ref, aggiungendo o sottraendo così una quantità fissa di carica ad ogni impulso generato da PG. La decisione su quale segnale di carica deve essere indirizzato all'OTA dipende dall'uscita dei comparatori, in modo da rimuovere una quantità fissa di carica da C_{int} . Questo si traduce in una variazione della tensione su C_{int} , fornita da $\frac{Q}{C_{int}}$. Se, dopo questa azione, la tensione del comparatore di ingresso rimane sopra (sotto) la soglia, PG continuerà a emettere impulsi e si fermerà quando la tensione ritorna sotto (sopra) la soglia. In parallelo, PG invia un impulso da contare al contatore. PG, a seconda del comparatore che agisce al suo ingresso, può decidere di contare in avanti o

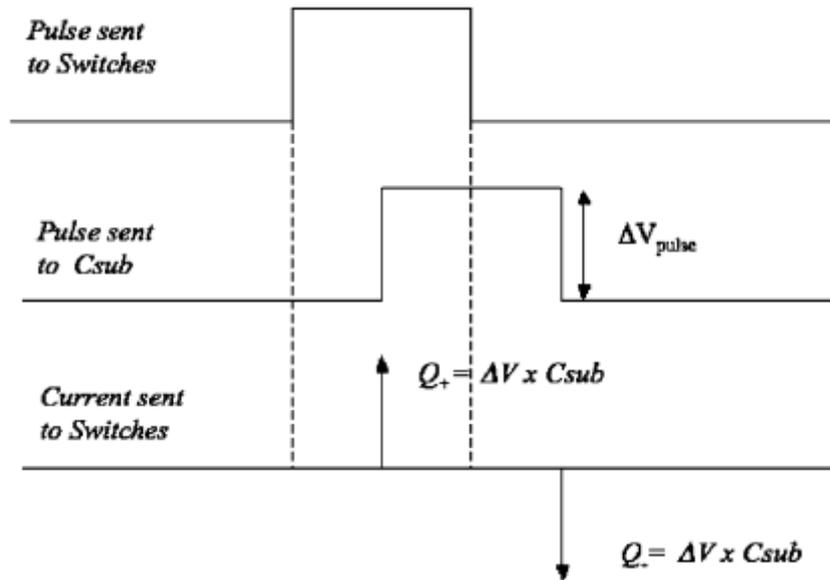


Fig.4.8. Forme d'onda relative al processo di sottrazione di carica [6].

indietro (Cnt_down o Cnt_up nella Fig.4.7). Tutte le operazioni sono sincronizzate con un master clock esterno e supervisionate tramite una macchina digitale a stati finiti (FSM) implementata nel blocco PG. Ad un dato impulso di clock, ogni volta che PG rileva CMP_1 (o CMP_2) attivo, i successivi tre cicli di clock saranno presi dal processo di sottrazione di carica e dagli impulsi di controllo dell'interruttore. Sono necessari due cicli di clock aggiuntivi prima di riattivare l'FSM per dare tempo all'OTA di diminuire la tensione di uscita [6]. Pertanto, per generare un conteggio, sono necessari cinque impulsi di clock master. La relazione tra la frequenza di uscita, f , e la corrente di ingresso, I_{in} , è: $f = \frac{I_{in}}{Q_C}$, dove Q_C , che è il charge quantum, è dato da: $Q_C = C_{sub}\Delta V_{pulse}$. La carica totale letta dal rivelatore è data dal numero di impulsi, generati durante il tempo di misurazione, moltiplicato per il valore del charge quantum Q_C . Il charge quantum può essere impostato nell'intervallo tra 25 fC, dato da $C_{sub} = 50$ fC e $\Delta V_{pulse} = 0,5$ V, e 1,155 pC, che si ottiene con $C_{sub} = 350$ fC e $\Delta V_{pulse} = 3,3$ V [6].

Lo schema generale del chip TERA07 è mostrato in Fig. 4.9. I seguenti segnali possono essere impostati esternamente e sono comuni a tutti i canali:

- tensione di riferimento OTA;
- soglie di confronto: V_{th_high} e V_{th_low} ;
- V_{pulse+} e V_{pulse-} ;
- C_{sub} , valore del condensatore di scarica, selezionabile tramite linee digitali.

Il condensatore di integrazione di tutti i canali può essere scaricato tramite un ingresso digitale comune, resetA. Allo stesso modo, tutti i contatori possono essere azzerati tramite un ingresso digitale asincrono comune, resetD. La lettura dei contatori può essere effettuata indipendentemente rispetto a qualsiasi altra operazione. Utilizzando il comando di latch, la configurazione momentanea dei bit di ciascun contatore viene memorizzata in un registro a 32 bit. Questa operazione viene eseguita in parallelo per tutti i 64 canali. Qualsiasi canale può essere acquisito indirizzandolo tramite i sei ingressi digitali (linee di selezione del canale). Il registro appropriato

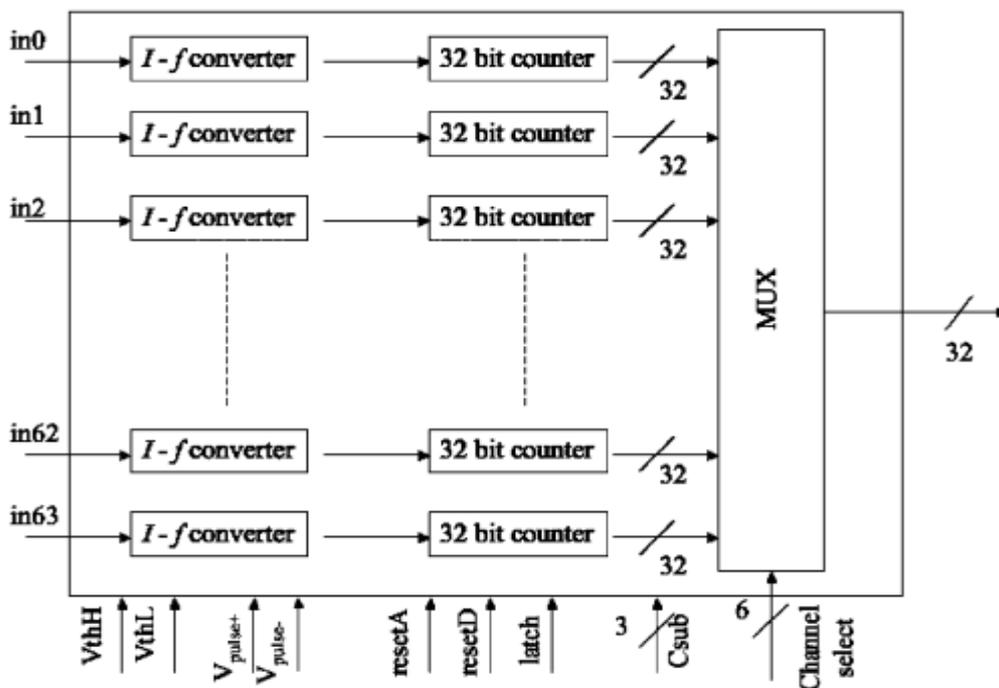


Fig. 4.9. Architettura del chip Tera07 [6].

viene indirizzato al segnale del bus di uscita a 32 bit tramite il MUX. Così come per il Tera06, l'operazione di latch viene eseguita contemporaneamente per tutti i canali (quindi i dati si riferiscono allo stesso tempo). Essendo l'acquisizione indipendente dal conteggio degli impulsi, l'acquisizione dei dati non arresta l'attività del contatore, quindi non c'è tempo morto dovuto alla lettura dei dati [6].

Le caratteristiche principali delle due versioni dei due chip sono riassunte nella Tabella 4.1.

	<u>TERA06</u>	<u>TERA07</u>
TECNOLOGIA	CMOS 0,8 μm	CMOS 0,35 μm
DIMENSIONE	6 x 7 mm ²	5,4 x 4,5 mm ²
NUMERO DI CANALI	64	64
CORRENTE D'INGRESSO	Unipolare	Bipolare
RANGE DEL CHARGE QUANTUM	100 fC – 800 fC	25 fC – 1.155 pC
MASSIMA FREQUENZA DI CONTEGGIO	5 Mhz	20 Mhz
MASSIMO INGRESSO DI CORRENTE	1 μA	4 μA
NUMERO DI BIT DEI CONTATORI	16	32

Tabella 4.1. Caratteristiche principali dei due chip TERA.

4.3) Sistema di controllo

Il ruolo del DDS consiste nell'erogare la dose secondo le prescrizioni, ossia irradiare ogni spot con proprietà definite: numero di particelle, energia selezionata e posizione specificata. Le proprietà dello spot sopra menzionate sono fornite come file DICOM RT dalla versione Syngo Siemens VC10 del Treatment Planning System (TPS), che supporta l'erogazione sia di protoni che di ioni carbonio con modalità di scansione attiva [9]. Il flusso di dati pre-trattamento relativo al paziente, a partire dal TPS attraverso il controllo del sincrotrone e del DDS è mostrato in Fig.4.10.

Lo strumento software Dicom Operating Powering (DOP) aggiunge informazioni, memorizzate in tag DICOM privati, al piano TPS per definire completamente le impostazioni dell'acceleratore e ottimizzare la consegna del fascio. Tra questi, si cita l'ottimizzazione del percorso di consegna del fascio, le istruzioni per disattivare l'irradiazione, quando è necessario coprire una grande distanza tra due spot consecutivi, e la selezione dell'intensità del fascio. Il piano di trattamento esteso è memorizzato nel Sistema Informativo Oncologico (OIS) che svolge il ruolo di organizzazione delle varie sessioni di trattamento. Pochi minuti prima dell'inizio del trattamento, la Console di Trattamento (TC) riceve il piano paziente dall'OIS e invia al controllo dell'acceleratore un elenco di codici binari (cycle-code) che definiscono l'energia, la dimensione del fascio, il flusso e il tipo di particella per ogni ciclo dell'acceleratore. Parallelamente, viene prodotto e scaricato il file "DDS_treatment", contenente l'elenco completo degli spot raggruppati in layer. La sequenza degli spot definita nel file viene utilizzata dal DDS durante l'erogazione del fascio per guidarlo durante l'irradiazione [9].

L'elettronica per il controllo dell'intero sistema DDS (inclusi gli alimentatori del rivelatore, la distribuzione del gas, l'interfaccia con il sistema di interlock e il sistema di recupero della dose erogata) è ospitata nell'armadio mostrato in Fig.4.11. Per evitare danni da radiazioni, l'armadio è posizionato a circa 2 m a lato della linea di fascio, in una stanza situata appena prima della sala di trattamento. L'architettura del sistema di controllo si basa su due cestelli (crate) PXI (PCI eXtensions for Instrumentation) e schede commerciali National Instruments (NI).

I due crate sono: il controllo rapido (Fast Control, FC), dedicato alle operazioni critiche in termini di tempo eseguite durante l'irradiazione e il controllo lento (Slow Control, SC) per verificare che la linea sia impostata nella maniera corretta [9].

La Fig.4.12 riassume tutte le interfacce utilizzate dall'FC e dall'SC del DDS, insieme al tipo di dati scambiati.

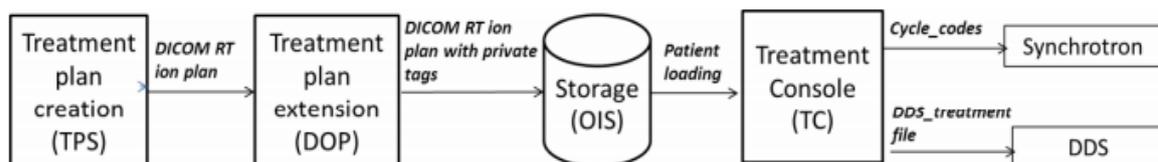


Fig.4.10. Flusso di dati pretrattamento [9].

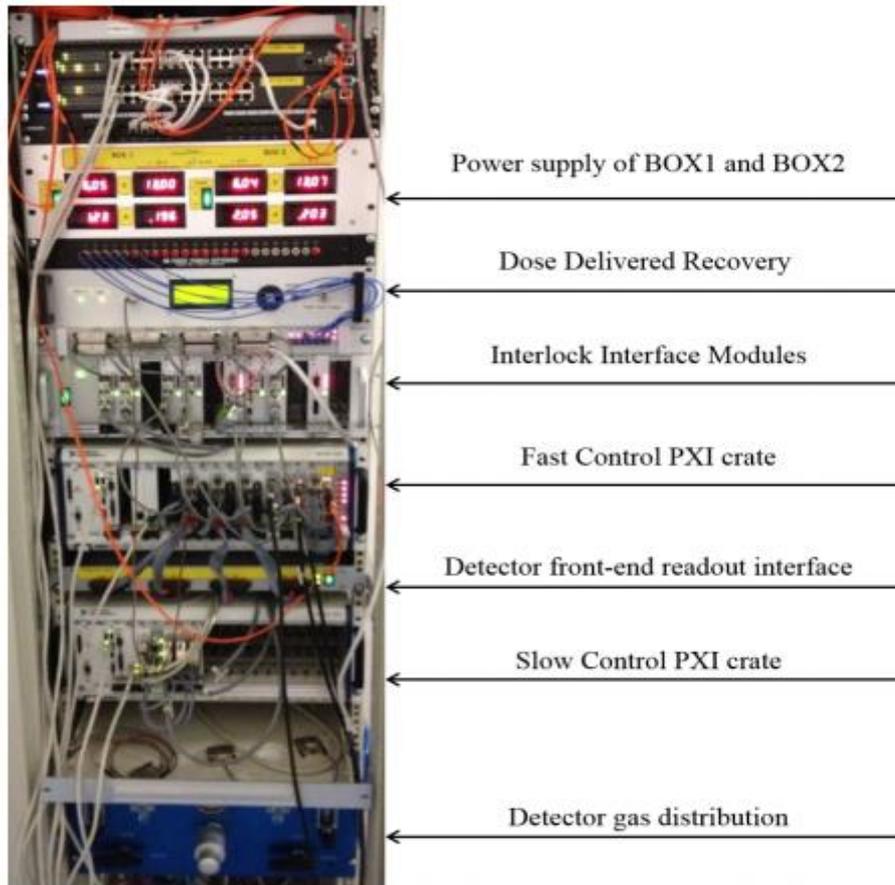


Fig. 4.11. Armadio ospitante il DDS [9].

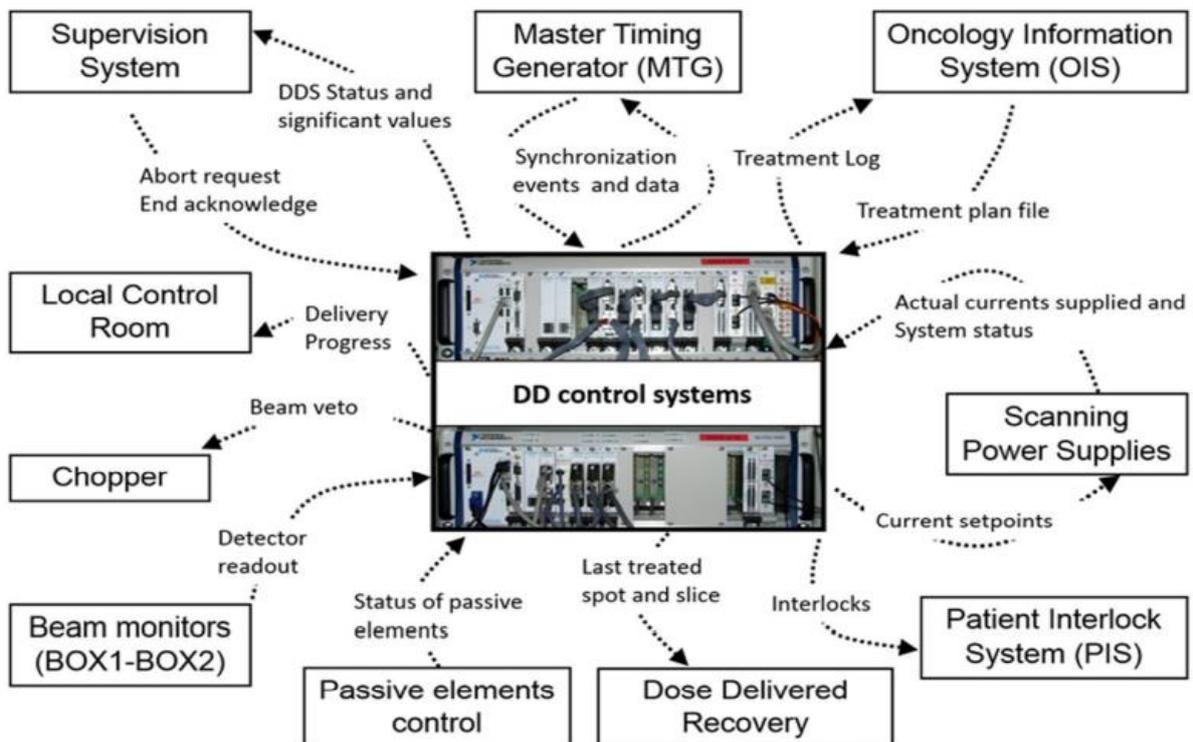


Fig.4.12. Sistemi e dispositivi esterni che comunicano con FC e SC. I dati scambiati sono descritti con una freccia che indica la direzione del flusso [9].

4.3.1) Fast Control

Il crate del Fast Control include una CPU che lavora con un sistema operativo in tempo reale (RTOS), cinque moduli NI PXI con FPGA a bordo (Int FPGA, Strip FPGA, Pixel FPGA, Scan FPGA e Timing FPGA), quattro schede di memoria, una scheda di rete aggiuntiva e tre moduli PXI costruiti espressamente per il DDS. L'intelligenza principale e la sicurezza del DDS sono affidate a cinque FPGA, che si trovano in quattro schede NI 7813R e una NI 7811R, le quali lavorano in modalità stand-alone per garantire la massima velocità e affidabilità durante la fase di estrazione del fascio. Durante il trattamento di uno spot, il ruolo principale è svolto dall'Int FPGA, che guida l'erogazione, accedendo sequenzialmente alle proprietà dello spot prescritte, memorizzate nelle schede di memoria, e misurando il numero di particelle erogate attraverso le camere di ionizzazione integrali (INT1 e INT2) a una frequenza di campionamento di 1 Msample/s (ossia un milione di campioni al secondo). Parallelamente, la Strip FPGA e la Pixel FPGA misurano la posizione e le dimensioni del fascio nel piano trasversale attraverso i rivelatori STPX, STPY e PXC a, rispettivamente, una frequenza di 20 e 10 kHz. Al raggiungimento di un numero minimo di conteggi predefinito viene applicato un algoritmo (centro di gravità) per ricostruire la posizione del fascio e confrontare il valore misurato con quello pianificato. La differenza calcolata dalla Strip FPGA viene utilizzata dalla Scan FPGA per fornire un feedback ai magneti di scansione, che correggono piccole deviazioni di posizione. Quando le deviazioni di posizione sono maggiori di una soglia, attualmente impostata a 2 mm, la Strip FPGA e la Pixel FPGA generano un interlock mettendo in pausa l'irradiazione. Se il numero di conteggi è inferiore alla soglia minima sopra descritta, non viene applicato alcun feedback e viene restituita la posizione della strip con il maggior numero di conteggi [9]. La Scan FPGA viene utilizzata per controllare i magneti di scansione, inviando la corrente richiesta agli alimentatori e rileggendo le correnti misurate a una frequenza di 40 kHz. Inoltre, la Scan FPGA ha anche il compito di spegnere il fascio quando la distanza tra i centri degli spot è maggiore di una soglia, attualmente 20 mm, agendo sul chopper. Infine, la Timing FPGA interfaccia l'FC con il master timing generator (MTG) per sincronizzare le operazioni dell'FC con le operazioni dell'acceleratore (preparazione del magnete, accelerazione del fascio, estrazione del fascio, ecc.). L'MTG trasmette cycle-code ed eventi del sincrotrone a diversi sistemi e apparecchiature (tra cui il DDS) che necessitano di segnali di trigger relativi a operazioni dell'acceleratore. La maggior parte degli altri sistemi legge e utilizza eventi di temporizzazione, mentre l'FC del DDS genera e invia eventi allo stesso MTG. In particolare, l'FC, alla fine di ogni estrazione del fascio, invia gli eventi 'Repeat_cycle' o 'Next_cycle' per chiedere un nuovo ciclo; questo potrebbe essere una ripetizione del ciclo precedente nel caso in cui un layer non sia stato completato. Inoltre, l'FC, tramite la Timing FPGA, convalida l'inizio del ciclo inviando all'MTG l'evento "Ack_cycle" per dare conferma che il cycle-code è quello atteso dal DDS. L'MTG, quando riceve questo evento, genera il ciclo; al contrario, il trattamento viene abortito. L'FC, tramite una scheda seriale collegata alle schede interne della BOX1 e BOX2, si occupa anche di leggere i parametri del gas che hanno un impatto sulla sensibilità delle camere. La misurazione della temperatura e della pressione del gas all'interno dei rivelatori e il controllo dell'alta tensione applicata alle camere sono importanti per misurazioni accurate del flusso del fascio.

4.3.2) Slow Control

Il crate SC include una CPU real-time, tre schede input-output digitali, una scheda di rete aggiuntiva e un modulo PXI per fornire segnali al sistema di interlock. Per trattamenti specifici possono essere necessari range shifter e modulatori, i quali sono definiti come elementi passivi. Questi elementi vengono montati manualmente. L'SC verifica la loro corretta installazione prima che il piano di trattamento venga erogato e, poi, a una frequenza di circa 1 Hz, esegue la verifica della configurazione durante il trattamento generando un segnale di interlock in caso di errore. Esegue, quindi, un controllo iniziale e, successivamente, un controllo in tempo reale.

4.4) Funzioni e operazioni del DDS

La sequenza di un trattamento può essere suddivisa nelle quattro fasi indicate nel diagramma di flusso della Fig.4.13 [9].

4.4.1) Off – trattamento

Per mantenere la sicurezza del paziente, un gran numero di componenti del sistema viene continuamente controllato prima dell'erogazione del trattamento. La gestione degli interlock, la pubblicazione dei dati e la gestione dei rivelatori sono sempre attive per emettere un avviso, se viene rilevata una condizione anomala. Ad esempio, la Scan FPGA invia un set preconfigurato di correnti agli alimentatori dei magneti di scansione e riceve le correnti misurate e lo stato degli alimentatori. Inoltre, Int FPGA, Strip FPGA e Pixel FPGA leggono continuamente i rivelatori per monitorare la corrente di fondo dell'elettronica di lettura.

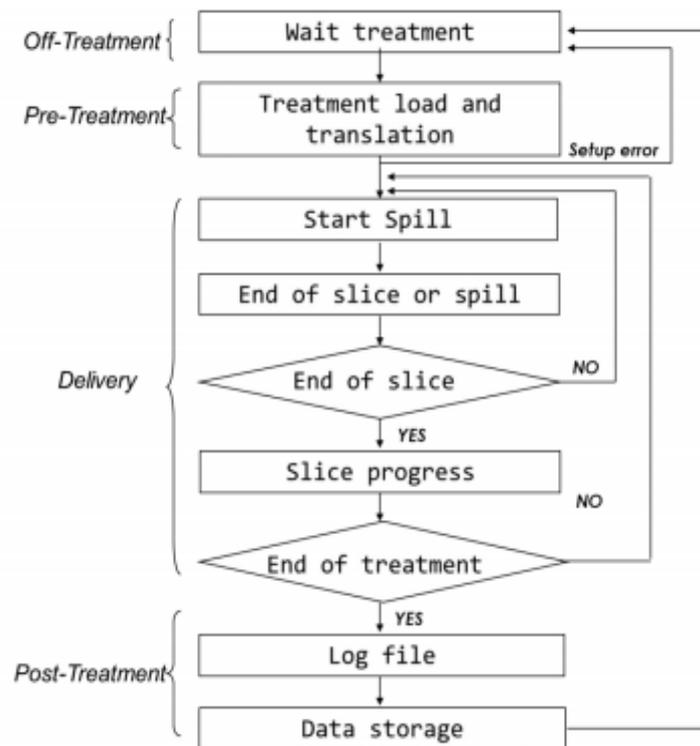


Fig.4.13. Diagramma di flusso delle operazioni del DDS [9].

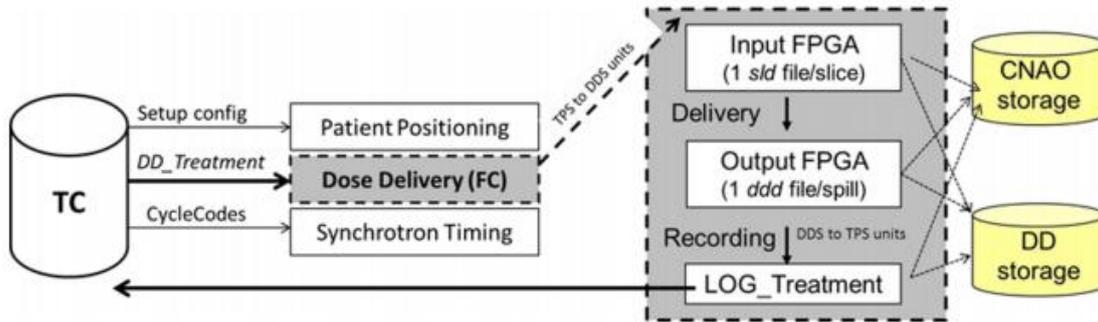


Fig. 4.14. Flusso di dati durante un trattamento [9].

4.4.2) Pre – trattamento

Pochi minuti prima del trattamento, durante la procedura di posizionamento del paziente, l'FC riceve il file "DD_Treatment" tramite la TC, come mostrato in Fig.4.14. Per ogni spot, l'FC esegue la conversione delle proprietà definite dal TPS nelle quantità corrispondenti utilizzate internamente dal DDS. In dettaglio, il numero di particelle da erogare a ciascuno spot viene convertito in numero di conteggi letti dalle camere integrali, le coordinate dello spot all'isocentro vengono convertite nella posizione del fascio in unità di canale nei rivelatori di posizione. Inoltre, vengono calcolate le correnti dei magneti necessarie per raggiungere la posizione di ciascuno spot tenendo conto dell'energia del fascio.

Gli spot da trattare con la stessa energia sono raggruppati in file binari (Input FPGA in Fig.4.14, denominato slice_data, sld file), uno per ogni layer. Successivamente, durante il trattamento, i dati memorizzati nei file sld verranno caricati sequenzialmente nelle schede di memoria, ciclo dopo ciclo, per essere consultati sequenzialmente dalle FPGA dell'FC. Vengono, inoltre, controllate le correnti di fondo del rivelatore, l'alta tensione, la temperatura e la pressione del gas, il setup degli elementi passivi e i registri di interlock; se non viene riscontrato alcun problema, il file DD si porta in uno stato di 'ready', che sarà riconosciuto dal Sistema di Supervisione (SS) come via libera per erogare il fascio.

4.4.3) Erogazione del fascio

Ogni ciclo inizia dopo che l'FC ha riconosciuto il cycle code. L'MTG avvia le operazioni per indirizzare il fascio richiesto nella sala di trattamento corretta. L'erogazione comincia seguendo la sequenza degli spot definita nei file sld. Non appena una delle due camere integrali raggiunge il numero prescritto di conteggi per un dato spot, un trigger dall'Int FPGA viene inviato alla Scan FPGA che imposta le nuove correnti dei magneti di scansione. Parallelamente, lo stesso trigger viene ricevuto dalle FPGA Strip e Pixel per caricare i dati di riferimento dello spot successivo utilizzati per l'interlock e il feedback di posizionamento. Una volta che l'ultimo spot di un layer è stato trattato (questo potrebbe richiedere più di un ciclo), la Scan FPGA interrompe immediatamente l'erogazione del fascio agendo sul chopper del fascio. Al termine di ogni estrazione, l'FC, tramite la Timing FPGA, invia una richiesta di un nuovo ciclo che può coincidere o meno con quello precedente, a seconda che il layer sia stato concluso o meno. In quest'ultimo caso, l'FC ricarica nelle memorie i dati del layer corrente, dove sono stati rimossi gli spot già erogati durante il ciclo precedente. Come mostrato in Fig.4.14, i dati di output FPGA, contenenti le

misurazioni online del DDS, sono temporaneamente memorizzati nell'FC come file binario specifico per il ciclo chiamato `dose_delivery_data` (ddd). La disponibilità di questi file, che contengono la lettura completa del monitor, è obbligatoria per la creazione del file Log.

4.4.4) Post – trattamento

Quando l'ultimo spot dell'ultimo layer viene irradiato, è creato un file Log che viene inviato all'OIS. L'OIS applica ulteriori controlli rispetto al trattamento pianificato, al fine di garantire la fine dell'erogazione della dose. Il file Log viene creato dall'FC utilizzando i dati memorizzati nei file ddd e applicando delle conversioni per fornire le seguenti quantità: numero di layer erogati, numero di spot erogati, numero di particelle erogate e posizione degli spot erogati. Infine, il DDS termina il trattamento salvando i file sld e ddd in un archivio remoto per l'analisi dei dati offline. Tale analisi produce un feedback importante sulle prestazioni del sistema.

4.4.5) Gestione degli interlock

La sicurezza del trattamento si basa principalmente su due sistemi di interlock: sistema di interlock paziente (PIS) e sistema di interlock di sicurezza. Questi sistemi rilevano eventuali condizioni anomale o di errore e forzano l'immediata interruzione dell'erogazione del fascio o inibiscono le operazioni fintanto che le condizioni persistono. Il PIS è dedicato alla sicurezza del paziente agendo sul chopper per interrompere il trattamento quando si verifica un interlock. L'erogazione può così essere interrotta per un breve periodo, solitamente pochi secondi, consentendo il ripristino della normale esecuzione del trattamento, oppure può essere interrotta quando persiste la condizione di guasto che ha provocato l'interlock, anche dopo il ripristino effettuato dall'operatore. Nell'ultimo caso, il trattamento viene interrotto attraverso una procedura di recupero sicuro, in cui la dose non erogata viene calcolata dall'OIS in base al numero di particelle e spot memorizzati nel file DD Log e utilizzata per fornire un nuovo campo di irradiazione da erogare appena possibile per completare la frazione di trattamento. Le condizioni di errore del DDS vengono raccolte e inviate al PIS tramite segnali a livello hardware in modo da garantire una rapida reazione. Diversi interlock vengono generati dalle FPGA dell'FC e SC quando alcune quantità sono al di fuori della finestra di tolleranza, come ad esempio l'intensità del fascio, il numero di conteggi, l'alta tensione o la temperatura e pressione del gas.

Un dispositivo a batteria tampone, chiamato sistema di recupero della dose erogata, riceve, memorizza e visualizza continuamente le informazioni relative all'ultimo layer e spot trattato. Può funzionare per almeno 20 minuti senza alimentazione esterna e fornisce le informazioni importanti per poter recuperare la parte mancante di un trattamento in caso di grave guasto del sistema durante il trattamento stesso.

4.4.6) Pubblicazione dei dati

Ciascuna sala trattamenti è dotata di una propria sala controllo locale, dove vengono gestite tutte le operazioni a distanza relative all'esecuzione del trattamento. Un programma dedicato riceve dall'FC e visualizza i dati misurati, lo stato del sistema e l'avanzamento del trattamento. La posizione del fascio, il numero di conteggi del monitor e la percentuale di spot e layer erogati vengono aggiornati continuamente nel display del monitor del DDS. In caso ci sia un interlock, il display lo visualizzerà

permettendo di individuare rapidamente il problema che arresta il fascio. Anche lo stato del DDS durante le fasi off, pre e post trattamento viene monitorato dall'SS e visualizzato nella sala di controllo principale.

Rivelatore per monitor a bassa intensità

Il progetto descritto in questa tesi è frutto della collaborazione e del lavoro in parallelo della Fondazione CNAO e dell'INFN (sezione di Roma). In particolare, l'INFN si è occupato di progettare e costruire il rivelatore in grado di rilevare singole particelle, attraverso l'utilizzo di fibre scintillanti accoppiate con fotomoltiplicatori al silicio (SiPM); il CNAO, invece, ha lavorato sul sistema di acquisizione dati, ossia sulla parte di elettronica che si va ad interfacciare con il DDS, in modo da generare un protocollo di trasmissione dati che possa essere compreso da quest'ultimo. Il monitor che si intende costruire farà parte del DDS, così come i monitor già esistenti presso il CNAO, descritti esaurientemente nel capitolo precedente, con la differenza che il primo è progettato per "vedere" intensità di fasci molto più bassi. Nel seguente capitolo viene esposto brevemente il funzionamento del rivelatore e della tecnologia alla sua base, ossia delle fibre scintillanti.

5.1) Fibre scintillanti

Le fibre scintillanti rappresentano uno dei possibili modi per tracciare e monitorare fasci di particelle cariche di diverse intensità, grazie, soprattutto, ai tempi di risposta molto rapidi e all'elevata flessibilità. Questo paragrafo è dedicato alla descrizione del meccanismo di scintillazione nei materiali organici e, in particolare, delle principali proprietà delle fibre scintillanti. Verrà analizzata la struttura interna delle fibre, il trasporto della luce attraverso la fibra stessa ed infine la necessità di accoppiamento con un fotomoltiplicatore per amplificare il segnale in uscita.

5.1.1) Gli scintillatori

Uno scintillatore è un materiale capace di emettere impulsi di luce, in genere visibile o ultravioletta, quando viene attraversato da fotoni di alta energia o da particelle cariche. La particella incidente cede parte della propria energia allo scintillatore, causando l'eccitazione di un elettrone che si sposta in un livello ad energia superiore; la successiva diseccitazione causa l'emissione di un fotone di energia relativamente bassa, tipicamente nel visibile. Questo fenomeno si chiama luminescenza, ossia l'emissione di radiazioni luminose da parte di un corpo conseguente a transizioni di elettroni da un livello energetico superiore ad un altro ad energia più bassa. Questa può essere provocata da diversi processi: nel caso degli scintillatori, avviene per azione di fotoni e, quindi, si parla di fluorescenza, concetto già esposto nel Cap. 1. Tale impulso di luce viene poi rivelato ed amplificato da opportuni sensori, tipicamente i fotomoltiplicatori. Esistono diverse tipologie di scintillatori che si distinguono principalmente per il tipo di materiale di cui sono composti, i tempi di risposta, le lunghezze d'onda emesse, l'efficienza di scintillazione (ossia quanta energia viene convertita in luce). Negli scintillatori organici la luminescenza è dovuta a singole molecole, tipicamente poste in una soluzione dove anche il solvente è costituito da una sostanza organica, in forma solida o liquida. Nel caso solido, come appunto nelle fibre scintillanti, si tratta solitamente di un polimero (polistirene, polivinil toluene o PMMA) o di un cristallo. La molecola responsabile della scintillazione è chiamata fluoro primario. Tipicamente, la lunghezza di attenuazione (ossia la lunghezza dopo la quale un fascio di particelle incidente su un materiale risulta attenuato di un

fattore $\frac{1}{e}$) della luce emessa da tale molecola è molto piccola e la radiazione viene riassorbita dallo stesso fluoro primario dopo pochi centimetri. Per questo motivo spesso viene inserita una seconda molecola scintillante, detta fluoro secondario: questo secondo elemento ha il compito di assorbire la luce emessa dal fluoro primario e di riemetterla ad una lunghezza d'onda maggiore in modo tale da garantirne la trasmissione fino all'estremità dello scintillatore. Solitamente il rapporto tra le concentrazioni dei due elementi è di 1:100.

Le proprietà degli scintillatori organici possono essere così riassunte:

- l'emissione della radiazione luminosa avviene in tempi molto brevi, minori di 10 ns;
- la risposta luminosa è inferiore rispetto agli scintillatori inorganici e sono, quindi, necessari amplificatori performanti per l'acquisizione della luce emessa;
- sono molto flessibili; grazie a questa proprietà sono realizzabili configurazioni geometriche non possibili con le rigide strutture cristalline degli scintillatori inorganici.

5.1.2) Fibre scintillanti

Le fibre scintillanti sono sostanzialmente fibre ottiche plastiche alle quali sono state aggiunte molecole di sostanze scintillanti. Le fibre sono composte da un nucleo interno trasparente, detto core, rivestito da un materiale, anch'esso trasparente, detto cladding. Nelle fibre scintillanti più comuni, il core è costituito da polistirene, drogato con molecole organiche, mentre il cladding è realizzato con PMMA. A volte è presente anche un ulteriore rivestimento esterno, chiamato Extra Mural Absorber (EMA) che funge da guaina protettiva e limita gli effetti di cross-talk tra fibre contigue. Alcuni tipi di fibre, infine, sono dotati di un ulteriore strato di cladding con indice di rifrazione inferiore rispetto al primo cladding, la cui funzione è quella di incrementare l'efficienza di trasporto della luce nella fibra stessa. I due materiali che costituiscono core e cladding sono coassiali e hanno indice di rifrazione diverso, in modo tale che la propagazione della luce all'interno del core avvenga tramite riflessione totale, che si ha se l'angolo di emissione della luce è inferiore rispetto al cosiddetto angolo critico θ_C . Lo schema della struttura di una fibra scintillante è mostrato in Fig. 5.1, mentre in Fig. 5.2 è mostrato il principio di riflessione su cui si basa il trasporto del segnale luminoso all'interno delle fibre, che adesso sarà illustrato.

Considerando un'emissione isotropa di luce, l'angolo critico θ_C , per cui si ha riflessione totale, si

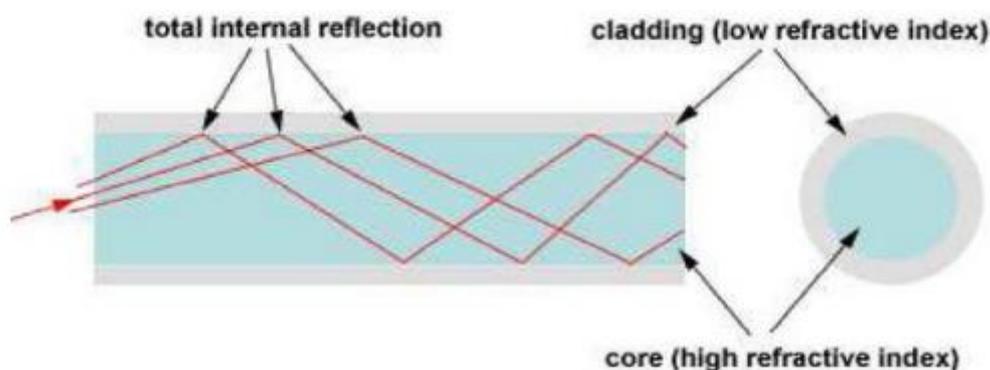


Fig. 5.1. Struttura interna della fibra scintillante.

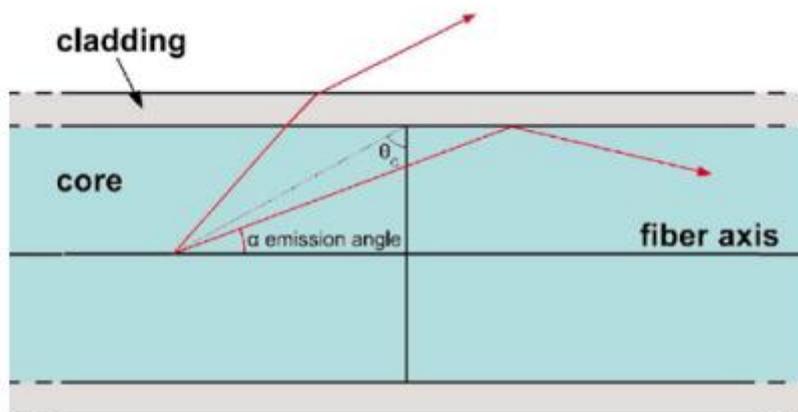


Fig. 5.2. Riflessione del segnale all'interno della fibra.

ricava dalla legge di Snell e vale:

$$\theta_C = \arcsin \left(\frac{n_{clad}}{n_{core}} \right) \quad (5.1)$$

dove n_{clad} e n_{core} sono, rispettivamente, l'indice di rifrazione del materiale che costituisce il cladding e l'indice di rifrazione del materiale che costituisce il core. Il principio di riflessione su cui si basa il trasporto del segnale luminoso all'interno delle fibre è il seguente: l'impulso luminoso emesso lungo l'asse della fibra si propaga all'interno del core, se l'angolo di emissione α è minore dell'angolo critico θ_C ; la frazione di luce che verrà incanalata e trasmessa verso le estremità della fibra dipende, quindi, dall'angolo α di emissione.

5.2) Fotomoltiplicatori

Il numero di fotoni che arrivano all'estremità della fibra è molto piccolo ed è necessario un sistema di amplificazione in grado di generare un segnale abbastanza grande da poter essere letto. I dispositivi solitamente utilizzati per questo scopo sono i fotomoltiplicatori. Un tubo fotomoltiplicatore è un rivelatore elettronico di luce estremamente sensibile nell'ultravioletto, nel range del visibile e nel vicino infrarosso. Lo schema di un fototubo è mostrato in Fig.5.3: è costituito da un tubo di vetro, in cui è applicato il vuoto, contenente un fotocatodo, diversi elettrodi (detti dinodi) e un anodo. Il funzionamento del fotomoltiplicatore è basato sull'interazione per effetto fotoelettrico della luce di scintillazione con il materiale del fotocatodo e sulla successiva emissione secondaria di elettroni da parte dei dinodi. La superficie del fotocatodo è ricoperta di uno strato di materiale che favorisce l'effetto fotoelettrico; l'emissione di elettroni, detti fotoelettroni, in

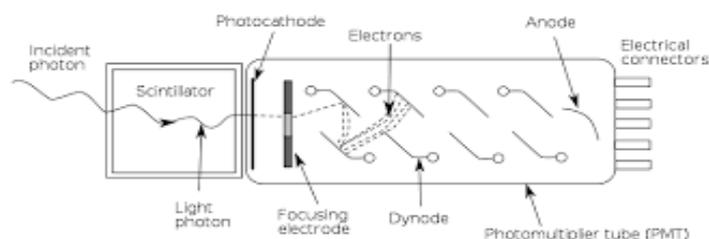


Fig 5.3. Fotomoltiplicatore.

seguito all'interazione con il catodo, dipende dall'efficienza di quest'ultimo. Gli elettroni vengono quindi focalizzati da un apposito elettrodo verso lo stadio di moltiplicazione, costituito da una serie di dinodi, ciascuno connesso ad un potenziale superiore al precedente. Il primo elettrone emesso dal catodo subisce un'accelerazione a causa del campo elettrico tra il catodo e il primo dinodo ed acquista energia cinetica sufficiente per raggiungere il primo dinodo, dove viene provocata l'emissione secondaria di diversi elettroni di minore energia. Tali elettroni vengono a loro volta accelerati dalla differenza di potenziale, maggiore, tra il primo e il secondo dinodo; il processo si ripete per ciascuno dei dinodi presenti nel fototubo. Si ha, così, un fenomeno a cascata per cui un singolo fotone che colpisce il catodo provoca la produzione di moltissimi elettroni. Al termine della sequenza di dinodi, gli elettroni vengono raccolti da un anodo generando un impulso elettrico, che indica la rivelazione del fotone.

In generale ogni fotomoltiplicatore può essere distinto da un altro per le diverse caratteristiche che possiede, quali:

- L'efficienza quantica, definita come il rapporto tra il numero di elettroni che giungono all'anodo e il numero di fotoni incidenti al catodo. Dipende fortemente dalla frequenza della radiazione incidente e dalla struttura del materiale;
- La sensibilità radiante, definita come il rapporto tra l'intensità della corrente catodica e la potenza luminosa incidente;
- L'efficienza del sistema di raccolta, definita come il rapporto tra gli elettroni che arrivano sul primo dinodo e gli elettroni emessi dal fotocatodo;
- Il guadagno del fotomoltiplicatore, definito come il rapporto tra la corrente anodica e la corrente catodica che dipende molto dal numero di dinodi presenti nel sistema di moltiplicazione.

5.2.1) SiPM

I SiPM (Silicon Photo-Multiplier) sono i fotosensori che sono stati scelti per il monitor per basse intensità. A differenza dei fotomoltiplicatori tradizionali, costruiti con tubi a vuoto, i SiPM sono prodotti direttamente da un wafer di silicio su cui vengono impiantate matrici costituite da array di microcelle. Ciascuna microcella è un fotodiodo a valanga (Avalanche PhotoDiode, o APD) a singolo fotone. Un APD è un fotodiodo semiconduttore altamente sensibile che sfrutta l'effetto fotoelettrico per convertire la luce in un segnale elettrico di corrente. Attualmente i moduli basati su un singolo APD in silicio raggiungono velocità di conteggio fino a 10-20 MHz. La limitazione della velocità di conteggio degli APD è dovuta a un tempo morto tipico di circa 50 ns o più. È possibile, tuttavia, ridurre i tempi morti combinando diversi APD in parallelo. L'idea è che considerando ad esempio un array di 100 APD, se un APD ha rilevato un fotone, 99 APD sono ancora attivi per rilevare il fotone successivo. È chiaro che, se i fotoni sono distribuiti sull'intero array di APD, il tempo morto del modulo di rilevamento è inferiore a quello di un singolo APD [11]. Tipicamente, i SiPM sono costituiti da 100-1000 fotodiodi collegati in parallelo montati su un chip comune. Una rappresentazione dello schema elettrico è mostrata in Fig. 5.4. I singoli APD sono collegati in parallelo, quindi i rilevamenti simultanei di singoli APD si sovrappongono per formare un segnale di uscita proporzionale al numero di rilevamenti simultanei. Se c'è un rilevamento in uno dei fotodiodi, vi sarà un impulso elettrico all'uscita del SiPM. Ma, se si verificano due o più rilevamenti contemporanei, l'ampiezza dell'impulso elettrico in uscita sarà uguale al numero di fotoni rilevati

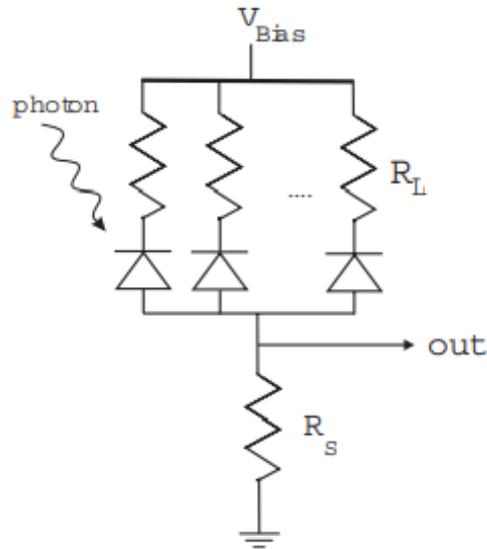


Fig.5.4. Schema elettrico di SiPM collegati in parallelo [11].

moltiplicato per l'ampiezza di un impulso relativo ad un singolo rilevamento. Quindi, l'altezza dell'impulso di uscita è proporzionale al numero di fotoni rilevati [11].

Il manuale dei SiPM utilizzati nel progetto del monitor per basse intensità è riportato in [12]. Si tratta di un Multi Pixel Photon Counter (MPPC), altro termine per indicare una matrice di SiPM.

In Fig. 5.5 è mostrata la matrice di fotomoltiplicatori in questione (serie S13361-3050). Quando una di quelle celle rileva un fotone, l'output generato potrebbe contenere degli altri impulsi spuri, dovuti a fenomeni detti afterpulse e crosstalk, che sono separati dagli impulsi di uscita relativi ai fotoni incidenti. Gli afterpulse sono dei picchi di segnale più piccoli che si osservano dopo l'impulso di output legato alla rilevazione di un fotone. Il crosstalk, invece, è un segnale di disturbo in una determinata cella che è dovuto a rilevamenti percepiti da celle adiacenti ad essa. Tuttavia, i Multi Pixel Photon Counter (MPPC) hanno delle ottime prestazioni in termini di afterpulse. Grazie ai



Fig. 5.5. MPPC (Multi Pixel Photon Counter) array. Serie S13361-3050.

materiali utilizzati e ai processi di miglioramento di produzione del wafer vengono prodotti MPPC con bassi afterpulse. In più, con l'MPPC serie S13361-3050, è stato migliorato anche l'aspetto relativo al crosstalk, in modo da avere degli output con una quantità bassa di questo disturbo rispetto ai prodotti precedenti [12].

5.3) Monitor a bassa intensità

Nel caso si abbia a che fare con fasci la cui intensità sia minore di 1 MHz (ossia un milione di particelle al secondo), i monitor standard CNAO racchiusi nelle BOX1 e BOX2 non sono in grado di rilevarli in maniera efficiente; non riescono, quindi, a contare il numero degli ioni primari passanti. L'obiettivo è quello di sviluppare un monitor per misurare intensità fino a 10 MHz, coprendo un'area attiva di circa 13 x 13 cm², in grado di fornire informazioni riguardanti la posizione del fascio con una risoluzione sperimentale su entrambe le direzioni spaziali x e y di circa 1 mm e il numero di ioni primari del fascio.

Il rivelatore consiste di due piani (x e y) di fibre plastiche scintillanti a sezione quadrata lette mediante SiPM. La dimensione dei SiPM determina la risoluzione e il numero di canali: sono previsti SiPM da 1 mm². Ogni layer è costituito da 128 fibre che verranno lette alternativamente sui due lati del piano, per un totale di 64 canali per lato e un numero complessivo di 256 canali per l'intero rivelatore. In Fig. 5.6 si può osservare l'assetto appena descritto per il layer x, costituito da 128 fibre e ai cui lati vi sono collegati i SiPM. Vi sarà, inoltre, una cornice in alluminio centrata rispetto all'isocentro del fascio, che sostiene meccanicamente i due layer di fibre.

Come sistema di lettura è stato scelto dal team INFN una scheda elettronica CAEN - DT5550W / A55CIT4 con FPGA di bordo adatta all'acquisizione di dati e finalizzata alla gestione dei 128 canali: 2 di queste schede possono gestire la lettura dell'intero rivelatore, dato che ciascuna di esse leggerà una determinata direzione spaziale del monitor (x e y). Questa scheda verrà analizzata attentamente nel capitolo successivo.

In Fig. 5.7 è mostrato come è stato progettato il design del monitor dal team INFN, con uno spazio dedicato all'alloggiamento delle due schede CAEN impiegate per l'acquisizione dati.

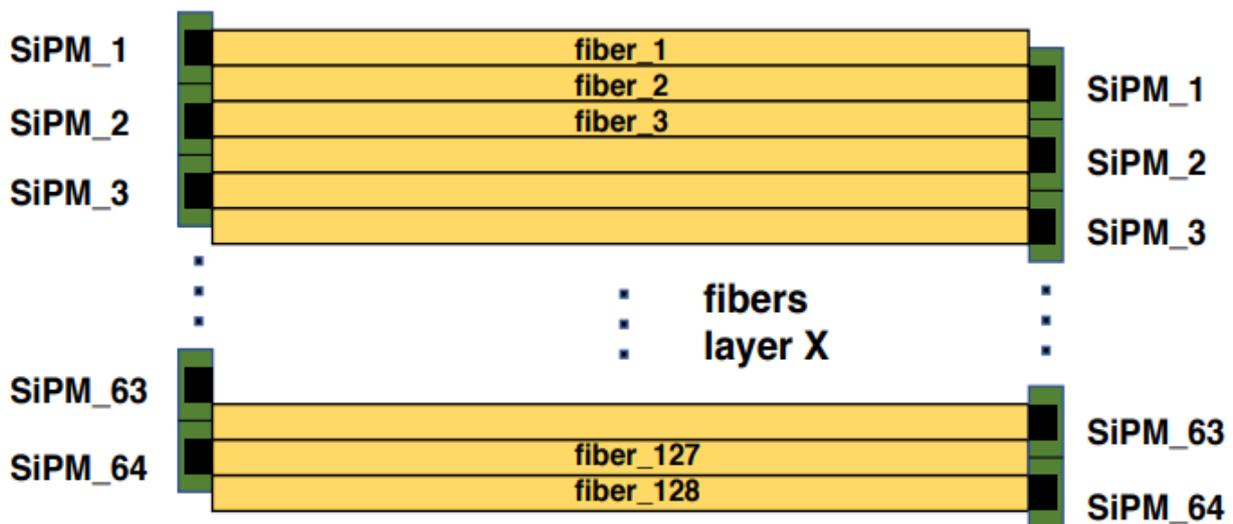


Fig. 5.6. Layer x di fibre scintillanti.

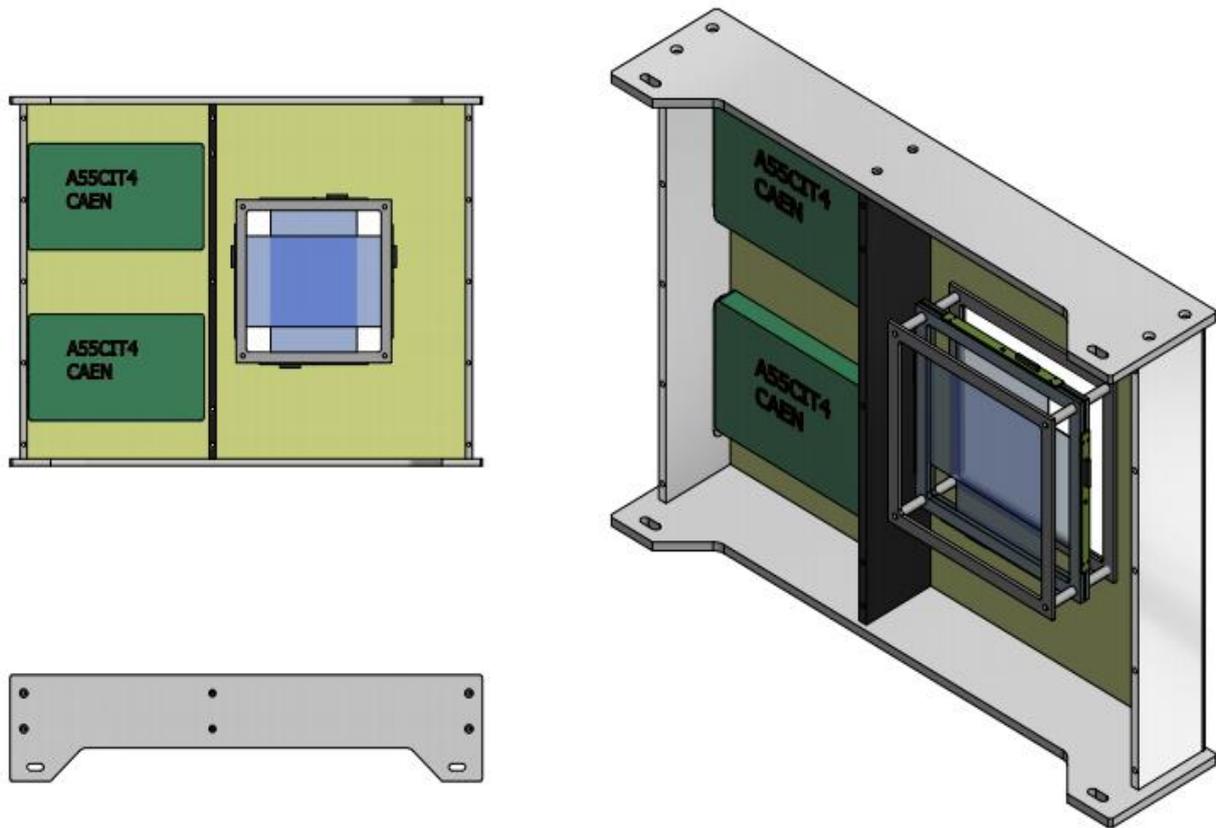


Fig. 5.7. Design del monitor con le schede per l'acquisizione dati.

Nelle Fig. 5.8, 5.9 e 5.10 viene mostrato il monitor in fase di assemblaggio. In particolare, nella Fig. 5.8 e 5.9 si vedono, rispettivamente, la parte frontale e posteriore del monitor; in Fig. 5.10 viene mostrato sempre il monitor ma in maniera più ravvicinata in modo tale da vedere meglio i due layer di fibre scintillanti, coperti da mylar alluminato. Il buio per le fibre viene garantito dalla paratia laterale e da due finestre in Tedlar che chiudono frontalmente e posteriormente il sistema di fibre scintillanti. Il monitor per basse intensità, inoltre, è progettato per essere posizionato nello stesso supporto delle BOX1 e BOX2 del CNAO. In Fig. 5.11 è illustrato il design del supporto del DDS sul quale è installato il monitor per le basse intensità.

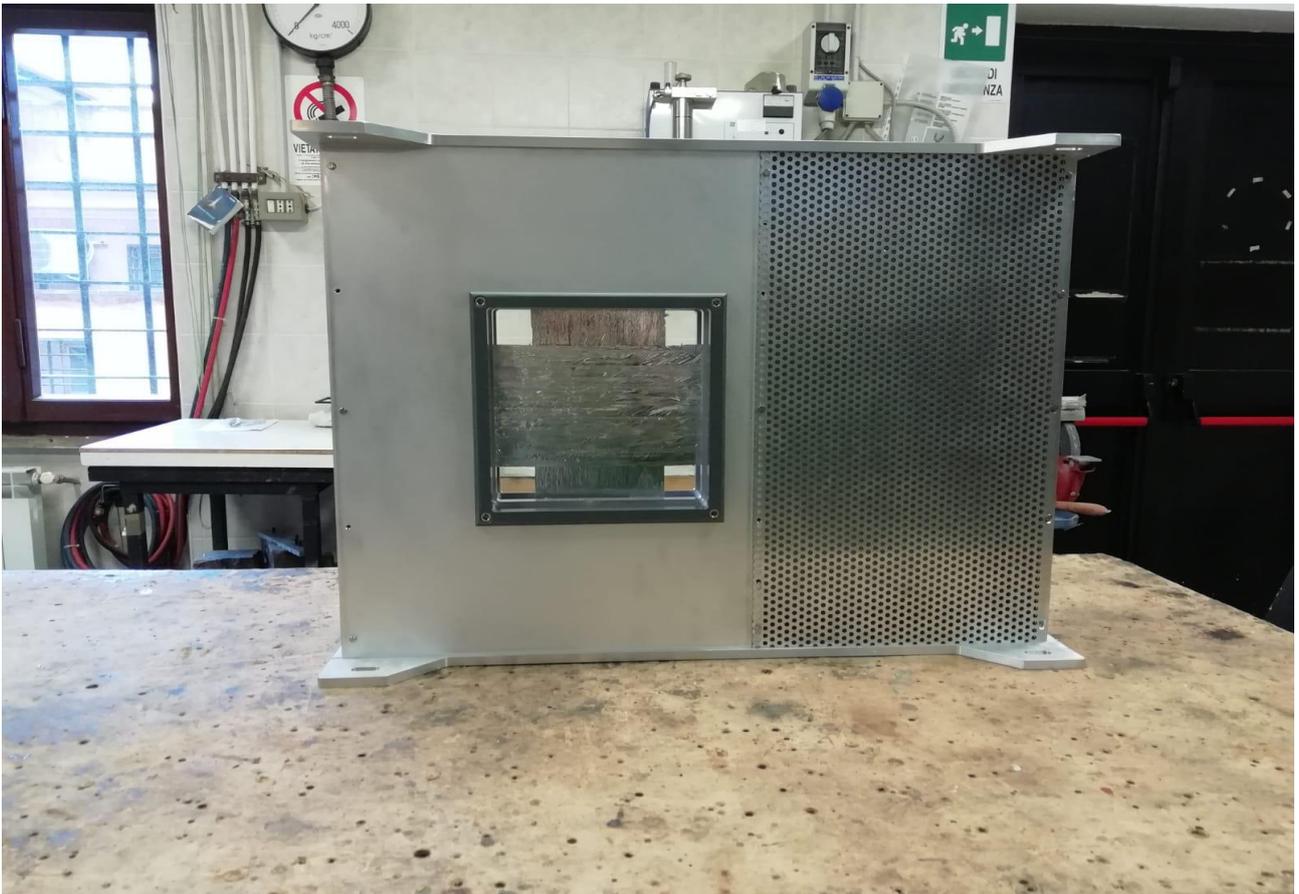


Fig.5.8. Visione frontale del monitor.



Fig. 5.9. Visione posteriore del monitor.



Fig. 5.10. Visione ravvicinata delle fibre scintillanti.

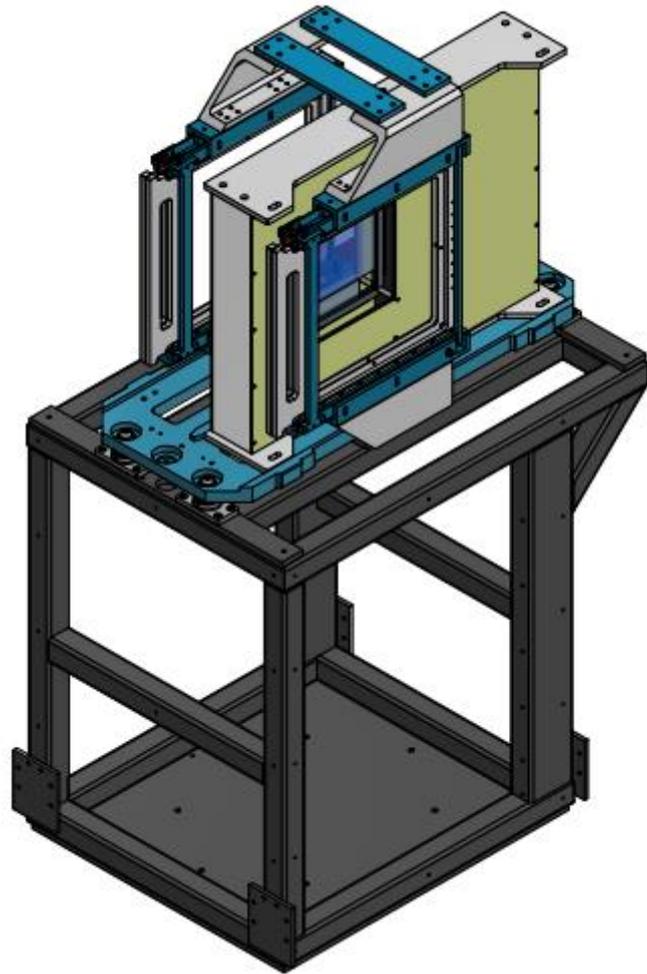


Fig. 5.11. Design del monitor per basse intensità alloggiato sul supporto del DDS.

Scheda di acquisizione (CAEN – DT5550W / A55CIT4)

Nel seguente capitolo vengono descritte le caratteristiche principali del sistema di lettura dati scelto. Esso è stato acquistato dalla Caen S.p.a. ed è costituito dalla scheda madre DT5550W che ha al suo interno una FPGA programmabile Xilinx XC7K160T (famiglia Kintex-7). La DT5550W ospita una scheda piggyback, A55CIT4, che va a completare il sistema di acquisizione, sulla quale si trovano i SiPM. I manuali delle due schede DT5550W e A55CITx possono essere consultati sul sito caen.it (UM6697 e UM7028) e sono messi in evidenza in [13].

6.1) DT5550W

La DT5550W è un sistema di lettura dati completo programmabile, basato su ASIC WeeROC (WeeROC è una famiglia di chip per la lettura di fotorivelatori). È una delle schede programmabili CAEN progettate per aiutare gli utenti a costruire un sistema di acquisizione personalizzato che si adatti alle esigenze dei loro rivelatori e alle condizioni sperimentali. Il sistema completo è costituito da una scheda madre che ospita una FPGA programmabile (DT5550W) e una scheda piggyback con un massimo di 4 ASIC WeeROC (A55xxxx). La DT5550W è completamente supportata da SCI-Compiler, un sistema di sviluppo grafico Windows per una facile programmazione FPGA. Questo strumento permette di sviluppare e compilare il codice del firmware utilizzando blocchi grafici che rappresentano determinate funzionalità. SCI-Compiler genera automaticamente il codice firmware VHDL partendo solo da blocchi logici e strumenti virtuali collegabili tra loro nella GUI e, inoltre, genera librerie C / C++ / C# / Python per il software personalizzato in Windows, Linux, MacOS, Android OS. Un firmware predefinito completo e pronto per l'uso viene fornito gratuitamente per ogni determinata scheda piggyback che si può utilizzare insieme alla DT5550W. Il firmware predefinito è stato sviluppato per sfruttare tutte le funzionalità degli ASIC con tutti i diversi metodi di lettura supportati da essi. L'utente, tuttavia, può aprire il design del firmware predefinito in SCI-Compiler e modificarlo oppure crearne uno nuovo. Lo SCI-5550W Readout Software è il software gratuito e open source Windows sviluppato per eseguire acquisizioni con la DT5550W, il quale funziona se la scheda è programmata con il firmware predefinito DT5550W precedentemente citato.

La dimensione della scheda è di 260 x 152 mm². Possiede delle linee di I/O digitali LEMO; per la precisione, dieci in totale: due sono linee di clock in ingresso e in uscita, mentre le altre otto linee LEMO sono linee general purpose digitali programmabili in ingresso o in uscita per segnali single-ended. L'impedenza d'ingresso Z_{in} di tali linee è di 50 Ω , così come l'impedenza caratteristica R_t . Uno strato di plexiglass copre la parte posteriore della scheda, fungendo da supporto. In Fig. 6.1 è mostrata la scheda DT5550W insieme ad una scheda piggyback. L'utente può usufruire di un cavo di comunicazione USB 3.0, di un cavo micro-USB per la programmazione dell'FPGA e di una dongle USB per la licenza del software SCI-Compiler. I tre oggetti sono mostrati nell'ordine in cui sono stati citati in Fig. 6.2. La DT5550W è alimentata da un alimentatore esterno che converte la tensione di rete da 220/110 V a 12V AC/DC. La massima tensione di lavoro è 12,8 V mentre la minima è 9 V. Sull'FPGA della DT5550W è installata una piccola ventola. Una temperatura



Fig. 6.1. Vista generale del sistema completo DT5550W con scheda piggyback A55PET2 montata superiormente [13].



Fig. 6.2. Cavi USB e Dongle USB [13].

eccessiva, infatti, può ridurre in primo luogo le prestazioni e la qualità della scheda e può anche danneggiarla. La DT5550W può funzionare nell'intervallo di temperatura compreso tra - 20 °C e + 50 °C.

La scheda madre DT5550W è la scheda responsabile dell'elaborazione dei segnali provenienti dagli ASIC nonché dell'interfaccia con il PC host. È dotata dell'FPGA Xilinx Kintex7 da utilizzare per lo sviluppo del firmware personalizzato e di un ADC (Analogic to Digital Converter) a 8 canali, 80 MSample / s, 14 bit, utilizzato principalmente per la lettura dei segnali multiplexati degli ASIC. Nella Fig. 6.3 sono evidenziati i componenti e i connettori più importanti della scheda madre DT5550W.

1. Interruttore ON/OFF per accensione e spegnimento della scheda.
2. Connettore Socket per alimentazione.
3. Connettore mini-USB JTAG. Questa porta permette di collegarsi direttamente all'FPGA per eseguire un veloce download del firmware, scrivendo direttamente nella memoria volatile in un paio di secondi.
4. LEMO Clock Input. Accetta segnali a 25 MHz, con 3,3 V di ampiezza e ha un'impedenza di ingresso di 50 Ω.

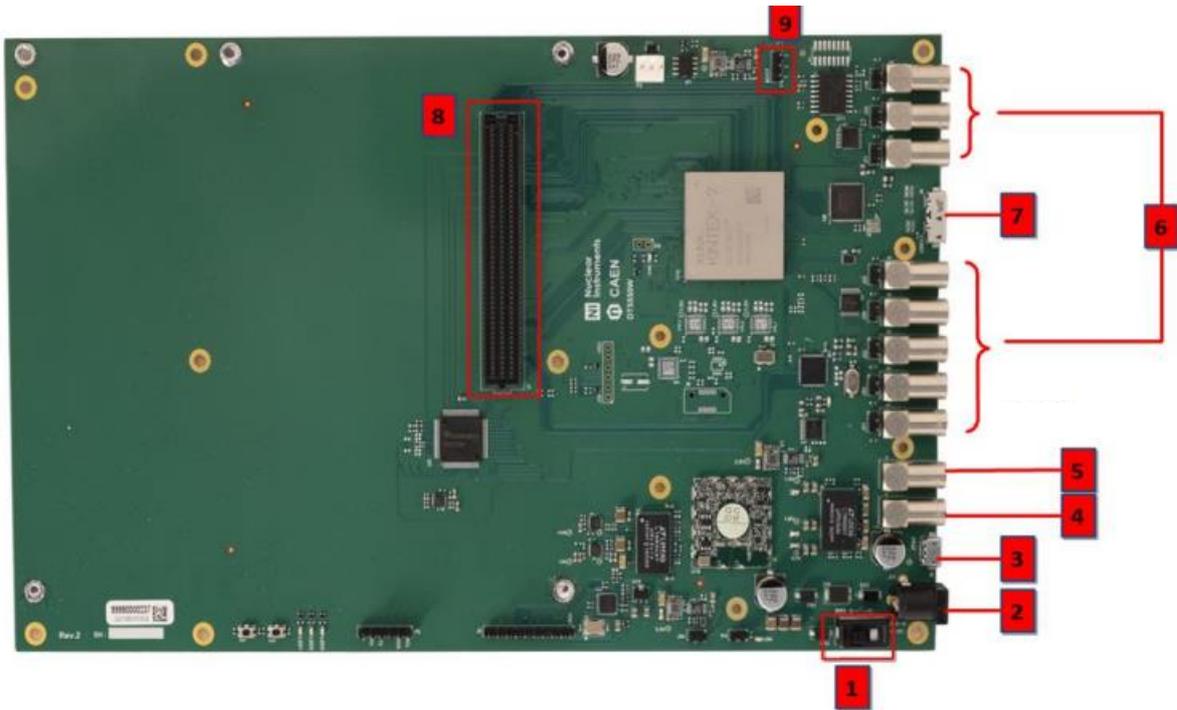


Fig. 6.3. Vista generale della scheda DT5550W senza alcuna scheda piggyback collegata [13].

5. LEMO Clock Output. Accetta segnali a 25 MHz, con 3,3 V di ampiezza e può essere usato per pilotare un cavo coassiale con impedenza caratteristica di 50 Ω .

6. Otto LEMO General Purpose I / O. Il firmware definisce la loro funzione (input o output). Questi I / O possono essere utilizzati per fornire segnali di sincronizzazione come trigger o segnali per comunicare con una scheda esterna. I connettori, quando funzionano in ingresso, possono essere utilizzati in modalità ad alta impedenza o a 50 Ω inserendo il relativo ponticello (jumper) dietro il connettore.

7. Porta USB 3.0. Consente il trasferimento dati ad alta velocità con il computer. Il bus di comunicazione è compatibile con USB 2.0. La velocità massima di trasferimento in modalità USB 3.0 è di 200 MB / s mentre in USB 2.0 è 24 MB / s.

8. Connettore mezzanina. Connettore array BGA utilizzato per collegare una delle schede piggyback CAEN fornite. Le linee di questo connettore sono direttamente collegate alle linee I / O dell'FPGA.

9. Jumper Bootloader. Il jumper deve essere posizionato in posizione 1-2 per avviare l'FPGA in modalità bootloader per l'aggiornamento del firmware, mentre la posizione 2-3 è per il normale funzionamento.

La DT5550W ha una struttura di organizzazione del clock molto versatile, come visibile in Fig. 6.4. Un generatore di clock CDCE62005 Texas Instruments viene utilizzato per generare il segnale di clock per tutti i dispositivi sulla scheda. Il clock di bordo, fissato a 80 MHz, è fornito all'ADC. Altri due segnali di clock sono connessi al connettore mezzanina e uno all'FPGA. A meno che non si modifichi il codice VHDL del firmware generato da SCI-Compiler, l'utente non può modificare la frequenza di clock. La personalizzazione dei clock non può essere eseguita in fase di utilizzo della

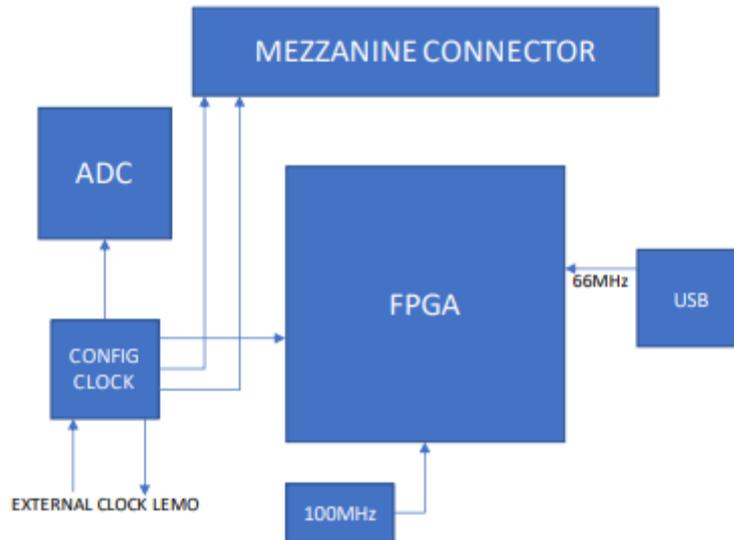


Fig. 6.4. Schema di distribuzione del clock sulla scheda DT5550W [13].

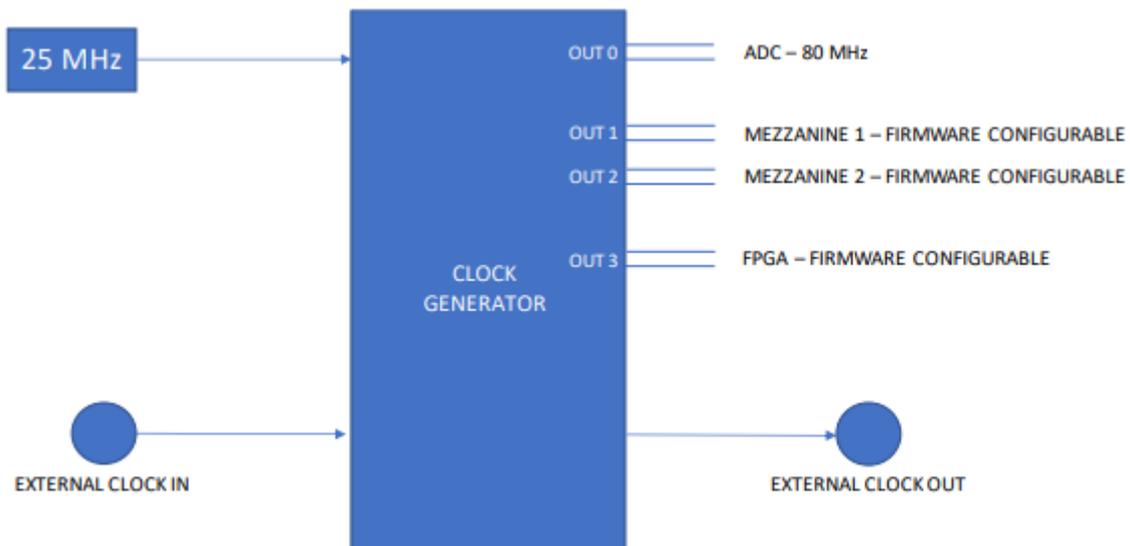


Fig. 6.5. Schema del generatore di clock [13].

scheda ma solo prima della compilazione del firmware.

In Fig. 6.5 è illustrato il funzionamento del generatore di clock. E' possibile vedere gli output di clock: uno fisso a 80MHz per l'ADC e gli altri tre, invece, configurabili attraverso il firmware. Inoltre, si vedono anche i segnali di clock esterni in input e output, citati precedentemente, passanti attraverso i due LEMO, i quali devono avere una frequenza di 25 MHz e un'ampiezza di 3,3 V.

La DT5550W supporta quattro metodi di lettura dei dati sfruttando i vari dispositivi che ha al suo interno, come ASIC, ADC e altri. Questi metodi sono descritti dettagliatamente sul manuale; tuttavia, per lo scopo di questo progetto, non è necessario riportarli in quanto nessuno di essi verrà usato, dato che è stato implementato nella scheda un nuovo firmware per adattarla alla tipologia di acquisizione dati del DDS. Questo vuol dire che lo SCI-5550W Readout Software non verrà utilizzato; esso, infatti, lavora in associazione con il firmware di default che implementa funzioni di

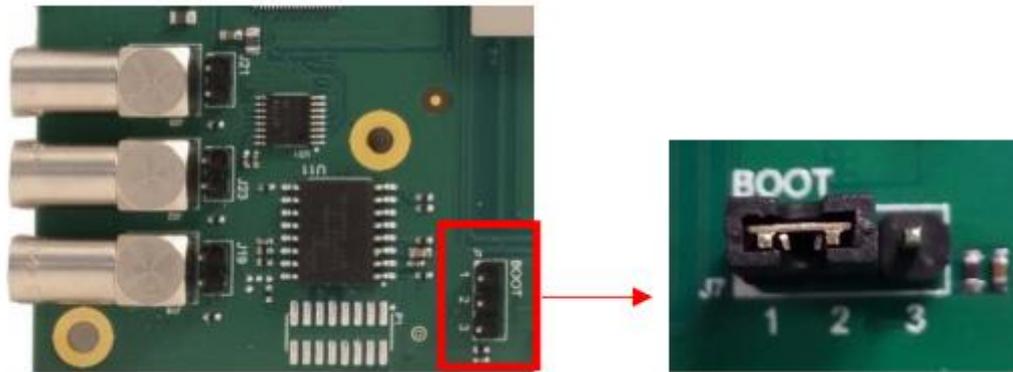


Fig. 6.6. Jumper del bootloader inserito in posizione 1-2 [13].

lettura e analisi dei dati.

È possibile modificare il firmware in due modi. In uno si utilizza il tool di aggiornamento del firmware, incluso in SCI-Compiler. Questo si interfaccia con il bootloader dell'FPGA per caricare il firmware nella memoria flash della DT5550W. Per entrare nell'FPGA in modalità bootloader, bisogna inserire il jumper “bootloader”, visibile nel punto 9 della lista dei componenti rappresentati in Fig. 6.3, nella posizione 1-2, come mostrato in Fig. 6.6. Quando si compie questa azione, la scheda deve essere spenta. L'altro modo consiste nell'aggiornare il firmware nella memoria FPGA volatile, utilizzando SCI-Compiler e la porta micro-USB della DT5550W visibile nel punto 3 della lista dei componenti rappresentati in Fig. 6.3. In questo modo è possibile collegarsi direttamente all'FPGA ed eseguire un veloce aggiornamento del firmware. Utilizzando questa porta, il firmware viene scritto in modo volatile direttamente nell'FPGA in un paio di secondi contro i pochi minuti necessari con la modalità bootloader flash.

6.2) A55CIT4

La scheda A55CITx è una scheda piggyback per la DT5550W, che ospita fino a quattro ASIC Citiroc 1A WeeROC. Quella utilizzata nel progetto in esame è la A55CIT4, dove il quattro finale indica la presenza di 4 Citiroc 1A su di essa. Può essere utilizzata insieme alla scheda madre DT5550W per eseguire misurazioni di energia ad alta risoluzione con SiPM e matrici di SiPM. I Citiroc 1A sono la soluzione ideale per sistemi di canali ad alta densità, con l'obiettivo di leggere sensori SiPM che estraggono informazioni in termini di energia, tempo e posizione. Ogni Citiroc 1A integra 32 canali di lettura. La scheda piggyback A55CITx è dotata di un modulo CAEN A7585D da utilizzare per l'alimentazione dei SiPM. L'alimentatore A7585D è saldato direttamente sul PCB. Il Citiroc 1A integra in un chip un intero sistema di lettura, richiedendo esternamente solo pochi componenti passivi, due linee di clock e una FPGA connessa con poche linee. Un sistema che sfrutti questo chip è la migliore soluzione per esperimenti in cui l'elevato numero di canali richiede la riduzione dei componenti hardware, per ragioni geometriche ed economiche. L'A55CITx piggyback può essere facilmente collegata alla scheda madre DT5550W per disporre un sistema di acquisizione dati completo per SiPM con un massimo di 128 canali. La lettura dei Citiroc è completamente gestita dal firmware precaricato sull'FPGA ospitata dalla scheda madre DT5550W.

La dimensione della scheda è di 160 x 145 mm², possiede un numero di canali pari a 128 e i 4 Citiroc 1A possono essere sfruttati, oltre che per la lettura dei SiPM per misurare l'energia delle particelle, anche per funzioni di imaging e spettroscopia. L'interconnessione con la scheda madre DT5550W avviene attraverso il connettore mezzanina che porta con sé più di 200 linee analogiche/digitali. Per la conversione digitale dei dati si usa l'ADC a 14 bit presente sulla DT5550W già citato nel Par. 6.1. A seconda del modello piggyback, l'utente è dotato di una o due dark box stampate in 3D (Fig. 6.7), che possono essere utilizzate per testare i SiPM.

In Fig. 6.8 è mostrato dall'alto l'insieme del sistema DT5550W / A55PET4 (è equivalente al DT5550W / A55CIT4 con l'unica differenza che al posto dei Citiroc 1A vi sono i Petiroc 2A). Nella Fig. 6.9 sono evidenziati i componenti e i connettori più importanti sul lato anteriore e posteriore di una A55CIT2:

1. 2 ASIC Citiroc 1A saldati sulla scheda.
2. Matrice Hamamatsu a 64 canali S13361-3050, collegata direttamente ai connettori disponibili sulla scheda piggyback.



Fig.6.7. Dark box [13].



Fig. 6.8. Vista dall'alto della scheda piggyback montata sulla scheda madre DT5550W [13].

3. Sensore di temperatura.
4. Connettore mezzanina. Connettore array BGA utilizzato per collegare la scheda piggy back alla scheda madre DT5550W.
5. Alimentatore SiPM A7585D saldato sul lato posteriore della scheda piggyback.

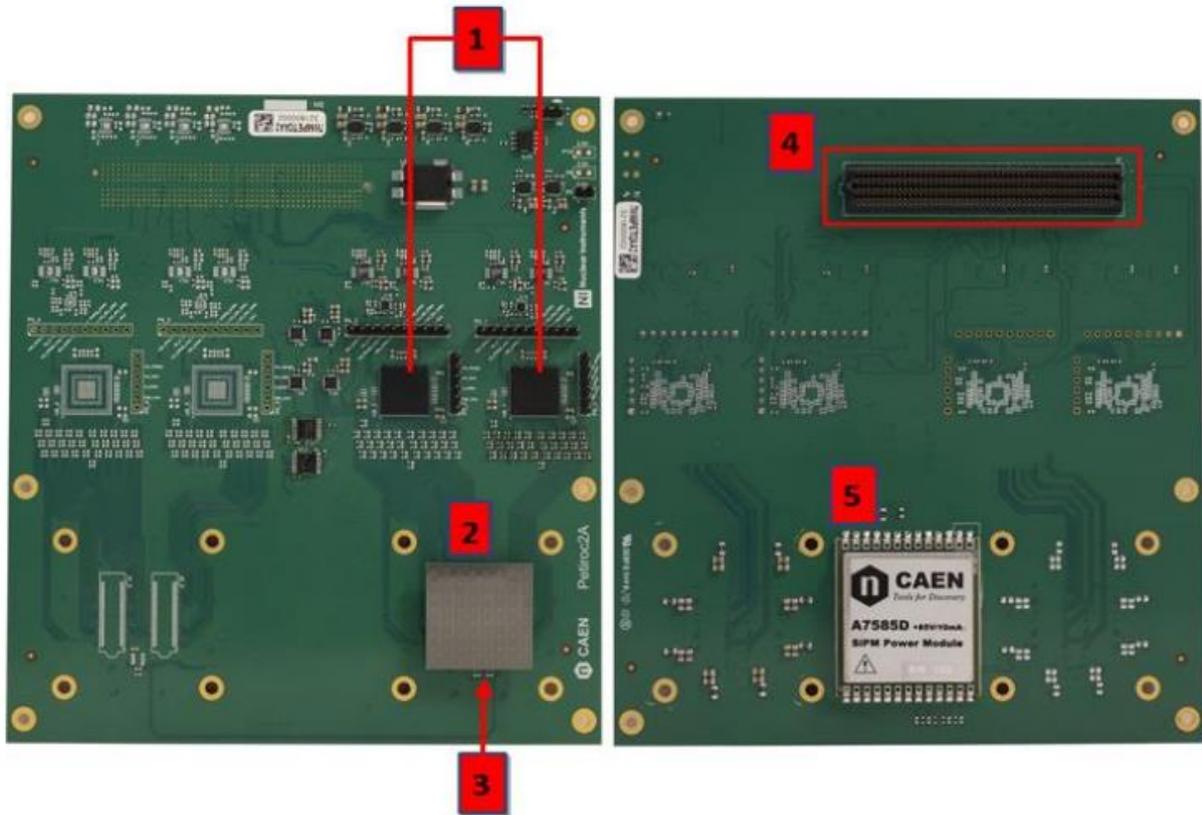


Fig. 6.9. Vista generale (parte anteriore a sinistra e parte posteriore a destra) della scheda piggyback A55CIT2 [13].

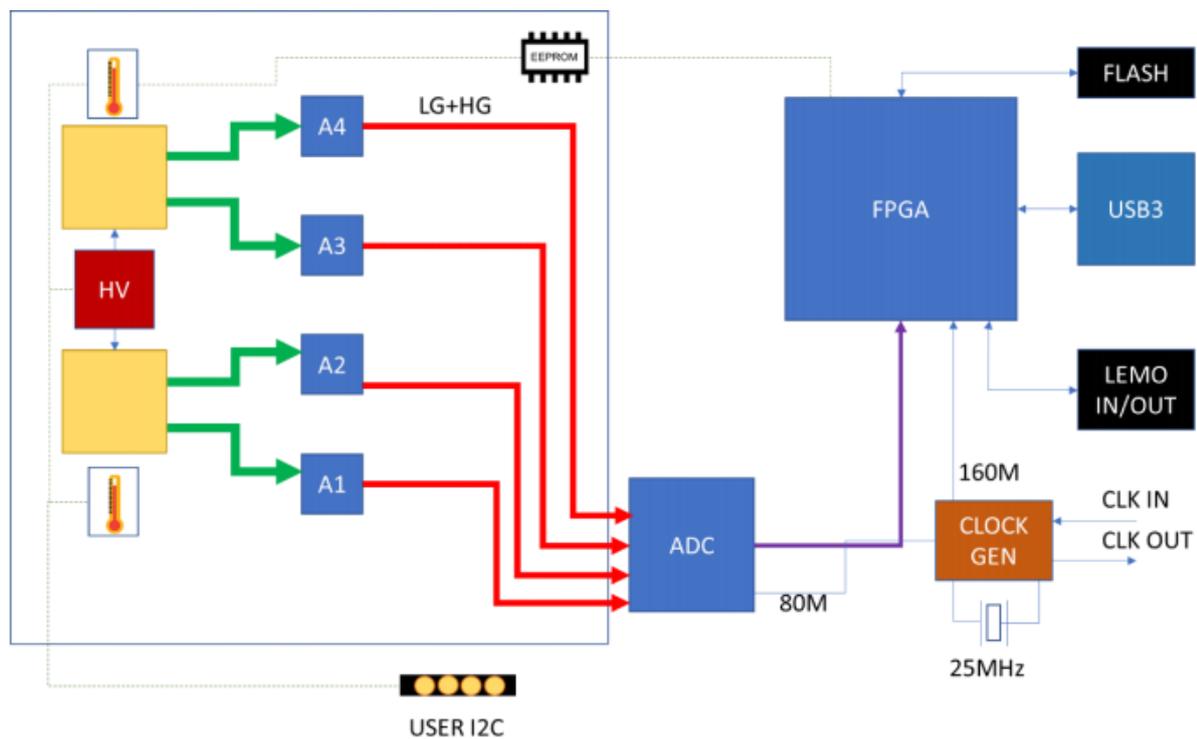


Fig. 6.10. Diagramma a blocchi che mostra i componenti principali delle schede A55CITx e DT5550W, nonché la loro interconnessione [13].

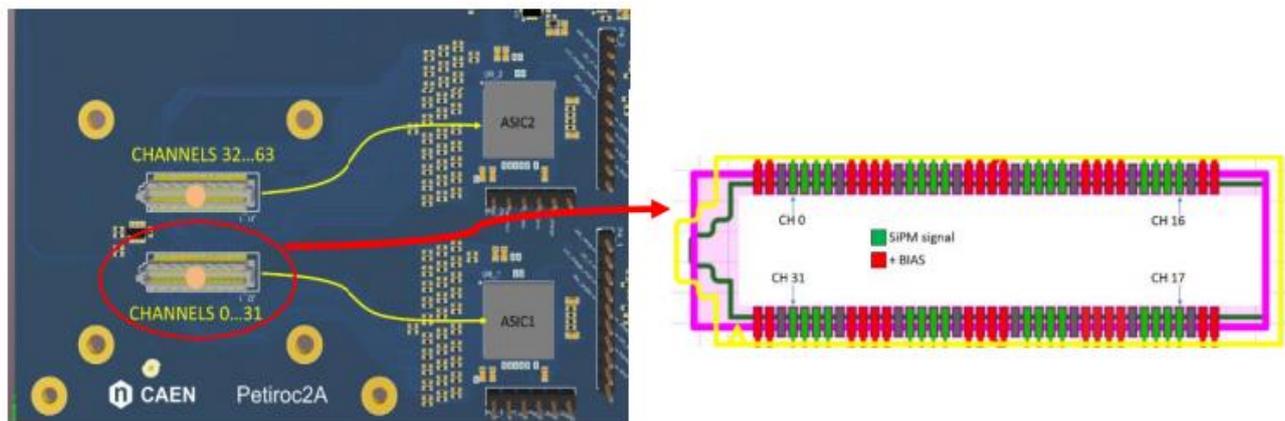


Fig. 6.11. Coppia di connettori Samtec SS4-40-3.0-L-D-K-TR montati su scheda A55CITx [13].

Come già detto, la scheda A55CITx è progettata per essere collegata alla scheda madre DT5550W tramite il connettore mezzanina, in modo che quest'ultima scheda fornisca tutte le linee di alimentazione, i clock, i segnali di controllo e di trasferimento necessari. Nella Figura 6.10 è mostrata l'interconnessione della scheda piggyback con i componenti della scheda madre.

La scheda A55CIT4 ospita due coppie di connettori Samtec SS4-40-3.0 L-D-K-TR, compatibili con la matrice di SiPM Hamamatsu S13361-3050. In Fig. 6.11, a sinistra, sono evidenziati i 64 canali disponibili per la connessione con una matrice. A destra, invece, è mostrata la piedinatura del connettore utilizzando dei colori come esempio: i pin verdi sono collegati ai canali di ingresso dei Citiroc, mentre i pin rossi sono le linee della tensione di alimentazione. Le linee di alimentazione per i SiPM sono raggruppate otto per otto e collegate al modulo A7585D tramite una resistenza da 50 Ω in serie. L'A7585D alimenta i SiPM con un'alta tensione positiva. Gli anodi dei SiPM sono collegati indipendentemente alle linee di ingresso del Citiroc, mentre i catodi sono collegati alla tensione di alimentazione, come si può vedere in Fig. 6.12. Affinché il SiPM generi un impulso positivo che venga elaborato dal Citiroc, il segnale proveniente dal SiPM viene fatto passare attraverso una configurazione di cortocircuito, costituita da una resistenza in serie 0 Ω e nessun componente montato verso terra (in Fig. 6.12 N.M. sta per componente Not Mounted). Nello stadio di ingresso dell'ASIC, un DAC (Digital to Analog Converter) a 8 bit consente di regolare l'alimentazione individualmente per ogni canale del SiPM. La tensione può essere regolata nel range 0-1 V. Questo è importante per regolare il guadagno di ogni elemento, soprattutto quando si collegano singoli SiPM, che possono avere cadute di tensioni differenti fino a 400 mV.

Vicino a ciascun connettore SiPM ospitato sulla scheda A55CITx, è saldato un sensore di temperatura. Questo sensore viene letto dall'FPGA e le informazioni vengono utilizzate dal SCI-5550W Readout Software per la compensazione attiva della tensione di alimentazione dei SiPM in caso di variazioni di temperatura.

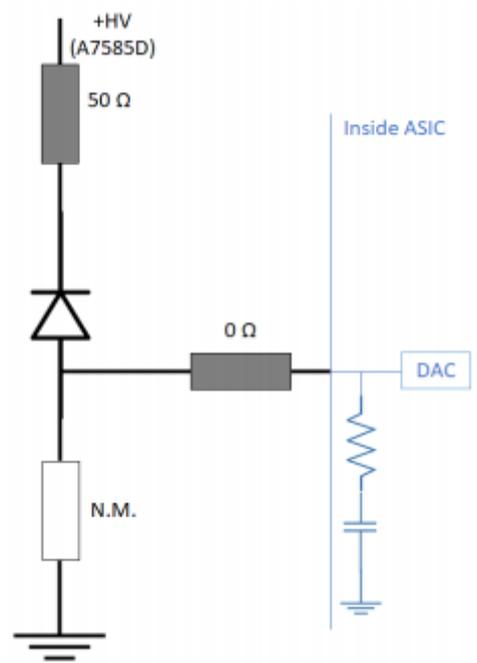


Fig.6.12. Schema di connessione SiPM alle linee di ingresso ASIC in configurazione standard [13].

SCI-Compiler

In questo capitolo si introduce un software innovativo per una facile programmazione FPGA, chiamato SCI (entific)-Compiler, un compilatore basato su diagrammi per semplificare lo sviluppo del firmware. Il manuale di utilizzo è disponibile sul sito caen.it (UM6519) ed è riportato in [14]. Il software sfrutta un metodo che si basa su un'interfaccia di programmazione grafica costituita da blocchi specificatamente sviluppati per applicazioni fisiche. Utilizza un set di librerie precostruito, contenente macroblocchi con funzioni complesse che possono essere collegati tra loro per disegnare un diagramma che descrive la struttura del firmware. SCI-Compiler, partendo da un diagramma a blocchi grafico, genera automaticamente un codice VHDL che implementa le funzioni richieste. Inoltre, è in grado di generare librerie C / C ++ / C # / Phyton, driver e codici di esempio da utilizzare in Windows, Linux e MacOS per il software personalizzato basato sul firmware compilato. Inoltre, SCI-Compiler fornisce i blocchi necessari per leggere e testare gli ASIC WeeROC, consentendo all'utente di sviluppare un sistema completo operante che utilizza tali ASIC.

Di seguito sono riportate le caratteristiche principali di SCI-Compiler:

- Possibilità di progettare e generare un firmware completo basandosi sulla programmazione di schemi grafici a blocchi.
- Disponibilità di blocchi di programmazione che implementano funzioni complesse e funzionalità di strumenti utilizzati nelle applicazioni fisiche, come integrazione di carica, misure di ampiezza, oscilloscopio, ecc.
- Ampio set di blocchi funzionali, per semplificare la progettazione del firmware e ridurre al minimo i tempi di sviluppo.
- I blocchi schematici non sono semplicemente funzioni logiche o flip flop, ma macroblocchi complessi che implementano funzionalità complete come strumenti per misurare ampiezza, energia, tempo di arrivo dei segnali in ingresso, ecc.
- Generazione di un codice VHDL / Verilog che può essere tradotto in un file di configurazione FPGA.
- Compilazione del codice e programmazione della scheda grazie all'esecuzione automatica degli strumenti Vivado / Quartus per FPGA Xilinx / Intel Altera.
- Generazione di librerie C / C ++ / C # / Phyton.
- Generazione di tutto il codice necessario per implementare l'interfaccia di comunicazione tra hardware e computer (supporto bus di comunicazione USB, Ethernet e VME).

SCI-Compiler è un generatore di codice VHDL / Verilog, non un compilatore FPGA. Per questo motivo, Xilinx Vivado o Intel Quartus devono essere installati sul PC locale per compilare il firmware generato con SCI-Compiler, creare il bitstream e programmare l'FPGA. La scheda DT5550W / A55CIT4 utilizzata ha una FPGA Xilinx. Per questo motivo è stato installato sul computer il software Xilinx Vivado per la compilazione del firmware e la programmazione dell'FPGA. La versione completa di SCI-Compiler funziona con una licenza e una dongle USB fisica da collegare al PC durante l'utilizzo del software.

7.1) Descrizione GUI del software

L'interfaccia principale è organizzata come segue ed è mostrata in Fig. 7.1:

Toolbars: ci sono due barre degli strumenti. La barra degli strumenti 'Home' contiene i controlli necessari per gestire le operazioni del diagramma, dell'editor e della compilazione mentre la barra degli strumenti 'Tools Box' contiene i pulsanti per inserire nello schema tutti i blocchi che possono essere implementati nella scheda FPGA.

Resources and Settings: questa sezione ha due schede. La prima è la scheda 'Project File', la quale, a sua volta, è divisa in due sezioni. La sezione in alto permette di gestire i file del progetto che vengono organizzati con una visualizzazione ad albero come si vede in Fig. 7.2. Il simbolo affianco alla parola 'top' in Fig.7.2 rappresenta il file principale del progetto. Al di sotto si vedono due file detti 'subdesign' del file principale del progetto. La sezione in basso della scheda 'Project File', invece, permette di vedere le informazioni di un blocco del diagramma attraverso l'"Help" online. La seconda scheda si chiama 'Hardware Settings' e consente di selezionare i connettori della scheda e le impostazioni dei registri da configurare.

Development GUI: questa sezione permette di progettare graficamente il firmware FPGA sfruttando e interconnettendo i blocchi disponibili (scheda 'Diagram'), di configurare i connettori della scheda e le impostazioni dei registri (scheda 'Editor') e di leggere i messaggi di output del processo di compilazione (scheda 'Compiler Output').

Status: questa parte mostra, a sinistra, le informazioni sulla licenza e lo stato del programma. Questo può essere nello stato "Compiling" se vi è una compilazione in esecuzione, "Programming" se vi è un processo di programmazione dell'FPGA in corso e "Idle" se non ci sono processi che coinvolgono l'esecuzione di Vivado / Quartus. A destra, si colloca una barra per indicare l'avanzamento del processo in corso, che può essere lo stato di avanzamento di compilazione del firmware o lo stato di avanzamento del processo di programmazione dell'FPGA.

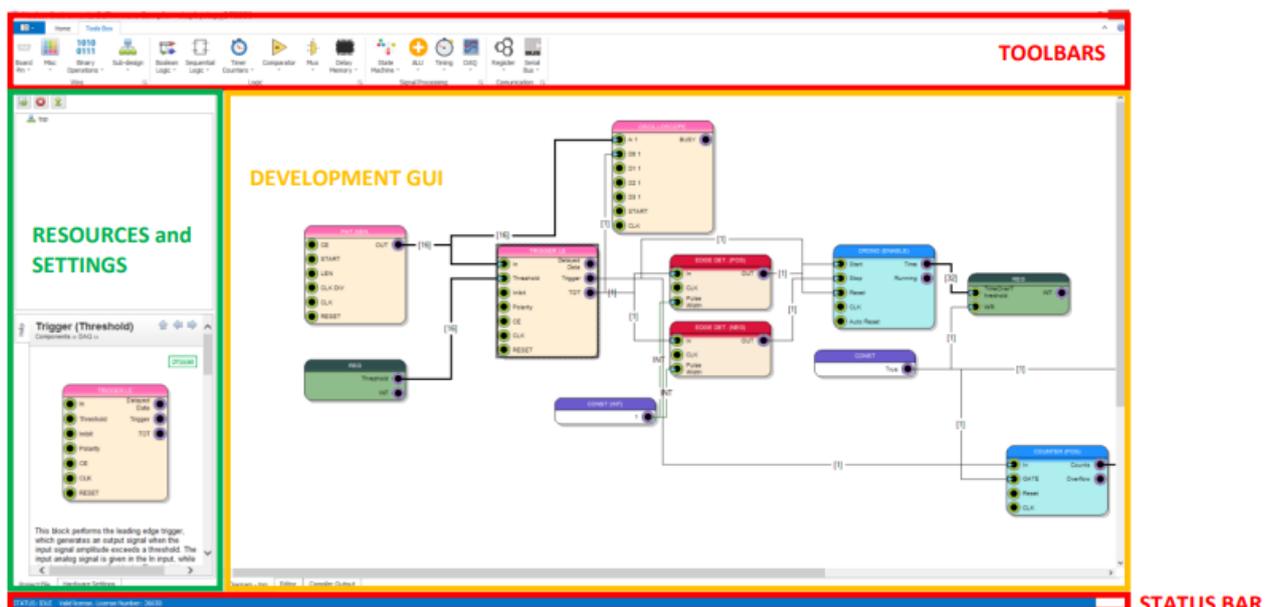


Fig. 7.1. Interfaccia grafica principale di SCI-Compiler [14].

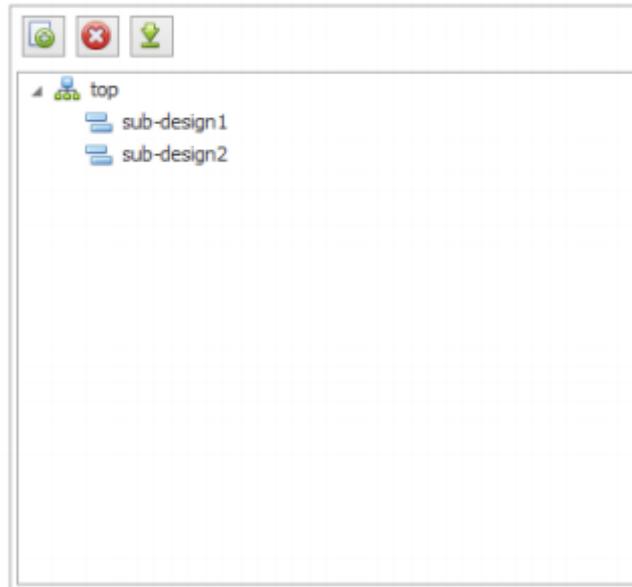


Fig. 7.2. Sezione superiore della scheda 'Project File' [14].

7.1.1) Toolbar “Home”

Il toolbar "Home" (Fig. 7.3) contiene alcuni controlli organizzati in sottogruppi dedicati a diversi tipi di azioni.

Il sottogruppo “File” contiene i comandi base per la creazione di un nuovo progetto, l’apertura di un progetto esistente e per il salvataggio. Il sottogruppo “Tools” è anch’esso molto base, con comandi molto comuni come Copia, Incolla, Taglia, Annulla e Ripeti.

Il sottogruppo "Compiler" contiene i controlli relativi alla procedura di compilazione. Il pulsante "Compile" avvia la compilazione sul computer dell'utente: viene generato un file .vhdl a partire dal firmware progettato graficamente dall'utente. Viene creato un file .tcl e il programma Vivado o Quartus, a seconda della scheda di destinazione scelta per il progetto, verrà eseguito dalla shell. Il pulsante “Compile SDK” permette di compilare l'SDK (Software Development Kit) C / Python compatibile con il progetto firmware corrente. Le relative librerie vengono salvate nella cartella del progetto. Il pulsante "Program FPGA" permette di programmare l'FPGA della scheda utilizzando il file bitstream generato durante il processo di compilazione del firmware, collegando la scheda al computer con l’apposito cavo programmatore micro-USB (Vivado o Quartus vengono richiamati automaticamente dalla shell). Il pulsante "Settings" permette di impostare la configurazione dei compilatori Vivado o Quartus e di definire il percorso per il salvataggio della cartella con all’interno il progetto e le varie cartelle generate dal processo di compilazione. Il pulsante "Resource Explorer" apre una finestra di interfaccia che permette di connettersi con l'hardware e testare il firmware.



Fig. 7.3. Toolbar “Home” di SCI-Compiler [14].

Il sottogruppo "Flash Tool" contiene i comandi necessari per generare il file flash del firmware e avviare il tool di aggiornamento del firmware FPGA per le schede supportate da SCI-Compiler. L' "Help" permette di accedere ad una guida completa del software con le descrizioni di tutti i blocchi funzionali disponibili.

7.1.2) Toolbar "Tools Box"

Il toolbar "Tools Box" (Fig. 7.4) contiene tutti i blocchi implementati precompilati per la scheda di destinazione, divisi in sottogruppi a seconda della loro categoria.

Il sottogruppo "Wire" contiene i seguenti sottomenù:

- il "Board Pin" consente di utilizzare i pin della scheda I / O e implementare le modalità di lettura per gli ASIC WeeROC;
- il "Misc" contiene valori costanti, variabili booleane, clock e conversioni tra valori binari e decimali; contiene anche la funzione "User HDL" che permette all'utente di creare un blocco grafico basato su un proprio codice HDL personalizzato;
- il "Bitwise Operations" consente di implementare le principali operazioni binarie bit per bit;
- il "Sub-design" permette di selezionare un subdesign e di inserirlo nel diagramma principale.

Nel sottogruppo "Logic" vi sono i seguenti sottomenù:

- il "Boolean Logic" permette di implementare tutte le operazioni base della logica booleana;
- il "Sequential Logic" dispone di rilevatori di fronti, latch e flip-flop;
- il "Timer Counter" contiene cronometri, timer, contatori e frequenzimetri;
- il "Comparator" dispone di blocchi che possono essere utilizzati per confrontare i segnali;
- il "Mux" permette di implementare multiplexer e demultiplexer;
- il "Delay Memory" contiene blocchi per implementare ritardi, serializzazioni e deserializzazioni, RAM, ROM e vari tipi di buffer.

Il sottogruppo "Signal Processing" contiene:

- il menu "State Machine" per implementare una macchina a stati finiti;
- il menu "ALU" contenente le operazioni di base delle unità logico-aritmetiche;
- il menù "Timing" contiene blocchi che implementano funzioni come il Time to Digital Converter e il Single Channel Analyzer;
- il menu "DAQ" consente di implementare blocchi come l'oscilloscopio, il digitalizzatore, il restauratore della linea di base, l'integratore della carica, il filtraggio trapezoidale, la logica di trigger, l'analizzatore logico e l'acquisizione spettrale.

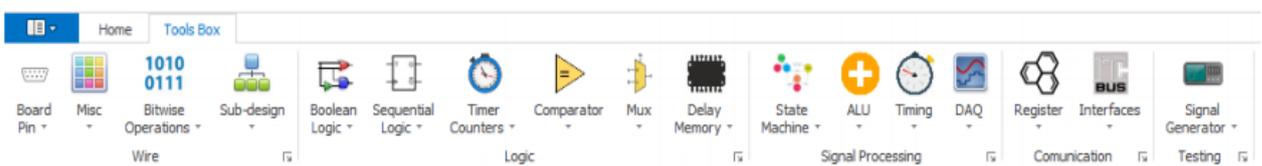


Fig. 7.4. Toolbar "Tools Box" di SCI- Compiler [14].

Il sottogruppo "Communication" è suddiviso nei seguenti sottomenù:

- il "Register" permette di leggere e scrivere direttamente sul registro della scheda;
- L' "Interfaces" offre la possibilità di scegliere tra i protocolli di comunicazione I2C, SPI o UART.

Il sottogruppo "Testing" contiene il menù "Signal Generator", per aggiungere segnali di test in ingresso al diagramma.

7.2) Programmazione con SCI-Compiler

Un esempio di diagramma molto semplice viene raffigurato in Fig. 7.5. In fase di programmazione vengono aggiunti blocchi facendo clic sulla voce di un sottomenu nel toolbar "Tools Box". Ogni blocco può essere abbinato ad un altro blocco collegando il cerchio che ne identifica l'uscita al cerchio che definisce l'ingresso dell'altro. Il filo che si crea corrisponde ad un segnale: la freccia ad una delle estremità del filo indica la direzione del segnale, mentre il numero riportato tra parentesi quadre lungo il filo rappresenta la dimensione del segnale, in termini di bit. È possibile collegare due blocchi solo se il tipo (intero o vettore) e la dimensione del segnale di uscita di un blocco corrispondono alle caratteristiche richieste del segnale di ingresso dell'altro blocco. Inoltre, l'uscita di un blocco può collegarsi a più di un altro blocco, mentre l'ingresso di un blocco può avere una sola connessione. Se l'ingresso e l'uscita principali (IN e OUT di ogni blocco) vengono lasciati scollegati, si verifica un errore durante la compilazione che viene notificato sul "Compiler Output". Agli altri segnali di ingresso, se vengono lasciati scollegati, viene assegnato un valore di default.

I componenti SCI-Compiler sono separati in:

- componenti detti "stand-alone";
- componenti detti "memory mapped";

I componenti stand-alone sono blocchi che non richiedono una comunicazione con un PC host per funzionare. Ad esempio, i blocchi di logica booleana, quelli riguardanti le operazioni matematiche,

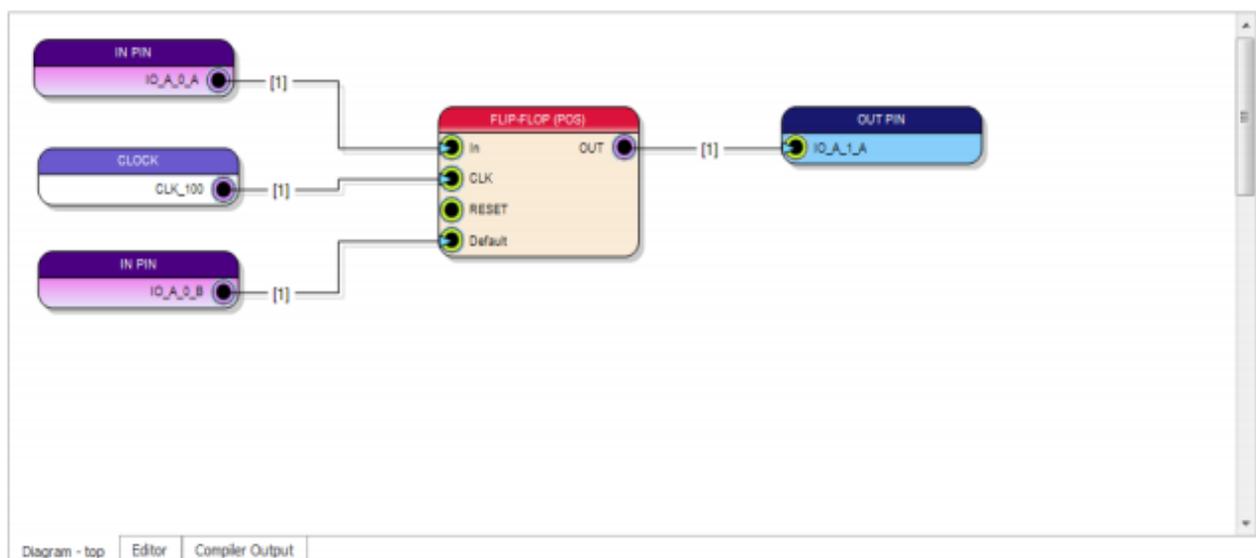


Fig. 7.5. Esempio di diagramma nella scheda "Diagram-top" [14].

il trigger trapezoidale e molti altri sono tutti componenti autonomi. Hanno diversi parametri operativi configurabili ma non hanno alcuna interfaccia con il PC. I componenti memory mapped sono tutti componenti che richiedono una comunicazione con il PC per poter funzionare. Se ne possono elencare alcuni: oscilloscopio, modulo di imaging, I2C master, analizzatore logico, registri di lettura e scrittura. Ad esempio, se si vuole creare un oscilloscopio, SCI-Compiler aprirà una finestra di configurazione in cui l'utente può configurare i parametri dell'endpoint. Il nome dell'endpoint deve essere univoco nell'intero progetto. L'endpoint è rappresentato con un blocco virtuale standard con una piccola icona in basso e il nome dell'endpoint, come visibile in Fig. 7.6 (destra). Tutti gli endpoint mappati in memoria vengono visualizzati nella scheda "Memory Mapping", come mostrato nella Fig. 7.7. Quando viene aggiunto un nuovo endpoint, l'utente deve cliccare sul pulsante 'Refresh', visibile in Fig. 7.7, per ricaricare l'elenco. SCI-Compiler non assegna automaticamente un indirizzo ai nuovi endpoint. L'indirizzo è un numero che deve essere univoco e rappresenta la regione di memoria assegnata a un particolare endpoint. L'utente può specificare solo l'indirizzo di base dell'endpoint mappato in memoria, dopodiché SCI-Compiler calcolerà automaticamente l'area di memoria richiesta dalla periferica. L'area di memoria viene visualizzata nella colonna "Area Size". Diverse periferiche utilizzano un'area di memoria più alcuni registri di controllo. Il numero del registro di controllo viene visualizzato nella colonna "Control Registers" (Fig. 7.7). L'indirizzo può essere assegnato manualmente o assegnato automaticamente. In generale, gli utenti possono semplicemente cliccare sul pulsante 'Refresh' e poi 'Auto Assign' per generare automaticamente uno spazio di indirizzi coerente con tutte le periferiche mappate correttamente sul bus. È possibile assegnare un indirizzo manualmente marcando la colonna "Manual Assigned" e inserire l'indirizzo nella colonna "Phy Hex Address". È molto importante evitare le sovrapposizioni tra gli spazi degli indirizzi, che potrebbero causare conflitti tra i diversi componenti.

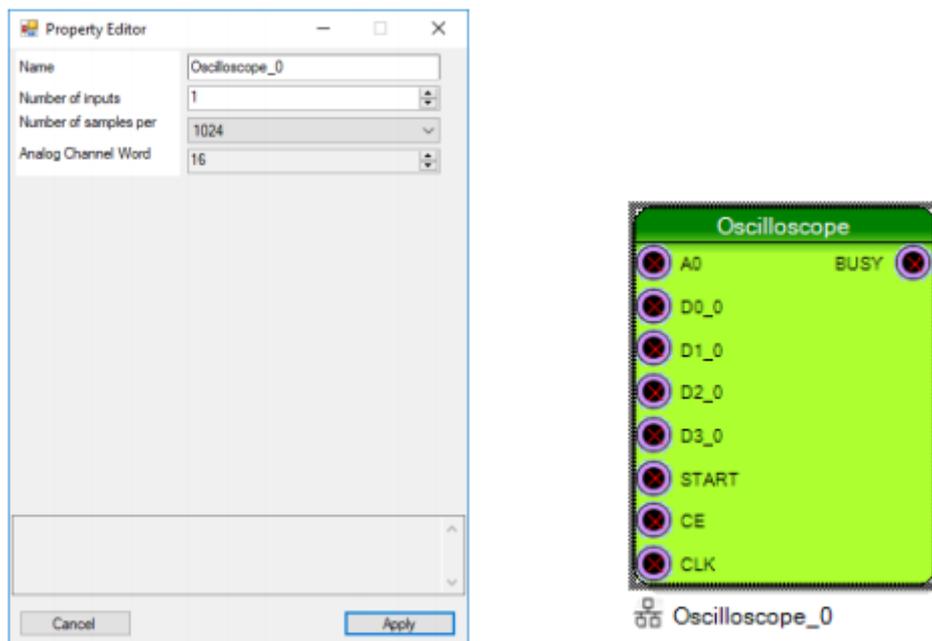


Fig.7.6. Configurazione di un blocco memory mapped (sinistra) e la sua rappresentazione nel diagramma a blocchi (destra) [14].

Phy Address	Friendly Name	Type	Area Size	Control Registers	Phy Hex Address	Manual Assigned	Reset Value
top	Counter_0	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_1	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_2	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_3	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_4	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_5	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_6	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_7	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_8	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_9	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_10	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_11	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_12	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_13	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_14	mmcRegister	1	1	0	<input type="checkbox"/>	
top	Counter_15	mmcRegister	1	1	0	<input type="checkbox"/>	
top	RESET	mmcRegister	1	1	0	<input type="checkbox"/>	

Fig.7.7. Scheda "Memory Mapping" che mostra la lista dei blocchi memory mapped [14].

7.2.1) Compilazione del firmware

Quando l'utente ha finito di posizionare i blocchi sullo schema e di collegarli, può compilare il suo firmware progettato graficamente. Prima di iniziare la compilazione, l'utente deve anche impostare correttamente il percorso dei file eseguibili di Vivado / Quartus nella finestra delle impostazioni. La compilazione del firmware sul computer si avvia premendo il pulsante "Compile" nel toolbar "Home". Il pulsante viene immediatamente trasformato nel pulsante "Stop Compilation" per interrompere il processo di compilazione, se necessario. Automaticamente, viene visualizzata la sezione "Compiler Output" per mostrare i messaggi che descrivono i processi in corso. La compilazione del firmware è composta da varie fasi: l'analisi e la sintesi di tutti i componenti, la loro mappatura, la loro collocazione e instradamento.

La prima fase di compilazione viene eseguita da SCI-Compiler e consiste nella generazione dei pezzi di codice VHDL dal diagramma progettato dall'utente. Quindi, per ogni blocco utilizzato nel diagramma, viene generato un codice HDL nella cartella "pcores" all'interno della cartella "HDL" del progetto corrente. A questo punto, il codice HDL può essere assemblato per tutti i blocchi utilizzati nello schema principale e tutti i suoi subdesign, e vengono generati i corrispondenti file .vhd nella cartella "HDL" del progetto corrente.

La seconda fase di compilazione prevede la creazione da parte di SCI-Compiler delle librerie C /

Phyton e del loro relativo codice di esempio nella cartella "library" del progetto corrente. Inoltre, nella stessa cartella, viene creato un file .json contenente tutti i registri, oscilloscopi, e blocchi di analizzatori logici definiti per trasmettere tutte le informazioni necessarie al tool "Resource Explorer", che permette di testare le funzionalità del firmware progettato.

Nella terza fase di compilazione, SCI-Compiler genera un file .tcl nella cartella "HDL" che serve per eseguire dalla shell il programma Vivado o Quartus, a seconda della scheda di destinazione scelta per il progetto, ad esempio, per la DT5550W, si eseguirà Vivado. Si ricorda, infatti, che Vivado e Quartus sono, rispettivamente, i software Xilinx e Altera che consentono la compilazione del firmware FPGA e la generazione del bitstream. L'output del compilatore, durante l'esecuzione del software, viene reindirizzato alla scheda "Compiler Output" di SCI-Compiler e visualizzato in nero, con i messaggi di 'warning' visualizzati in arancione e i messaggi di errore riportati in rosso; un esempio è riportato in Fig. 7.8. In blu sono visualizzati i messaggi che descrivono le operazioni eseguite da SCI-Compiler stesso, come la creazione di una nuova cartella di progetto, l'apertura di un progetto, la creazione di tutti i file e le cartelle necessarie alla compilazione del firmware.

La fase finale di compilazione è rappresentata dalla generazione del file bitstream che serve per programmare l'FPGA, ovvero un file di configurazione compatibile con la piattaforma supportata: nel caso della DT5550W si tratta di un file .bit. Tutti i file di output vengono generati nella cartella "output" del progetto corrente.



```
Start compiling project: list_D
Creating register map...
Mapping register count on address: 0x00010000
Processing page: top (b9a3479d-986e-4935-896b-17a37705c3e4)
Generate components...
Creating signal map...
Creating HDL project...
Creating HDL output folder in: C:\Users\Desktop\list_D
Generating HDL of: U0 (adc/listmodule)
Generating HDL of: U1 (timer/counter_rising)
Generating HDL of: U2 (timer/scaler)
Generating HDL of: U3 (misc/clock)
Generating HDL of: U4 (misc/int)
Generating HDL of: U5 (misc/binary)
Generating HDL of: U6 (register_file/regw)
Memory Mapped Component U0 on address: 0x80000000
Memory Mapped Component Mapping register ListReadout_0_STATUS On address: 0x80000001
Memory Mapped Component Mapping register ListReadout_0_CONFIG On address: 0x80000002
Assembling HDL of top page...
Generating C library...
Generating project description library...
Design fully generated
Starting HDL Compiler...

***** Vivado v2016.3 (64-bit)
**** SW Build 1482963 on Mon Oct 10 19:07:27 MDT 2016
**** IP Build 1481267 on Mon Oct 10 21:28:31 MDT 2016
** Copyright 1986-2016 Xilinx, Inc. All Rights Reserved.

source C:/Users/Desktop/list_D/HDL/list_D.tcl
# set outputDir C:/Users/Desktop/list_D/output/list_D
# file mkdir $outputDir
# create_project -force list_D C:/Users/Desktop/list_D/output/list_D -part XC7K160TFFG676-2
# set_property target_language VHDL [current_project]
# set_property XPM_LIBRARIES {XPM_CDC XPM_MEMORY XPM_FIFO} [current_project]
# add_files pcores/FTDI_FIFOs.xci
INFO: [IP_Flow 19-234] Refreshing IP repositories
INFO: [IP_Flow 19-1704] No user IP repositories specified
INFO: [IP_Flow 19-2313] Loaded Vivado IP repository 'C:/Xilinx/Vivado/2016.3/data/ip'.
WARNING: [IP_Flow 19-2162] IP 'FTDI_FIFOs' is locked:
* IP 'FTDI_FIFOs' shares a common output directory with other IP. It is recommended that each IP be placed in its own directory.
Please select 'Report IP Status' from the 'Tools/Report' menu or run Tcl command 'report_ip_status' for more information.
# add_files pcores/system_clock_generator.xci
WARNING: [IP_Flow 19-2162] IP 'system_clock_generator' is locked:
* IP 'system_clock_generator' shares a common output directory with other IP. It is recommended that each IP be placed in its own directory.
Please select 'Report IP Status' from the 'Tools/Report' menu or run Tcl command 'report_ip_status' for more information.
# add_files C:/Users/Desktop/list_D/HDL/top_list_D.vhd
# add_files pcores/counter_rising.vhd
# add_files pcores/counter_rising.vhd
```

Fig. 7.8. Il "Compiler Output" durante la compilazione del firmware [14].

7.2.2) Programmazione dell'FPGA

Una volta che il processo di compilazione del firmware è terminato con successo e il file bitstream è stato generato, l'utente può programmare l'FPGA. La scheda DT5550W deve essere alimentata e collegata al computer tramite il suo cavo programmatore specifico micro-USB; questo nel caso si voglia effettuare un aggiornamento del firmware veloce e volatile. È possibile fare clic sul pulsante "Program FPGA" nel toolbar "Home" per avviare il processo di programmazione FPGA. Il "Compiler Output" mostra automaticamente l'output di Vivado / Quartus reindirizzato. Per la DT5550W, SCI-Compiler genera un file .tcl nella cartella "HDL" del progetto corrente, per eseguire dalla shell il software Vivado. In questo file viene specificato il file .bit che deve essere utilizzato per programmare l'FPGA, che viene automaticamente preso dalla cartella "output" del progetto attualmente aperto in SCI-Compiler. Vivado programma l'FPGA con il file bitstream e, quando il processo è terminato, se non si è verificato alcun errore, il messaggio "Target device programmed successfully!" viene visualizzato, come si può vedere in Fig. 7.9. In caso contrario, il processo viene interrotto e viene segnalato un messaggio di errore in rosso per notificare all'utente il problema che si è verificato.

7.2.3) Come testare il firmware: Resource Explorer

SCI-Compiler offre la possibilità di connettersi con la scheda e di esplorare le caratteristiche del firmware FPGA che è stato graficamente progettato dall'utente, compilato e caricato sull'FPGA.

Lo strumento che consente il test del firmware FPGA è il "Resource Explorer", che può essere avviato facendo clic sul pulsante "Resource Explorer" nel toolbar "Home". Nella finestra "Connection" che appare (Fig. 7.10), l'utente può configurare la connessione alla scheda desiderata.



```
Start programming target device

***** Vivado v2016.3 (64-bit)
**** SW Build 1682563 on Mon Oct 10 19:07:27 MDT 2016
**** IP Build 1681267 on Mon Oct 10 21:28:31 MDT 2016
** Copyright 1986-2016 Xilinx, Inc. All Rights Reserved.

source C:/Users/Documents/Projects/list_D/HDL/list_D_program.tcl
# open_hw
# connect_hw_server
INFO: [Labtools 27-2285] Connecting to hw_server url TCP:localhost:3121
INFO: [Labtools 27-2222] Launching hw_server...
INFO: [Labtools 27-2221] Launch Output:

***** Xilinx hw_server v2016.3
**** Build date : Oct 10 2016-19:47:06
** Copyright 1986-2016 Xilinx, Inc. All Rights Reserved.

# open_hw_target
INFO: [Labtoolstcl 44-466] Opening hw_target localhost:3121/xilinx_tcf/Digilent/210205910013A
# set_property PROGES.FILE {} [lindex [get_hw_devices xc7k160t_0] 0]
# set_property PROGRAM.FILE {C:/Users/Documents/Projects/list_D/output/list_D/list_D_runs/impl_1/TOP_list_D.bit} [lindex [get_hw_devices xc7k160t_0] 0]
# program_hw_devices [lindex [get_hw_devices xc7k160t_0] 0]
INFO: [Labtools 27-3164] End of startup status: HIGH
program_hw_devices: Time (s): cpu = 00:00:06 ; elapsed = 00:00:06 . Memory (MB): peak = 235.926 ; gain = 0.324
# refresh_hw_device [lindex [get_hw_devices xc7k160t_0] 0]
INFO: [Labtools 27-1434] Device xc7k160t (JTAG device index = 0) is programmed with a design that has no supported debug core(s) in it.
# exit
INFO: [Common 17-206] Exiting Vivado at Wed Feb 21 13:19:30 2018...
Target device programmed successfully!
```

Fig.7.9. Il "Compiler Output" dopo aver programmato con successo l'FPGA [14].



Fig.7.10. La finestra "Connection" del "Resource Explorer" [14].

In particolare, è possibile scegliere il modello della scheda cliccando su una delle icone che mostrano una lista delle schede supportate da SCI-Compiler, selezionare una delle opzioni della lista a tendina "Connection Type" tra USB, Ethernet o VME e inserire nel campo "IP / Serial Number" l'indirizzo del protocollo internet della scheda o il numero di serie (a seconda del tipo di connessione scelto). Il firmware caricato sulla scheda può essere testato selezionando il file .json che si trova nella cartella "library" del progetto SCI-Compiler e generato durante il processo di compilazione. L'utente può selezionare il file facendo clic sul pulsante "Browse..." e cercare tra i file del computer. Il campo "Select Json File" mostrerà il percorso del file selezionato. Il pulsante "Connect" abilita la connessione alla scheda con tutte le impostazioni specificate.

A questo punto apparirà la finestra rappresentata in Fig. 7.11, che mostra l'elenco di tutti i Registri, Analizzatori Logici, Oscilloscopi, ecc. disponibili sul firmware FPGA della scheda collegata, con i quali è possibile interagire. Nell'esempio in Fig. 7.11 vengono elencati tutti gli elementi disponibili per il firmware considerato.

7.2.4) Aggiornamento permanente del firmware

Quando si utilizza il pulsante "Program FPGA" e il cavo programmatore micro-USB, il firmware viene scritto nella memoria FPGA volatile. Per aggiornare il firmware dell'FPGA permanentemente nella DT5550W, è possibile generare un file flash e quindi caricarlo sull'FPGA con il Flash Program Tool. Il pulsante "Generate Flash File" genera un file del firmware .bin, compatibile con l'FPGA Xilinx. Il pulsante "Flash Program Tool" avvia l'aggiornamento del firmware. Questi due

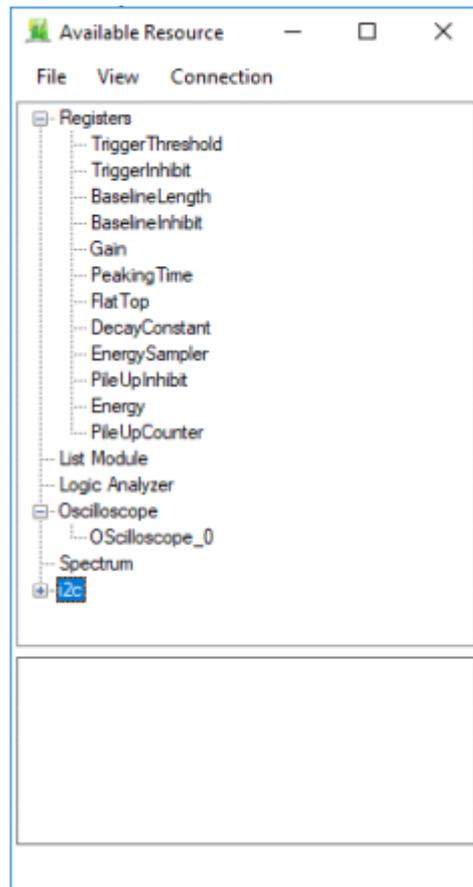


Fig.7.11. Esempio di finestra "Available Resource" del "Resource Explorer" [14].

pulsanti sono visibili in Fig. 7.12 e fanno parte del sottogruppo "Flash Tool" del tool "Home", come visto nel Par. 7.1.1. La procedura per eseguire l'aggiornamento permanente del firmware è descritta di seguito:

- Generare un file Flash .bin, che contiene il codice del firmware, attraverso il pulsante "Generate Flash File".
- Portare il jumper Bootloader, visto nel Cap. 6.1, in posizione 1-2 per avviare l'FPGA in modalità "Bootloader"; il tutto con la scheda spenta.
- Collegare la scheda al PC tramite la porta di comunicazione USB 3.0 e accendere la scheda.
- Premere il pulsante "Flash Program Tool".
- Vi sarà l'apparizione di una procedura guidata nel processo di aggiornamento del firmware. Tutto l'hardware supportato collegato tramite USB al computer verrà elencato nella casella combinata. Si seleziona l'hardware su cui si desidera caricare il firmware e poi il pulsante "Identify" che farà lampeggiare un LED sulla scheda selezionata.

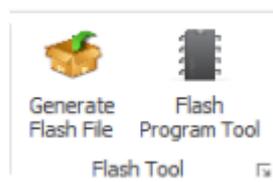


Fig.7.12. Il sottogruppo "Flash Tool" del toolbar "Home" [14].

- Si carica il file .bin, contenente il codice del firmware e si preme il pulsante “Start”.
- Quando il firmware è completamente caricato un messaggio invita l'utente a spegnere e riaccendere la scheda per caricare il nuovo firmware.
- Reinserire il jumper Bootloader sulla modalità di normale funzionamento, in posizione 2-3, per consentire alla scheda di funzionare in modalità standard e caricare il firmware dalla memoria flash.

Programmazione della scheda DT5550W / A55CIT4

Il presente capitolo è dedicato all'analisi e alla spiegazione di tutte le fasi di progettazione del firmware, con il quale programmare l'FPGA della DT5550W, e all'esposizione precisa della struttura dell'acquisizione dei dati, che deve avere un'organizzazione rigorosa ed ordinata. Come è stato spiegato nel Cap.5, il rivelatore consiste di due layer di fibre scintillanti, uno orizzontale e uno verticale, per un totale di canali di lettura pari a 256, ossia 128 per layer. Per questo motivo, sono necessarie due schede DT5550W / A55CIT4, in quanto ognuna possiede due matrici di SiPM per un totale di 128 celle, ossia 128 canali di lettura portati nei quattro Citiroc, i quali estraggono le informazioni provenienti dai sensori. Tuttavia, nel corso della trattazione, si farà sempre riferimento ad una sola scheda, sulla quale il programma è stato sviluppato, dato che, una volta che questo è implementato per una scheda, è applicato anche all'altra; quindi, nella spiegazione della struttura dell'acquisizione dati si considereranno sempre 128 canali di lettura.

L'acquisizione dei 128 canali dei Citiroc è stata impostata in maniera tale da avere una logica molto simile a quella del DDS. Nel Cap.4 sono stati introdotti i chip Tera per le camere di ionizzazione: avevano il compito di leggere in parallelo la corrente proveniente dagli anodi segmentati delle diverse camere, convertirla in conteggi delle particelle passanti nella camera a gas e contare il numero di particelle con un contatore. L'organizzazione dei chip Tera è stata riassunta nelle Fig. 4.7 e 4.10, rispettivamente per il Tera06 e Tera07. Questi hanno alcune differenze tecnologiche tra loro ma, a livello di funzionalità, sono praticamente identici. Nel monitor per basse intensità è stata

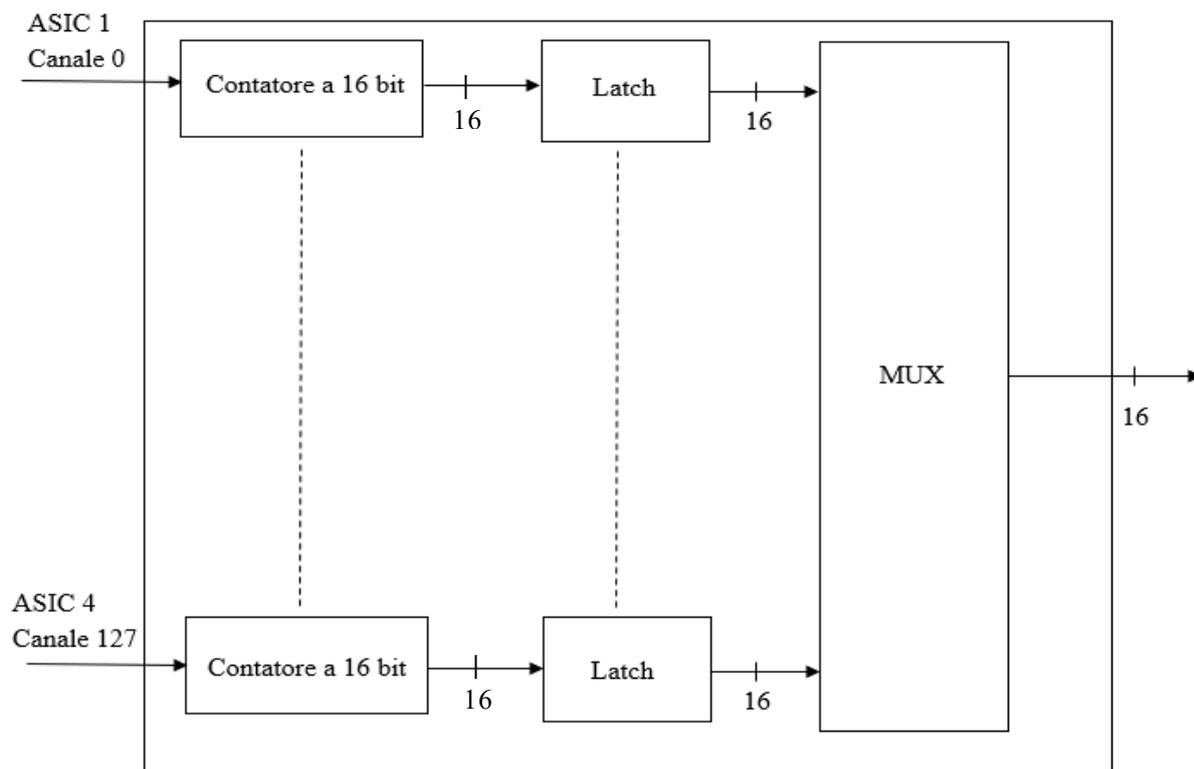


Fig.8.1. Struttura dei canali di acquisizione dati.

utilizzata un'architettura di acquisizione dati molto simile a quella dei chip Tera con alcune differenze. In particolare, ciò che faceva il chip Tera a livello hardware è stato implementato a livello firmware (e quindi a livello di FPGA) nella DT5550W / A55CIT4. In Fig. 8.1 è illustrato schematicamente come sono organizzati i 128 canali di acquisizione. È possibile notare una forte somiglianza con l'architettura del chip Tera. La differenza principale è la mancanza del convertitore corrente-frequenza. Nel chip Tera, il convertitore corrente-frequenza aveva il compito di convertire la corrente di ogni canale in numero di conteggi; quindi, tanto più era il numero di particelle passanti nella camera, maggiore era l'intensità della corrente che arrivava ai canali e maggiori erano i conteggi. Nel monitor per basse intensità questo non è necessario: l'insieme di fibre scintillanti e SiPM rilevano il numero di particelle che attraversano il monitor con una sensibilità molto maggiore rispetto al convertitore corrente-frequenza e i Citiroc leggono ed elaborano queste informazioni che possono essere prelevate a livello firmware. Il software SCI-Compiler permette di collegare i vari canali dei Citiroc a dei contatori a 16 bit, come illustrato in Fig. 8.1, in modo che possano contare le transizioni presenti sui canali dei Citiroc, rappresentanti il passaggio delle particelle. I 128 canali vengono acquisiti indirizzandoli attraverso un Multiplexer. Così come per i Tera, anche se l'acquisizione viene effettuata canale per canale, vi è un'operazione di latch (tra il contatore e il canale del MUX), che blocca contemporaneamente l'uscita di tutti i contatori, per cui i dati acquisiti fanno riferimento tutti allo stesso istante.

Prima di passare ad illustrare dettagliatamente il firmware e a comprendere le varie tappe che hanno portato al suo completamento, occorre una panoramica dell'intero sistema di acquisizione dati del monitor per fasci a bassa intensità. La Fig. 8.2 mostra l'intero flusso dei comandi e dei dati relativi all'acquisizione dei 128 canali dei Citiroc. È possibile osservare come l'intera acquisizione sia governata dal DDS attraverso i comandi Latch, ResetD e StrobeOut. Il comando di Latch, oltre che essere un segnale necessario per effettuare l'operazione di latch nominata poc'anzi, è anche da considerare come un segnale di Start, ossia un segnale che fa partire l'acquisizione dei 128 canali. Il Latch, quindi, impone la frequenza alla quale viene fatta l'acquisizione; è richiesta come specifica che il segnale di Latch abbia un periodo di 100 μ s, rimanendo a livello logico alto per 100 ns. In questo modo, la frequenza di lettura dei dati è di 10 kHz. Il secondo comando inviato dal DDS è il ResetD, ossia un impulso, il cui compito consiste nell'azzerare i contatori visti in Fig. 8.1, ovvero azzerare i conteggi delle transizioni rilevate dai SiPM. Si tratta di un comando asincrono che può essere mandato in qualsiasi momento dell'acquisizione. Lo StrobeOut è l'ultimo comando del DDS, mandato sincrono con il Latch. In generale, il segnale di strobe viene mandato sincrono al comando di Latch ogni volta che si vogliono leggere dei dati. Lo StrobeOut è un segnale con periodo di 100 μ s, che rimane a livello logico alto per 100 ns, mandato dal DDS per leggere i dati. Questo segnale

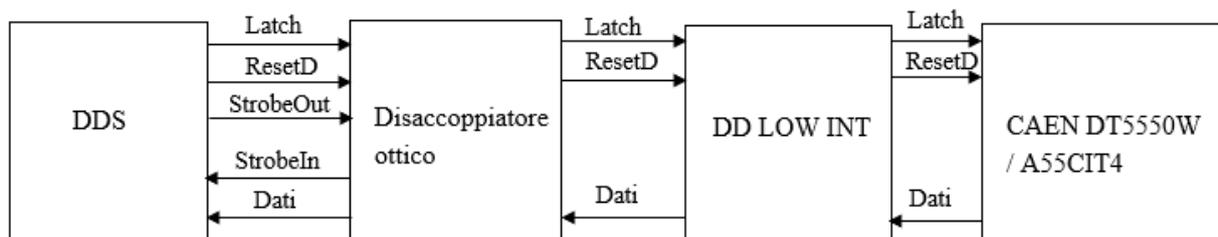


Fig.8.2. Schema a blocchi dell'intero sistema di acquisizione dati.

entra nel Disaccoppiatore Ottico; quest'ultimo è una scheda elettronica, già esistente e usata per gli altri monitor del CNAO, con il compito di ricevere lo StrobeOut e mandare indietro al DDS il segnale di StrobeIn, il quale ha la stessa forma del segnale di StrobeOut e segnala al DDS l'arrivo dei dati. L'utilità di questi segnali è gestire il ritardo tra l'invio del segnale di Latch e la ricezione dei dati dovuto alla distanza spaziale, che potrebbe non essere costante, tra il DDS e il monitor. Si ricorda, infatti, come detto nel Cap. 4.3, che il DDS e le BOX non si trovano nella stessa stanza. I segnali di Latch e ResetD passano, successivamente, attraverso la scheda DD Low Int, che è stata progettata, assemblata e testata appositamente per il sistema di rilevazione per basse intensità e di cui si parlerà dettagliatamente nel capitolo successivo. In tal modo, i due segnali arrivano alla scheda CAEN e vengono trattati a livello firmware, così da restituire in uscita i dati che, attraverso la DD Low Int, passano al Disaccoppiatore Ottico e, in sincronia col segnale di StrobeIn, arrivano al DDS. L'acquisizione dei dati può essere visualizzata tramite un'interfaccia che dialoga con un'FPGA del DDS, che svolge la funzione di lettura dei dati. Il programma di acquisizione utilizzato nel DDS è LabVIEW. Come è stato anticipato, si ha una specifica sulla frequenza dell'acquisizione dati: tutti i 128 canali devono essere letti a una frequenza di 10 kHz e, in particolare, ogni canale deve essere letto ogni 200 ns. I singoli canali sono rappresentati da una parola binaria a 16 bit che riporta un valore numerico, il quale simboleggia il numero di conteggi dei contatori dei canali di acquisizione, ovvero il conteggio del numero di particelle che hanno attraversato il monitor.

Nei prossimi paragrafi verrà spiegato passo dopo passo il codice a blocchi implementato su SCI-Compiler che costituisce il firmware per l'acquisizione. Il codice è stato suddiviso in vari subdiagram, ognuno dei quali svolge una precisa funzione. I subdiagram vengono poi richiamati nel diagramma principale (top), come se fossero delle vere e proprie funzioni chiamate durante la stesura di un programma di un qualunque linguaggio di programmazione. Verrà fatta un'analisi dettagliata di ogni subdiagram fino ad arrivare a quella del top. In Fig. 8.3 è rappresentato un diagramma a blocchi utile per spiegare l'intera logica del firmware. Ognuno dei singoli blocchi sarà

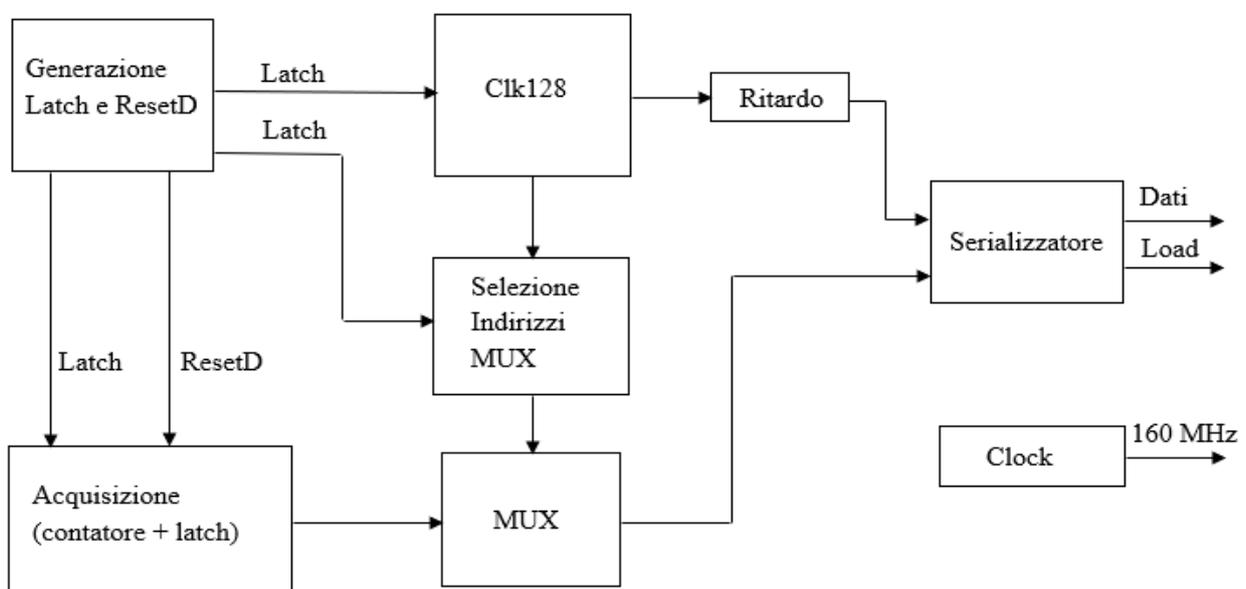


Fig.8.3. Schema a blocchi del funzionamento del firmware.

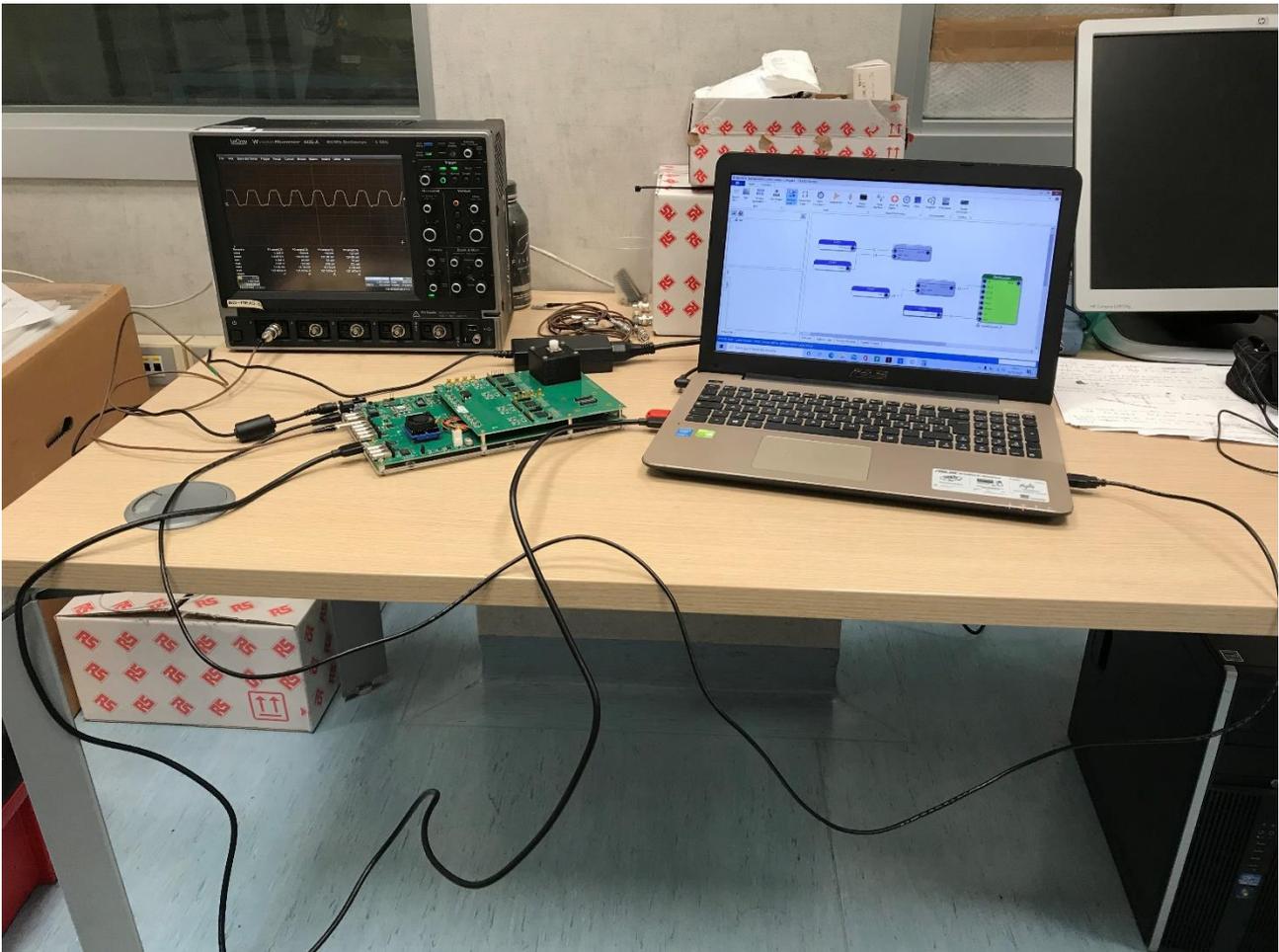


Fig.8.4. Collegamento PC con scheda e visualizzazione segnale su oscilloscopio.

analizzato dettagliatamente nei prossimi paragrafi.

A destra della suddetta figura, è possibile vedere tre frecce (Dati, Load e un clock da 160 MHz), che rappresentano tre segnali portati in uscita dalla DT5550W / A55CIT4, attraverso i connettori LEMO programmabili in I / O di cui dispone la scheda, alla scheda DD Low Int. Tutti i vari diagrammi, provati per arrivare all'implementazione del codice finale, sono stati testati osservando su un oscilloscopio i segnali di interessi presentati in uscita sui connettori LEMO, come in Fig. 8.4, dove è possibile vedere il PC sul quale è stato creato un programma SCI-Compiler. Grazie a quest'ultimo, si è programmata l'FPGA della scheda DT5550W / A55CIT4, in figura collegata al PC attraverso il cavo USB 3.0 e il cavo micro-USB. Si può osservare, quindi, un cavo LEMO con il quale viene visualizzato sull'oscilloscopio un determinato segnale mandato in uscita su uno degli otto LEMO della scheda.

8.1) Generazione Latch e ResetD

Il Latch e il ResetD sono due comandi che, come anticipato, provengono dal DDS. Tuttavia, sia per costruire il codice SCI-Compiler sia per testarlo, è necessario simulare questi comandi, rendendoli uguali a quelli che arriverebbero dal DDS. Questo viene fatto nel subdiagramma denominato "Generazione Latch". Il diagramma a blocchi è visibile in Fig. 8.5. Il blocco principale è il Pulse Generator (Fig. 8.6), che permette la generazione di un'onda periodica dandogli degli input relativi

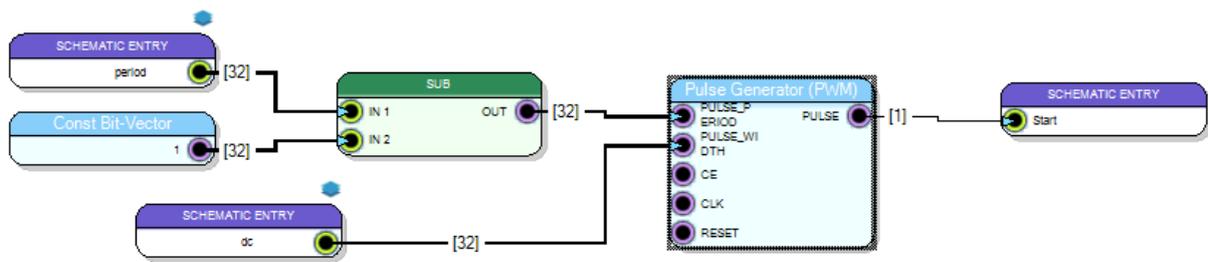


Fig.8.5. Subdiagram "Generazione Latch".

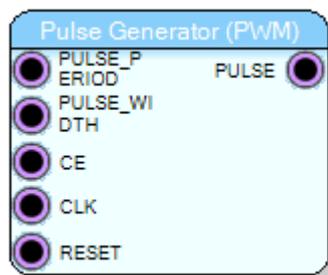


Fig.8.6. Pulse Generator.

alle caratteristiche dell'onda. L'input "PULSE_PERIOD" e "PULSE_WIDTH" indicano rispettivamente il periodo dell'impulso e la sua larghezza, ossia il tempo in cui si trova a livello logico alto sull'intero periodo. Il valore di questi input deve essere dato in termini di cicli di clock e l'input deve essere una parola a 32 bit, come segnalato sull'help del Pulse Generator. L'input "CE" abilita la generazione dell'output quando vede in ingresso un 1 logico; in caso contrario il segnale non viene generato. L'input "CLK" indica il clock interno del blocco. Molti blocchi hanno un proprio clock interno su SCI-Compiler; in questo caso, infatti, i valori di cicli di clock inseriti nei primi due input del blocco fanno riferimento proprio a questo clock. Infine, l'input "RESET" serve per resettare il processo di generazione dell'onda in uscita, che avviene quando quel pin vede in ingresso un 1 logico.

Nella Fig. 8.5 si notano tre blocchetti denominati "SCHEMATIC ENTRY". Questi blocchi sono utilizzati per trasferire segnali dal top a un subdiagram (Schematic Port In), o da un subdiagram al top (Schematic Port Out). Nel diagramma ci sono due Schematic Port In ("period" e "dc") e una Schematic Port Out ("Start"). Il segnale "period" viene mandato nel blocco SUB che sottrae il primo input ("period") e il secondo (in questo caso 1). A breve verrà spiegato il motivo della presenza di questo blocco. Gli ultimi tre pin del Pulse Generator non hanno alcun blocco collegato. Questo significa che in quei pin viene dato in ingresso un valore di default come suggerito dall'help:

- In CE il valore di default è 1; la generazione dell'output è sempre abilitata;
- In CLK viene dato il clock di default che è quello a 160 MHz;
- In RESET il valore di default è lo 0 logico; il blocco, quindi, non viene resettato mai.

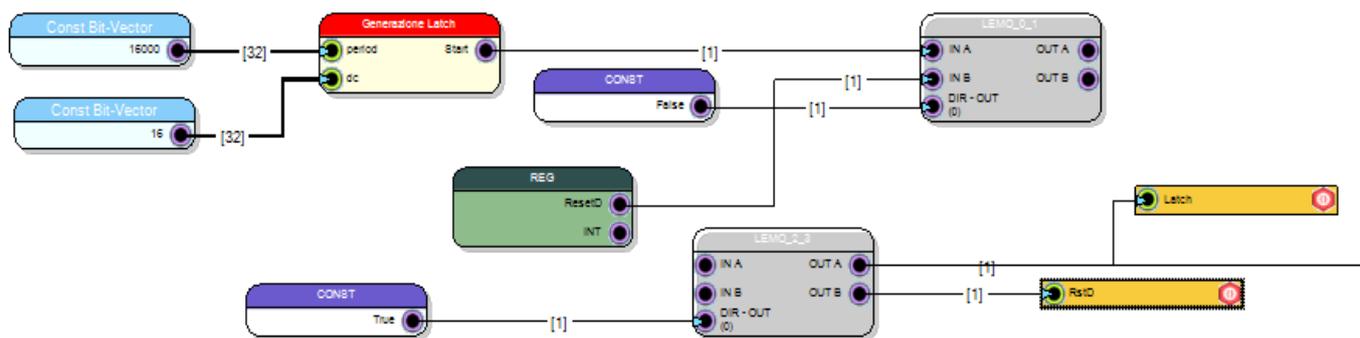


Fig.8.7. Latch e ResetD (parte di diagramma sul top).

Il subdiagram viene richiamato sul top. La parte di diagramma sul top relativa alla generazione di Latch e ResetD è quella in Fig. 8.7. Il blocco rosso, denominato “Generazione Latch”, sarebbe il diagramma rappresentato in Fig. 8.5, richiamato sul top, come si può notare dai due ingressi “period” e “dc” e dall’uscita “Start”. A “period” si dà il valore di 16000. Questo perché il segnale di Latch, da specifica, deve avere un periodo di 100 μ s. Considerando che quel valore indica 16000 cicli di un clock da 160 MHz, si può intendere che questo equivale a dare all’onda da generare un periodo di 100 μ s. Effettuando lo stesso ragionamento, dando il valore di 16 a “dc”, si impone che l’onda deve rimanere a livello logico alto per 100 ns, come da specifica. Tuttavia, considerando che al valore 16000 di period gli viene sottratto 1 con il blocco sottrattore SUB, si dà in ingresso al pin “PULSE_PERIOD” del Pulse Generator il valore 15999. Questa operazione è stata necessaria perché è stato possibile osservare sperimentalmente tramite l’oscilloscopio che, dando 16000 come valore di periodo, si otteneva un impulso con un periodo di 100 μ s più un ciclo di clock, che, per un clock da 160 MHz, equivale a circa 6,25 ns. Facendo quella sottrazione, invece, l’onda ha un periodo precisamente di 100 μ s. Questo vale in generale nell’utilizzo del Pulse Generator; anche più avanti nella spiegazione del codice, si troverà sempre un valore di ingresso a “PULSE_PERIOD” pari al valore desiderato meno un ciclo di clock. Questo discorso, invece, non vale per il pin “PULSE_WIDTH”, in questo caso pari a 16 che, come è stato osservato dall’oscilloscopio, genera un impulso di larghezza pari esattamente a 100 ns.

Per poter avere il Latch e il ResetD in ingresso alla scheda come se provenissero dal DDS, è stata adottata la seguente tecnica: i due segnali vengono mandati in output su due LEMO differenti (programmati come output), portati verso l’oscilloscopio con due cavi LEMO, dove è possibile visualizzarli e, con un connettore a T, vengono riportati indietro, verso altri due LEMO (programmati come input). In Fig. 8.8 è possibile osservare questa configurazione: nel caso in figura è riportata la configurazione per un solo segnale. Per quanto riguarda il firmware, questo meccanismo è stato implementato nel seguente modo (riportato sempre in Fig. 8.7): sono stati selezionati i blocchi di SCI-Compiler che si riferiscono ai LEMO, i quali possono essere scelti a coppie di due. In Fig. 8.7, infatti, si vedono le coppie di LEMO 0 e 1 e di LEMO 2 e 3. I LEMO 0 e 1 sono stati programmati in uscita, come si può notare dal collegamento del blocco “False” (che indica uno 0 logico) al pin “DIR-OUT (0)”, il quale imposta quei due LEMO come output. In “IN A” è stato collegato il segnale “Start” (ossia il Latch), in “IN B” è stato collegato il segnale di ResetD. Questo è stato implementato come un registro in lettura (blocco verde). Utilizzando il

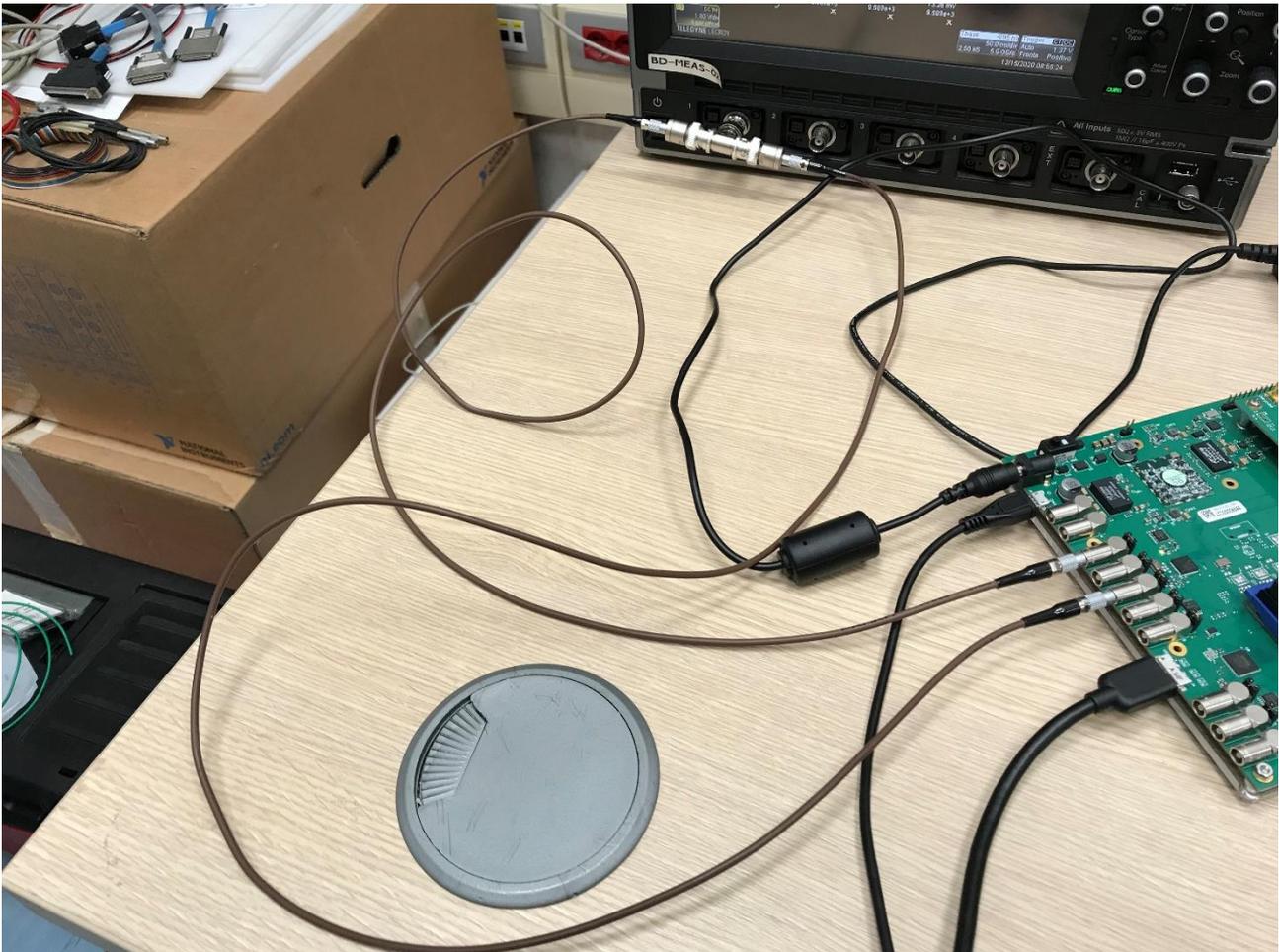


Fig.8.8. Configurazione di connettori e cavi per mandare in uscita un segnale da un LEMO e riportarlo indietro in ingresso su un altro LEMO.

registro, l'utente può scrivere un qualsiasi valore da tastiera che verrà letto sul pin al quale il registro si collega. In questo caso al registro è stata data la possibilità di scrivere in quel pin due bit, ossia 0 o 1. Quando si ha 0 il ResetD non ha alcun effetto sul codice. Quando si fa leggere sul pin del LEMO l'1 logico, allora avviene il reset dei contatori dei 128 canali di acquisizione. È possibile, così, interagire con il programma in esecuzione sulla scheda grazie al pulsante "Resource Explorer" presente su SCI-Compiler, di cui si è parlato già nel Par. 7.2.3. In questo modo, si possono resettare in qualsiasi momento i contatori, rispettando la specifica che vuole il segnale di ResetD essere un comando asincrono. Infine, si ha il blocco LEMO 2 e 3, che viene programmato come input, grazie al blocco "True" (che indica il livello logico alto) collegato al pin "DIR-OUT (0)", il quale imposta i due LEMO come input. Questi LEMO ricevono in ingresso i segnali che prima sono stati mandati in uscita dai LEMO 0 e 1. Una volta portati in ingresso, i due segnali possono essere usati all'interno del firmware e collegati ad altri blocchi a partire dai pin "OUT A" e "OUT B" del blocco LEMO 2 e 3. I rettangoli gialli a cui si collegano i fili da "OUT A" e "OUT B" sono dei blocchi denominati "Variable Read". Questi servono per leggere e memorizzare il valore del pin del blocco al quale è collegato. Esisterà nel programma un blocco graficamente identico che si chiama "Variable Write", il quale serve per scrivere quello stesso valore su un pin di un altro blocco al quale il "Variable Write" si collega. "Variable Read" e "Variable Write" sono usati insieme per evitare di collegare due pin di due blocchi diversi utilizzando un filo, nel caso in cui i due blocchi da collegare siano molto lontani nel diagramma. In questo caso, questa tecnica è stata utilizzata con

il Latch e il ResetD che dovranno essere collegati a vari pin nel diagramma top, in maniera tale da far risultare il diagramma più ordinato avendo meno fili.

In conclusione, il segnale di Latch viene portato fuori sul LEMO 0 e ricevuto in ingresso sul LEMO 2 mentre il segnale di ResetD viene portato fuori sul LEMO 1 e ricevuto in ingresso sul LEMO 3.

8.2) Clk128

Il subdiagram “Clk128” è una parte del codice che ha il compito di generare 128 impulsi ad una frequenza di 5 MHz, ossia un impulso ogni 200 ns. Si ricorda che 200 ns è il tempo entro il quale i dati a 16 bit di ogni singolo canale, relativi alle transizioni rilevate dai Citiroc, devono essere acquisiti. Questi 128 impulsi sono mandati ogni volta che si ha il fronte di discesa del segnale di Latch, quindi ogni 100 μs. Gli impulsi sono mandati, all’interno del firmware, nel subdiagram “SelIndirizzi” e nel subdiagram “Serializzatore”. Nel primo caso servono per la generazione degli indirizzi di selezione dei canali del MUX mentre nel secondo caso servono al processo di serializzazione dei 16 bit di dati.

In Fig. 8.9 viene mostrato il subdiagram “Clk128”. In alto a sinistra, si vede un Pulse Generator che genera un impulso con periodo di 200 ns. L’ingresso al pin “PULSE_PERIOD”, infatti, è pari a 31, ossia 32 cicli di clock che, con un clock a 160 MHz, danno un segnale periodico di 200 ns, meno un ciclo di clock, come spiegato nel paragrafo precedente. Al pin “PULSE_WIDTH” viene dato in ingresso il valore 1, in quanto non è importante quanto tempo il segnale rimanga alto, ma è fondamentale che, ogni 200 ns, ci sia una transizione da livello logico basso a livello logico alto. Sotto il Pulse Generator è presente il blocco “GATE And DELAY” (Fig. 8.10). Questo ha il compito di prendere in ingresso nel pin “In” un segnale, dargli una certa larghezza, attraverso un valore in termini di cicli di clock da collegare nell’input “GATE”, e un certo ritardo, in base al valore in termini di cicli di clock assegnato al pin “DELAY”. In base al valore in “GATE” il segnale in uscita dal blocco sarà più o meno largo rispetto al segnale d’ingresso. Questo blocco è stato utilizzato per precauzione in quanto, nel caso il segnale di Latch dovesse avere dei problemi durante il tragitto di arrivo al LEMO della DT5550W / A55CIT4 quando il sistema sarà collegato

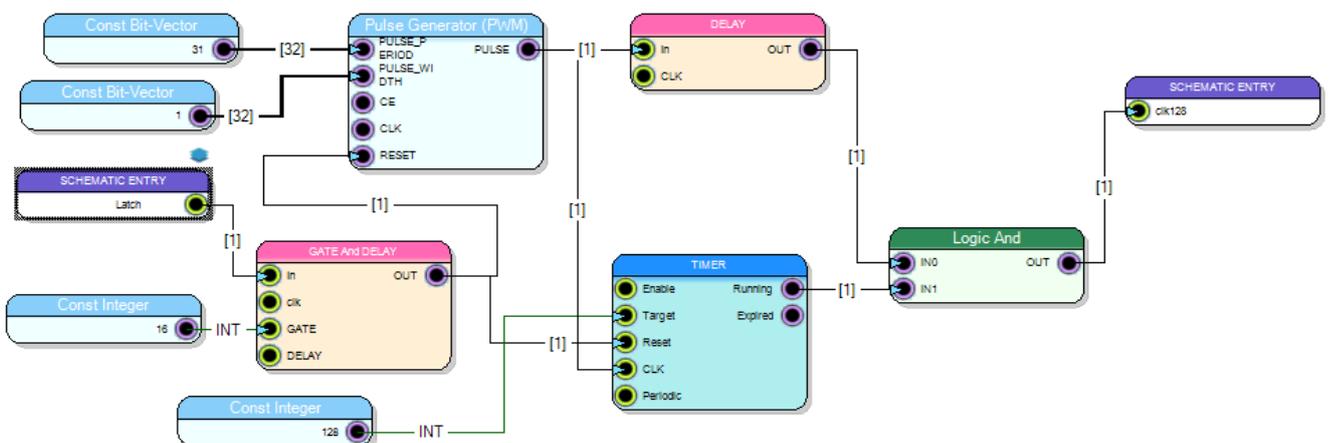


Fig.8.9. Subdiagram “Clk128”.

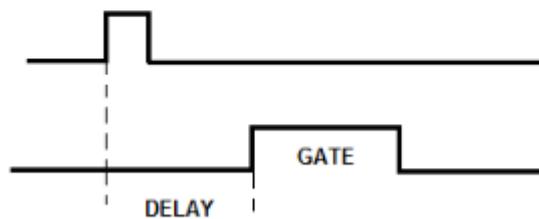
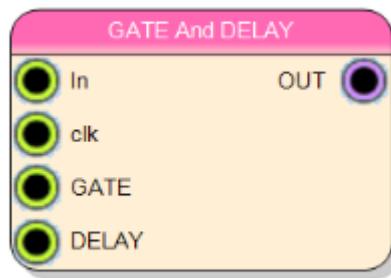


Fig.8.10. Blocco Gate and Delay.

con il DDS, oppure nel caso in cui risultassero degli errori durante la generazione del segnale in termini di larghezza, il segnale di Latch fuoriuscente da quel blocco avrà sicuramente la morfologia desiderata. Nel subdiagram in questione, il segnale di Latch viene collegato al pin “In” del blocco GATE and DELAY. Il pin “clk” rimane privo di collegamento, lasciando, quindi, il valore di default di 160 MHz. Al pin “GATE” viene dato il valore di 16, indicando che il segnale di uscita dovrà rimanere alto per 100 ns. Per quanto riguarda il pin “DELAY”, dato che non si ha interesse nel dare un ritardo al Latch, gli viene assegnato il valore di default pari a 0. Tuttavia, occorre fare una piccola digressione riguardo il pin “DELAY”. Effettuando dei test per capire se, dando un ritardo pari a 0 a quel pin, il segnale di uscita uscisse effettivamente senza ritardo oppure ci fossero dei ritardi fisiologici, si è giunti alla seguente conclusione: il blocco GATE And DELAY, nel momento in cui viene collegato al suo pin “DELAY” il valore 0, assegna un ritardo al segnale di uscita dal blocco, rispetto a quello di ingresso, pari a circa 30 ns. Nella Fig. 8.11 è illustrato ciò che è stato appena detto. In un diagramma di prova è stato mandato un segnale proveniente da un Pulse Generator in ingresso al GATE And DELAY, il cui valore in ingresso al pin “DELAY” era pari a 0. Il segnale di uscita e il segnale di ingresso sono stati mandati in output su due LEMO diversi e visualizzati sull’oscilloscopio. Come si può vedere dalla figura, avendo impostato la griglia orizzontale dello strumento a passi di 10 ns per divisione, il segnale di uscita (rosso) ha il fronte di salita all’incirca 30 ns dopo il fronte di salita del segnale d’ingresso (giallo). Inoltre, bisogna aggiungere che, ponendo 0 in ingresso al pin “GATE”, non si ottiene in uscita un segnale nullo, bensì un segnale largo quanto un ciclo di clock che, per un clock di 160 MHz, risulta essere circa 6,25 ns. Quindi, nel subdiagram “Clk128” si ottiene dall’uscita del GATE AND DELAY un segnale di Latch largo 106,25 ns circa e ritardato rispetto a quello di ingresso di circa 30 ns.



Fig.8.11. Ritardo di 30ns dato dal blocco "Gate and Delay".

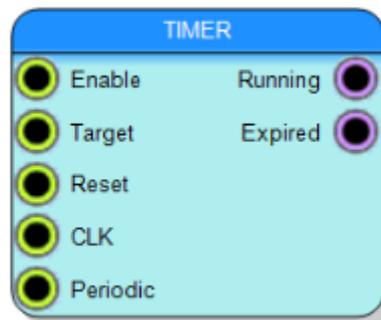


Fig.8.12. Blocco Timer.

Queste differenze dalla condizione ideale, tuttavia, non creano alcun problema all'acquisizione dati. Tornando alla spiegazione del funzionamento del subdiagram, viene di seguito presentato il blocco di maggiore importanza di questa sezione, ossia il Timer (Fig. 8.12). Questo blocco ha la funzione di contare le transizioni di un segnale che viene portato in ingresso al pin "CLK" e confrontarle con il valore collegato in ingresso al pin "Target". Finché il pin "Enable" rimane a livello logico alto, il blocco continua a contare mantenendo il pin di uscita "Running" fisso a livello logico alto. Raggiunto il valore di target, il pin di uscita "Expired", che finora era rimasto basso, va a livello alto. A questo punto, se in ingresso al pin "Periodic" c'è un 1 logico, il numero di conteggi viene automaticamente riaszerato e il blocco ricomincia a contare; se in ingresso a "Periodic" c'è uno 0 logico, il blocco non ricomincia il conteggio e l'uscita "Running" passa da 1 a 0. Il conteggio, inoltre, può essere resettato e riavviato tramite il pin "Reset" che resetta il conteggio quando vede

una transizione da 0 a 1 e ricomincia a contare quando passa da 1 a 0. Dalla Fig. 8.9 si vede che il segnale le cui transizioni vengono contate è quello uscente dal Pulse Generator. Il target impostato è pari a 128. Inoltre, si nota come il segnale di Latch che esce dal GATE And DELAY vada in ingresso sia al pin di reset del Pulse Generator che al pin di reset del Timer. In questo modo le due azioni di generazione della forma d'onda dal Pulse Generator e di conteggio delle sue transizioni da livello logico basso ad alto, effettuato dal Timer, vengono sincronizzate al segnale di Latch e fatte partire nello stesso istante grazie alla funzione di reset. In particolare, le due operazioni di generazione della forma d'onda e relativo conteggio partiranno in corrispondenza del fronte di discesa del Latch. Dopodiché, l'uscita "Running" del Timer, che è sempre pari a 1 durante il conteggio fino al raggiungimento delle 128 transizioni, viene messa in AND con il segnale in uscita del Pulse Generator. In questo modo, vengono generati 128 impulsi distanziati tra loro di 200 ns, i quali partono dopo il fronte di discesa del segnale di Latch. Dopo il 128esimo impulso non ne vengono generati altri, grazie al fatto che il pin "Periodic" è stato lasciato al suo valore di default, ovvero 0. Il conteggio, e quindi la generazione dei 128 impulsi, ricomincerà al successivo impulso di Latch. Inoltre, anche il pin "Enable" è lasciato al suo valore di default di 1.

L'ultimo concetto da spiegare è l'inserimento del blocco "DELAY" tra l'output del Pulse Generator e il primo ingresso della porta logica AND. In un diagramma di prova, per testare il funzionamento della generazione dei 128 impulsi, è stato inserito come target al blocco Timer il valore 8, in modo tale che sarebbe stato più semplice visualizzare sull'oscilloscopio solo 8 impulsi piuttosto che 128. Non inserendo il blocco "DELAY", ciò che è venuto fuori portando l'uscita della porta AND su un

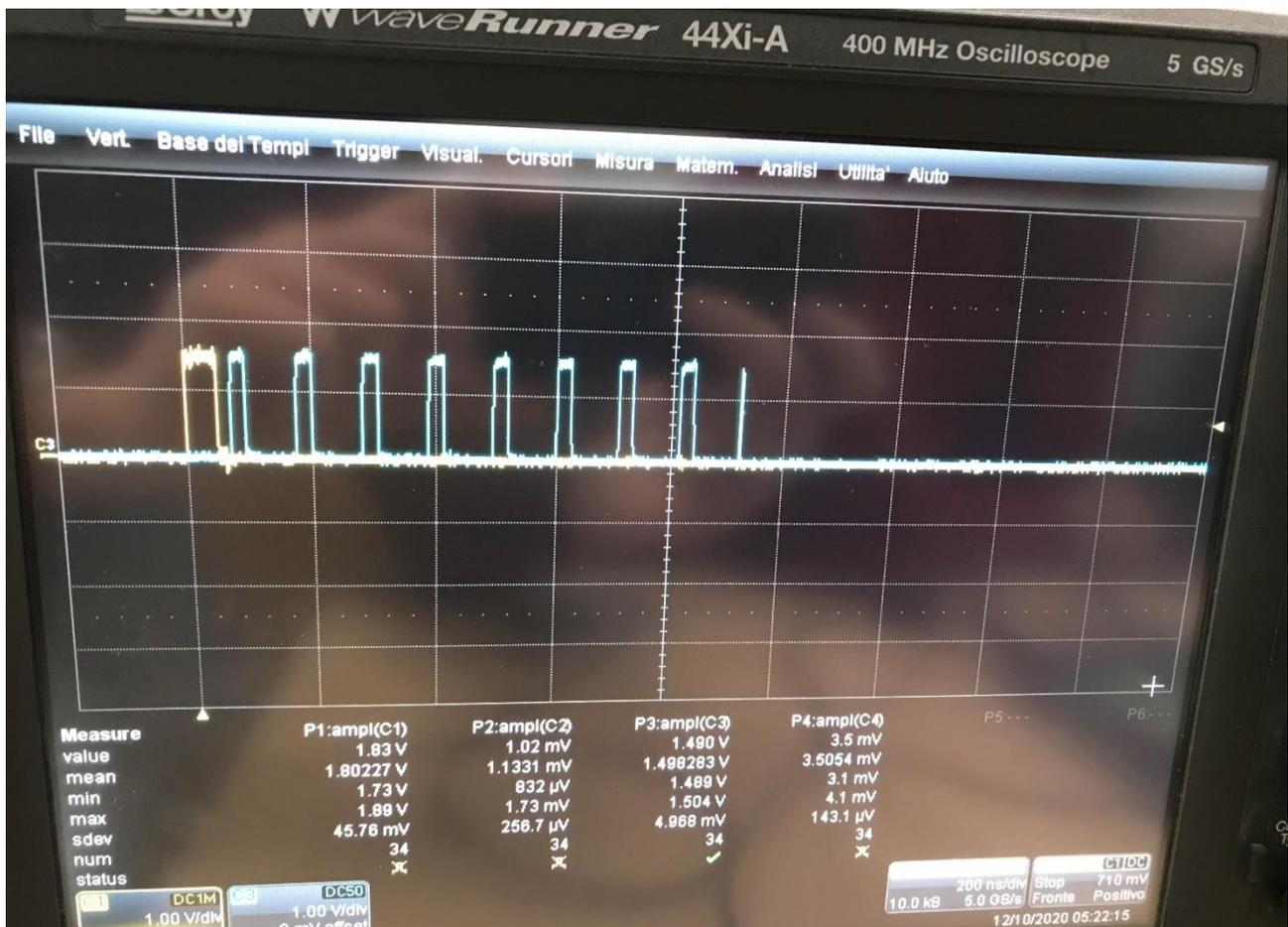


Fig.8.13. Prova di generazione di 8 impulsi.

LEMO è quello che viene mostrato in Fig. 8.13. Dalla figura è possibile notare l'impulso di Latch (giallo) seguito dagli impulsi (azzurri), i quali non sono 8 come ci si aspettava, bensì 9. Il nono impulso risulta essere più corto rispetto a quelli precedenti. Questa situazione è dovuta al fatto che sono stati messi in ingresso alla porta AND due segnali dipendenti dallo stesso blocco (il Pulse Generator), dove, però, uno entra direttamente nella porta logica, mentre l'altro passa attraverso il blocco Timer che ha un suo tempo di lavoro e quindi impone un piccolo fisiologico ritardo al segnale. Si viene a creare la condizione rappresentata in Fig. 8.14. Nella suddetta figura sono rappresentati IN0 e IN1, che sono gli input della porta AND, e OUT, che è l'output della stessa porta. IN1 sarebbe il segnale in uscita dal pin "Running" del Timer, che è sempre alto finché il blocco conta. A causa di ciò che è stato detto prima, si può notare come il segnale di Running si abbassi quando in IN0 è già partito il nono impulso; di conseguenza, l'OUT sarà fatto in quel modo rappresentato in figura e come è stato visto anche sull'oscilloscopio.

Per evitare la formazione dell'ultimo impulso in più è necessario, quindi, inserire un ritardo al segnale in uscita dal Pulse Generator di pochi cicli di clock. Attraverso varie prove sperimentali la scelta è stata quella di inserire un ritardo pari a 7 cicli di clock, considerando sempre un clock di 160 MHz (al pin "CLK" del blocco DELAY viene lasciato il valore di default che sarebbe il clock da 160 MHz). Inserendo questo ritardo si può osservare dalla Fig. 8.15 che viene generato l'esatto numero di impulsi; tuttavia, il primo impulso si è leggermente spostato rispetto al fronte di discesa del Latch. In ogni caso, quest'ultimo effetto non va ad incidere sul corretto funzionamento dell'acquisizione dati.

8.3) Acquisizione

Il subdiagram "Acquisizione" è rappresentato in Fig. 8.16. E' un diagramma che funzionalmente è equivalente allo schema visto in Fig. 8.1. Ciò nonostante, vi sono dei blocchi in più nel diagramma SCI-Compiler che verranno di seguito illustrati e spiegati. Lo scopo di questo subdiagram è contare gli impulsi provenienti da un canale del Citiroc, effettuare un'operazione di latch per memorizzare il valore di uscita del contatore e mandare questo valore in uscita dal subdiagram.

Il primo blocco di cui parlare è il contatore. Questo è rappresentato in Fig. 8.17. E' stato utilizzato, per la precisione, un contatore sincrono sensibile ai fronti di salita (denominato "rising edge" su SCI-Compiler). Questo blocco conta il numero di impulsi che vede nel pin d'ingresso "In", in maniera sincrona con i fronti di salita di un clock. Nel caso del subdiagram implementato si è

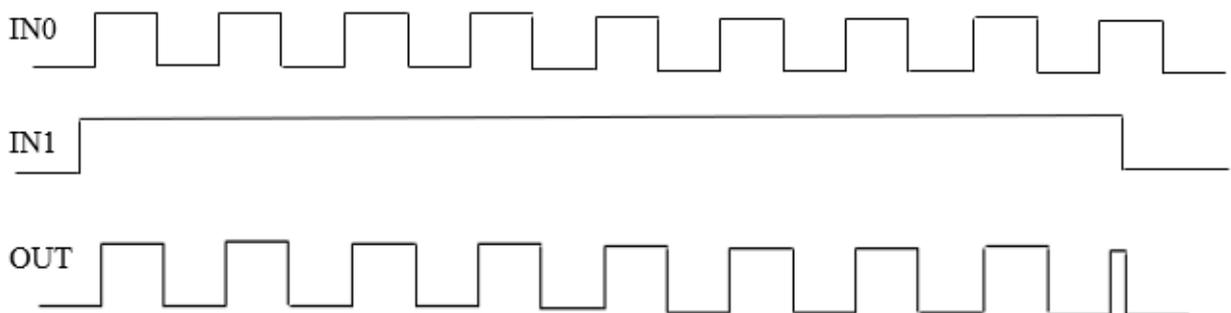


Fig.8.14. I due input (IN0 e IN1) e l'output (OUT) della porta logica AND del subdiagram "Clk128".

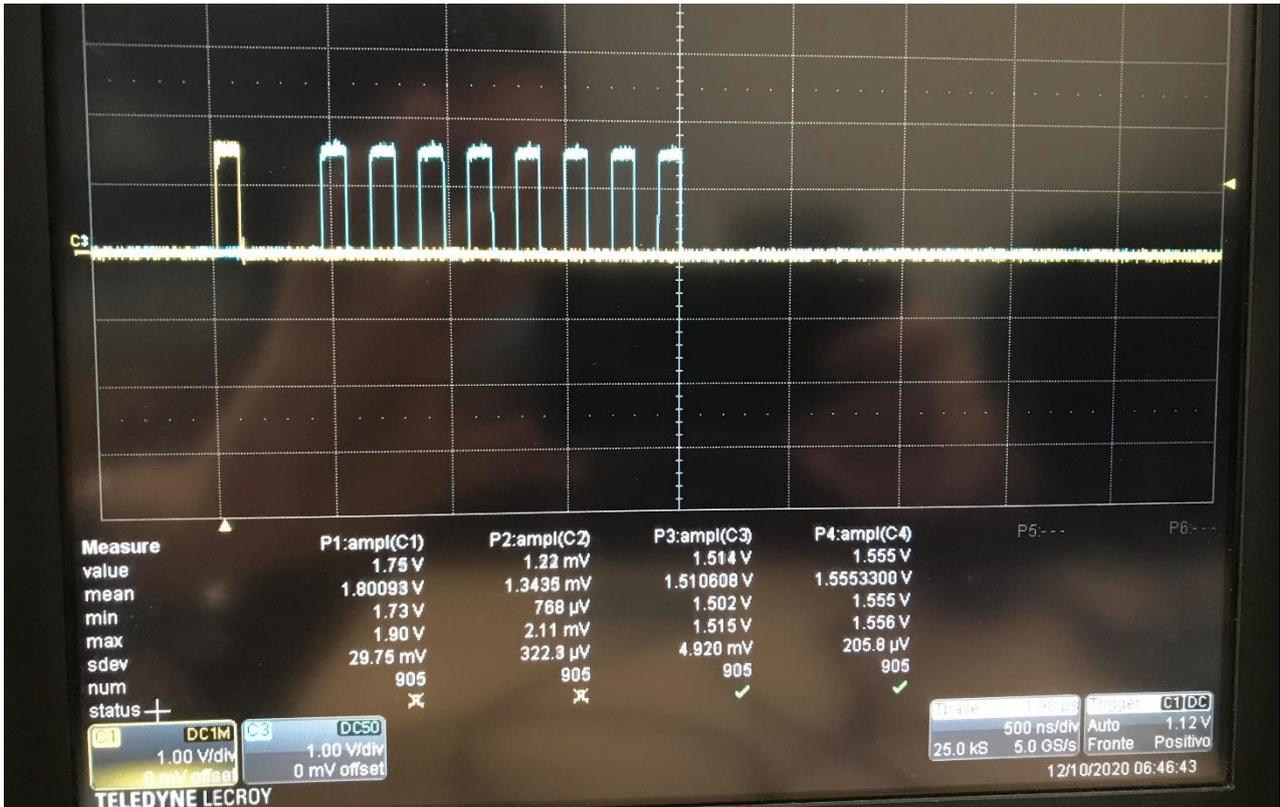


Fig.8.15. Corretta generazione degli impulsi dopo l'inserimento del blocco DELAY.

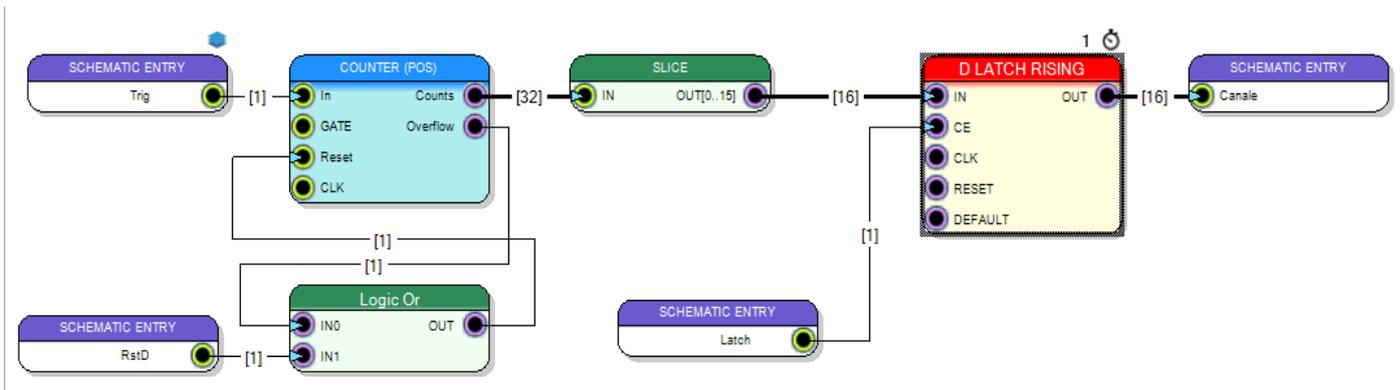


Fig.8.16. Subdiagram "Acquisizione".

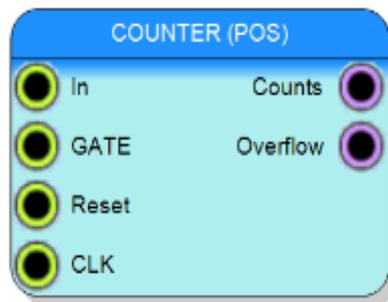


Fig.8.17. Blocco contatore sincrono (rising edge).

lasciato il clock di default (160 MHz). Il conteggio dell'input avviene se il pin "GATE" è lasciato a livello logico alto, il quale è il valore di default di quel pin. Vi è, poi, l'output "Counts" che dà in uscita il valore numerico di impulsi contati. Tutti i contatori su SCI-Compiler sono contatori a 32 bit. Quando il valore del conteggio supera il massimo valore esprimibile con 32 bit, il pin di output "Overflow" genera un impulso. Inoltre, il pin "Reset", quando è a livello logico alto, può essere utilizzato per azzerare l'uscita del contatore e far ripartire il conteggio. Nel subdiagram "Acquisizione" è collegato al pin "In" del contatore una Schematic Port In denominata "Trig". Essa rappresenta un canale del Citiroc. Viene rappresentato in questo modo perché il canale viene collegato all'ingresso del contatore da un altro subdiagram del programma che non è ancora stato introdotto. Dopodiché è possibile vedere il comando di ResetD che va in ingresso ad una porta OR, il cui secondo ingresso è l'overflow del contatore. L'uscita della porta logica viene collegata al pin "Reset" del contatore. In questo modo si è implementato un meccanismo che permette di resettare il conteggio del contatore sia attraverso il comando asincrono ResetD, che grazie all'overflow del contatore che indica il valore massimo esprimibile raggiunto per quel blocco.

Come è già stato detto in precedenza, i dati trattati nella scheda, che descrivono il passaggio delle particelle nel monitor, così come lo era per il Tera06, sono a 16 bit. Per questo motivo, il valore in uscita dal contatore a 32 bit, viene fatto passare attraverso il blocco "SLICE". Questo ha la capacità di estrarre un qualsiasi numero di bit susseguenti da una parola binaria. Nel caso in questione, gli vengono dati in ingresso i 32 bit del contatore per ottenere in uscita i primi 16 (a partire dai meno significativi) e rispettare la specifica relativa al numero di bit dei dati.

I 16 bit sono portati in ingresso in un Latch D. Questo blocco funziona nel seguente modo: riceve un segnale di ingresso e un segnale di clock. Quando il clock è a livello logico basso, l'uscita non cambia e mantiene il valore precedente, indipendentemente dalle variazioni del segnale di ingresso. Quando, invece, il segnale di clock è a livello logico alto, l'ingresso si propaga direttamente attraverso il blocco e viene riportato in uscita. L'ingresso del blocco è il pin "IN", il clock è il pin "CLK", mentre il pin "OUT" rappresenta il segnale di uscita. Il funzionamento appena descritto è rappresentato in Fig. 8.18.

E' stato utilizzato il pin "CE" del blocco Latch D il quale serve ad abilitare la generazione dell'uscita quando è a livello logico alto. A questo pin è stato collegato il comando di Latch. In questo modo, nei 100 ns in cui il segnale di Latch è alto, il blocco Latch D funziona normalmente; quindi, funziona in trasparenza, riportando in uscita i dati che vede in ingresso quando il clock è a livello logico alto. Anche in questo blocco è stato lasciato il clock di default da 160 MHz. Dopo 100 ns il Latch si abbassa e, quindi, l'uscita non varia fino al comando di Latch successivo. Quando il

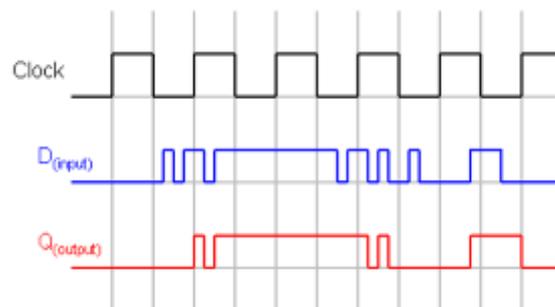


Fig.8.18. Funzionamento Latch D.

segnale di Latch si abbassa, in uscita dal Latch D, rimane l'ultimo dato a 16 bit che era stato mandato in uscita dal blocco stesso un attimo prima che il segnale di Latch si abbassasse. Quando il segnale di Latch ritorna a livello logico alto, l'uscita del Latch D torna nuovamente a variare in base a ciò che gli arriva dal contatore. Questo certifica il fatto che il segnale di Latch è, effettivamente, coinvolto nell'operazione di latch dei canali di acquisizione, come era stato anticipato nella parte introduttiva del capitolo, oltre che nell'avvio dell'intera acquisizione dati. L'uscita del Latch D è collegata ad una Schematic Port Out chiamata "Canale", la quale è mandata in uscita dal subdiagram e verrà utilizzata in un altro subdiagram non ancora commentato.

8.4) Citiroc

Viene descritto, di seguito, il subdiagram "Citiroc". Questo è caratterizzato dal fatto che, al suo interno, viene richiamato il subdiagram appena visto, ossia "Acquisizione". In particolare, come si può osservare in Fig. 8.19, il subdiagram "Acquisizione" viene richiamato e ripetuto 32 volte. Si nota, infatti, che vi sono 32 ingressi "Trig_x" e 32 uscite "Canale_x". Gli ingressi Latch e ResetD sono dei segnali comuni per tutti i canali. Per questo motivo non occorre replicarli 32 volte dato che, si agiscono in tutti e 32 i canali, però sono sempre gli stessi in ciascuno di essi. Questa caratteristica dei segnali Latch e ResetD di non essere replicati nel caso in cui il subdiagram fosse diventato multi-canale è stata decisa nel subdiagram "Acquisizione", in fase di creazione delle Schematic Port In "Latch" e "RstD", scegliendo l'opzione COMMON nella voce "Multichannel

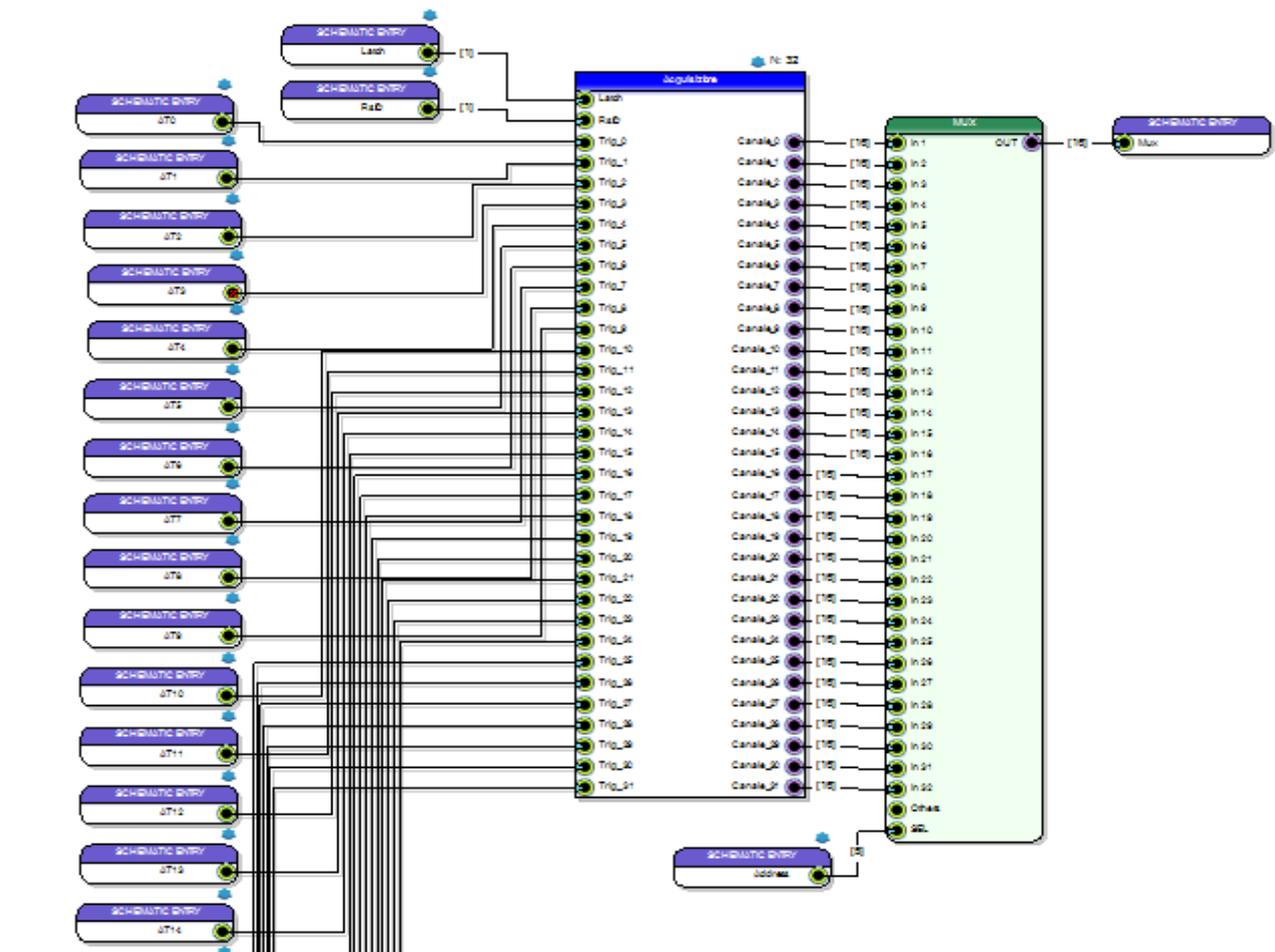


Fig.8.19. Subdiagram "Citiroc".

Behaviour”. Il motivo per cui il subdiagram precedente è stato ripetuto 32 volte è il seguente: in questa fase della progettazione è presente l’inserimento del Multiplexer per canalizzare i 128 canali. Tuttavia, in SCI-Compiler, non è possibile selezionare un MUX da 128 canali; il minimo di canali da poter selezionare è 64. Per questo motivo, è stato scelto di creare questo subdiagram in cui è presente un MUX da 32 canali, che è il numero di canali di un singolo Citiroc.

Negli ingressi “Trig” del blocco “Acquisizione” in versione multi-canale, dovrebbero esserci collegati i canali dei Citiroc. Esiste in SCI-Compiler, nel sottomenù “Board Pin” che permette di utilizzare i pin I / O della scheda, la possibilità di selezionare singolarmente ognuno dei quattro Citiroc presenti sulla scheda. Il suddetto blocco è rappresentato in Fig. 8.20. Tuttavia, gli elementi presenti nel sottomenù “Board Pin” non sono selezionabili quando si lavora in un subdiagram, ma solo nel top del diagramma. A causa di questo, nel subdiagram che si sta descrivendo, sono state usate ancora delle Schematic Port In denominate “ATxx” (con xx da 0 a 31). Questi saranno gli input del subdiagram “Citiroc”, ai quali, quando il subdiagram verrà richiamato nel top del diagramma, verranno collegati i 32 output del blocco “Citiroc Trigger” della Fig. 8.20.

Si può capire, quindi, l’utilità dell’aver replicato 32 volte il subdiagram “Acquisizione”. In questa maniera, si andranno a richiamare nel top del diagramma SCI-Compiler quattro subdiagram “Citiroc”, ognuno dei quali avrà 32 input, e, ad ognuno di essi, verrà collegato il blocco “Citiroc

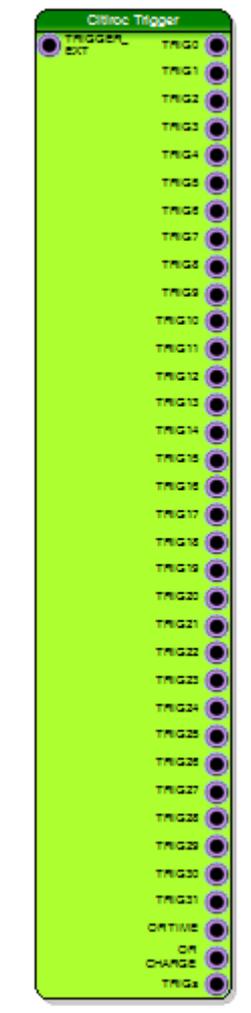


Fig.8.20. Blocco Citiroc Trigger.

Trigger”, selezionando i quattro Citiroc presenti sulla scheda, per un totale di 128 collegamenti. Concludendo la descrizione del subdiagram della Fig. 8.19 (la quale non mostra tutte le Schematic Port In “ATxx” in quanto, includendole tutte, non sarebbero state visibili le scritte dei blocchi), si può notare che i 32 output del blocco multi-canale vengono collegati agli ingressi di un MUX a 32 canali. Il MUX presenta un pin denominato “SEL” al quale si collega una Schematic Port In (“Address”) proveniente da un subdiagram che verrà descritto nel paragrafo successivo. Questo pin serve per la selezione dei canali del MUX da mandare in uscita da esso. Nel caso illustrato, essendo 32 i canali da indirizzare, il segnale che si collega al pin “SEL” è un segnale a 5 bit. L’uscita del MUX è una Schematic Port Out denominata “Mux”, a 16 bit, che è l’output del subdiagram “Citiroc”.

8.5) SelIndirizzi

Il seguente subdiagram, il quale è visibile in Fig. 8.21, ha lo scopo di generare gli indirizzi da inviare al MUX per selezionare i canali dei Citiroc. In questo subdiagram si utilizza un contatore asincrono sensibile ai fronti di salita. Si vede, infatti, che il blocco relativo a questo contatore è identico a quello utilizzato nel subdiagram “Acquisizione”, con la differenza che è presente un pin di input in meno, ossia quello relativo al clock (dovuto al fatto che questo è un contatore asincrono). Il contatore asincrono conta semplicemente i fronti di salita del segnale datogli in ingresso nel pin “In”, fintantoché il pin “GATE” è a livello logico alto.

Nel subdiagram “SelIndirizzi” viene collegato in ingresso al contatore il segnale in uscita dal subdiagram “Clk128”, ovvero 128 impulsi alla frequenza di 5 MHz. Il contatore riporta in uscita sul pin “Counts” un segnale da 32 bit che conta da 1 a 128 alla frequenza di 5 MHz; quindi, gli indirizzi di selezione dei canali del MUX cambieranno a quella frequenza. Si è scelto un contatore asincrono, in quanto non serve sincronizzare il conteggio del contatore con un clock, dato che il fine del contatore di questo subdiagram è contare fino a 128 grazie ad un segnale di ingresso che viene generato in maniera tale che abbia sempre 128 transizioni ad una frequenza fissa; per tale scopo, il contatore asincrono si presta perfettamente. Viene dato in ingresso al contatore anche il segnale di Latch collegandolo al pin “Reset”, in modo tale che il conteggio venga resettato ogni 100 μ s. Se così non fosse, dato che i contatori su SCI-Compiler sono tutti a 32 bit, il contatore supererebbe la quota di 128 conteggi e non ci sarebbe più una corrispondenza con gli indirizzi dei canali del MUX.

Il valore dei conteggi in uscita dal contatore passa attraverso un blocco sottrattore SUB, dove viene sottratto il valore 1 ad ogni valore in uscita dal contatore. Questo passaggio è necessario perché il contatore fornisce in uscita, come primo valore contato, il numero 1. Dato che i valori in uscita dal

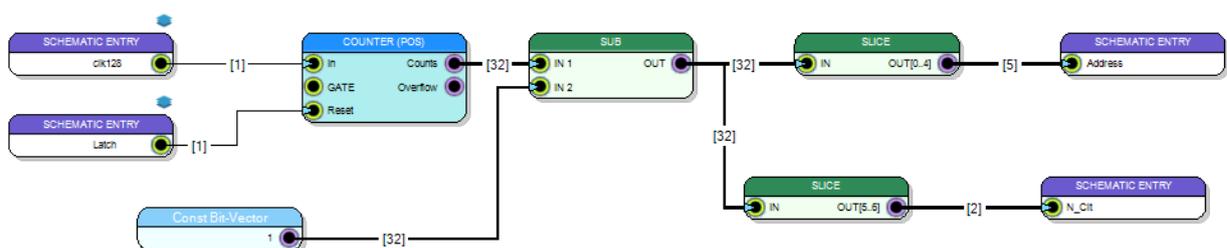


Fig.8.21. Subdiagram "SelIndirizzi".

contatore corrispondono agli indirizzi del MUX, non sottraendo il valore 1 ad essi non si andrebbe mai a selezionare il primo canale del MUX, ossia il canale 0 corrispondente all'indirizzo 0. Inserendo il blocco SUB e sottraendo 1 al segnale in uscita da "Counts" si permette la selezione dei canali da 0 a 127.

L'uscita del blocco SUB è una parola binaria a 32 bit. I bit necessari per selezionare 32 canali del MUX del subdiagramma "Citiroc" sono 5. Si usa, quindi, il blocco SLICE, già visto precedentemente, per estrarre i primi 5 bit, partendo sempre dai bit meno significativi, e collegarli alla Schematic Port Out "Address". Questo è il primo output del subdiagramma "SelIndirizzi" che, come si è visto, diventa un input del subdiagramma "Citiroc" collegandosi al pin "SEL" del MUX a 32 canali presente nel diagramma, permettendo la selezione dei 32 canali. E' stato detto, però, che il subdiagramma "Citiroc", e quindi anche il MUX al suo interno, viene chiamato quattro volte nel diagramma top per ottenere 128 canali. Si ottengono quattro configurazioni identiche a quella in Fig. 8.22 nel top del diagramma, con l'unica differenza che il blocco "Citiroc Trigger" che si collega a tutti gli input "ATxx" si riferirà a un Citiroc diverso per ognuna delle configurazioni. Si notano i comandi di Latch e ResetD espressi come blocchi "Variable Write" che si collegano in ingresso al subdiagramma "Citiroc", insieme al segnale "Address", quest'ultimo, anche, utilizzato come blocco "Variable Write". Si ricorda che questi blocchi sono utilizzati per evitare l'uso di molti fili per i collegamenti tra i vari pin, in modo da rendere il diagramma più ordinato. L'uscita "Mux" dal blocco "Citiroc" sul top, illustrata in Fig. 8.22, è data da una parola binaria a 16 bit, rappresentante l'informazione

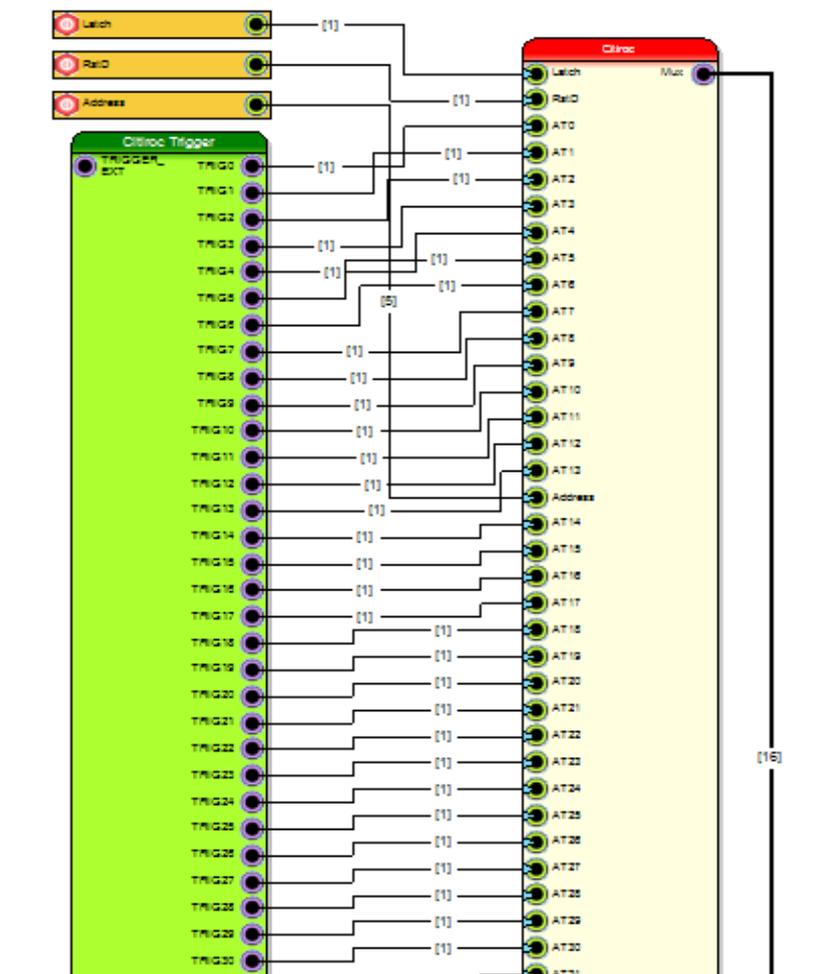


Fig.8.22. Parte di diagramma sul top.

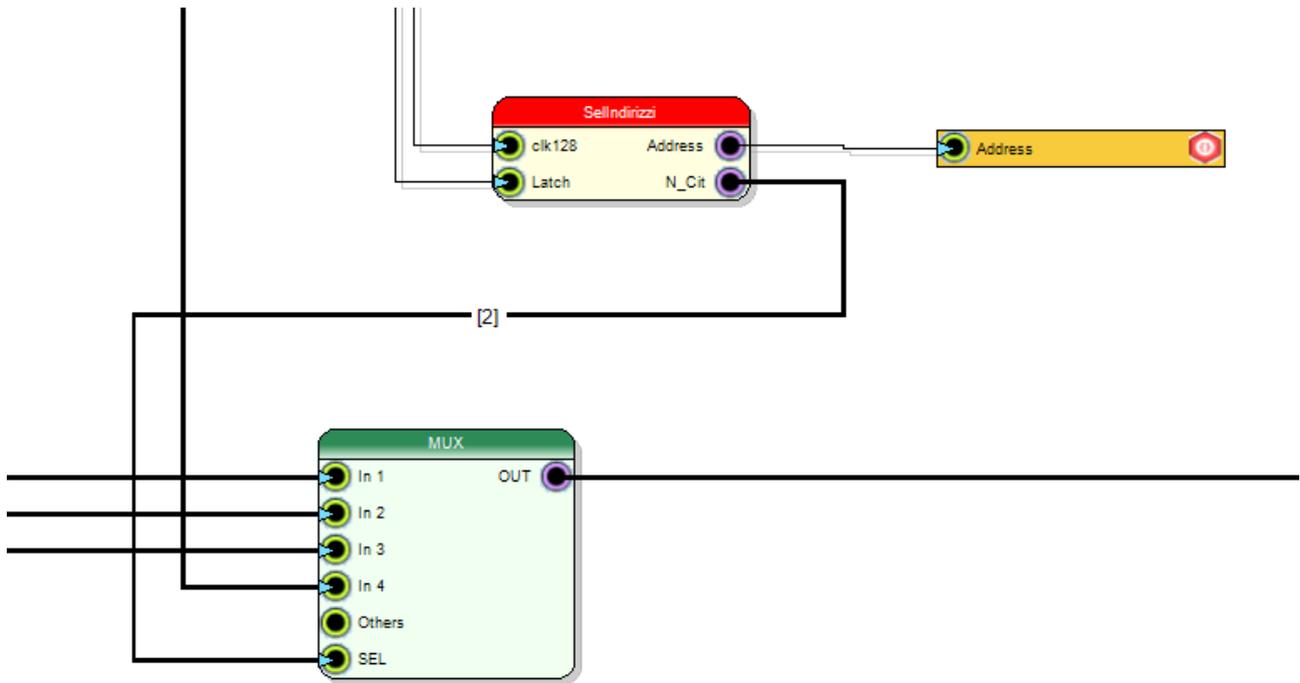


Fig.8.23. Parte di diagramma nel top relativa alla selezione dei 32 canali provenienti dai 4 blocchi "Citiroc".

binaria relativa alle particelle rilevate dai SiPM. Questa cambia ogni 200 ns, poiché con quel periodo temporale sono fatti variare i canali del MUX dal segnale Address in uscita dal subdiagramma "SellIndirizzi". Le uscite "Mux" dei quattro subdiagrammi "Citiroc" presenti sul top vengono collegate ad un altro MUX (presente nel top) a quattro ingressi, come mostrato in Fig. 8.23.

Questo MUX a 4 canali ha il seguente scopo: fornire in uscita i 32 canali del primo Citiroc che provengono dall'uscita "Mux" del blocco "Citiroc" al quale è collegato il blocco "Citiroc Trigger" relativo al Citiroc 1; mandare, poi, in uscita i 32 canali del secondo Citiroc che provengono dall'uscita "Mux" del blocco "Citiroc" al quale è collegato il blocco "Citiroc Trigger" relativo al Citiroc 2, e così via fino al quarto Citiroc. Affinché questo avvenga occorre sincronizzare l'indirizzamento dei 32 canali del MUX presente nel blocco "Citiroc", con l'indirizzamento dei 4 canali di questo MUX. Al pin "SEL" viene collegato un segnale, il quale è il secondo output del subdiagramma "SellIndirizzi", denominato "N_Cit" (tale collegamento è rappresentato in Fig. 8.23). Come raffigurato in Fig. 8.21, il segnale N_Cit è una parola binaria a 2 bit e corrisponde al sesto e settimo bit dei 32 provenienti dal pin "Counts" in uscita dal contatore asincrono. Per prelevare questi due bit si è usato ancora una volta il blocco SLICE.

All'inizio del conteggio del contatore i bit 6 e 7 dei 32 totali dell'uscita "Counts" sono pari a 0. Il conteggio procede fino a 31, per cui la parola binaria dei primi 7 bit meno significativi dei 32 totali è 0011111. In questo momento il MUX a 32 canali si trova in corrispondenza del 32esimo input mentre il MUX a 4 canali si trova in corrispondenza del primo input. Al conteggio successivo si ha 0100000: con questi 7 bit si sta puntando il primo canale del secondo Citiroc. Infatti, il segnale N_Cit porta in ingresso al pin "SEL" del MUX a 4 canali la parola binaria 01, passando dal primo al secondo canale e, quindi, dal primo al secondo Citiroc. Dopo che vengono indirizzati tutti i 32 canali del secondo Citiroc, il segnale N_Cit diventa 10 e va a selezionare il terzo canale del MUX a 4 canali; infine, con la parola binaria 11 si va a selezionare anche l'ultimo canale del MUX a 4

canali, quindi l'ultimo Citiroc. Dato che ogni canale di acquisizione viene mandato ogni 200 ns, il totale dei 128 canali dei Citiroc viene spedito in 25,6 μ s; dopodiché, il subdiagramma "Clk128" non manda più impulsi fino al Latch successivo, quindi il contatore non conta più e in uscita dal MUX a 4 canali rimane il valore relativo all'ultimo conteggio, ossia il 128esimo. Il conteggio viene azzerato dal segnale di Latch come spiegato precedentemente, facendo ripartire, poi, un nuovo conteggio con l'arrivo degli impulsi provenienti da "Clk128". In Fig. 8.23, oltre al MUX a 4 canali si può osservare il subdiagramma "SelIndirizzi" richiamato nel top, a cui vi arrivano gli ingressi "Latch" e "clk128" da un'altra parte del diagramma e in uscita fornisce "N_Cit" che si collega al MUX a 4 canali e "Address" che viene collegato ad una "Variable Read" per evitare di adoperare troppi fili.

8.6) Serializzatore

Il subdiagramma "Serializzatore" ha lo scopo di serializzare i dati a 16 bit che provengono dal MUX a 4 canali, a causa del fatto che la scheda DT5550W / A55CIT4 è in grado di comunicare con l'esterno attraverso gli otto LEMO I / O che permettono una trasmissione seriale dei dati. Questo è il subdiagramma dal quale vengono mandati in uscita, tramite i LEMO, alcuni segnali che comunicheranno con la scheda DD Low Int, appositamente progettata per gestire i dati provenienti dalla DT5550W / A55CIT4 ed adattarli al protocollo di trasmissione dati del DDS. I segnali mandati in uscita dalla scheda sono tre e sono chiamati Serial, Load e Clock.

In Fig. 8.24 è riportato il subdiagramma in questione. Questo prende in ingresso i 128 impulsi provenienti dal subdiagramma "Clk128". Questi impulsi sono fondamentali anche in questo subdiagramma in quanto partecipano attivamente alla generazione del segnale Serial e del segnale Load. Prima di tutto occorre definire la funzione eseguita dal blocco di maggior importanza del subdiagramma, ossia il blocco SERIALIZER (Fig. 8.25). Tale blocco ha la facoltà di convertire un dato da una forma parallela ad una forma seriale. Il pin "DATA" in ingresso deve ricevere il dato da serializzare; nel caso in esame, esso riceve una Schematic Port In denominata "Parallel", che rappresenta l'uscita del MUX a 4 canali, quindi il dato a 16 bit (questo è il secondo e ultimo input del subdiagramma in esame). L'input "CE" deve rimanere a livello logico alto durante tutto il tempo

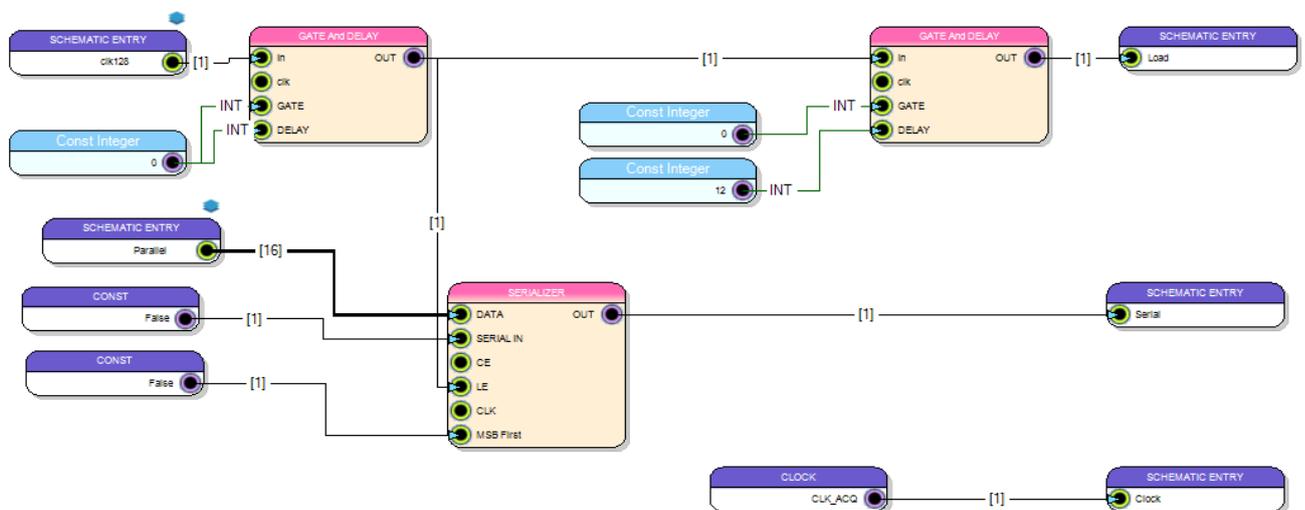


Fig.8.24. Subdiagramma "Serializzatore".

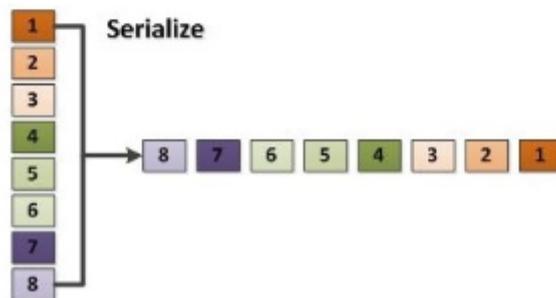
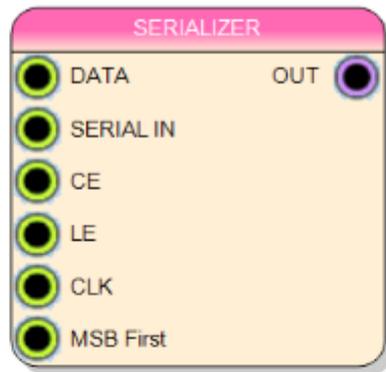


Fig.8.25. Blocco SERIALIZER.

necessario alla serializzazione del segnale mandato in “DATA” e, infatti, gli viene assegnato il suo valore di default, ovvero 1. Il pin “CE”, se lasciato sempre al valore 1, permette di concatenare altri bit alla serializzazione dopo che il segnale d’ingresso in “DATA” è stato serializzato. Questi bit addizionali sono presi dal segnale che viene mandato in ingresso al pin ”SERIAL IN”. Nel subdiagram, in quest’ultimo pin, viene collegato il blocco False che indica lo 0 logico, in modo da concatenare una serie di 0 poiché, terminata la serializzazione dei 16 bit, non è necessario concatenare nessun bit aggiuntivo. Il pin “LE”, quando è a livello logico alto, sancisce l’inizio della serializzazione dei dati presenti in “DATA”. Nel subdiagram “Serializzatore”, in tale pin, viene collegata la serie di impulsi provenienti da “Clk128”. Questa operazione serve per sincronizzare la serializzazione dei 16 bit con la tempistica con la quale questi arrivano in ingresso al pin “DATA”. Si è detto che i vari canali del MUX sono fatti scorrere per portare in uscita il dato ad una frequenza di 5 MHz, quindi anche la serializzazione deve avvenire ad una frequenza di 5 MHz. Per questo motivo i 128 impulsi vengono mandati in ingresso a “LE”, in modo che si faccia partire la serializzazione dei dati ogni 200 ns. Al pin “CLK” non si collega nulla, lasciando, quindi, il suo valore di default a 160 MHz. Si ricorda che un ciclo di clock a quella frequenza è pari a 6,25 ns circa. Questo vuol dire che ogni 6,25 ns il blocco SERIALIZER manda in uscita un bit della parola

binaria da serializzare. Essendo 16 i bit che devono essere serializzati, in circa 100 ns il dato parallelo a 16 bit viene serializzato dal blocco SERIALIZER. Nei restanti 100 ns vengono mandati in uscita una serie di 0 logici, dato che è stato collegato il blocco False al pin “SERIAL IN”. Dopodiché arriva un altro impulso dei 128 al pin “LE”, che fa partire una nuova serializzazione di un nuovo dato proveniente dal canale successivo del MUX. Infine, vi è il pin “MSB First” che, impostato a 0 come nel caso del subdiagram in analisi, fa sì che i bit mandati in “OUT” dalla serializzazione vadano dal meno significativo al più significativo dei 16.

Si può notare che i 128 impulsi, prima di arrivare al pin “LE” del SERIALIZER, passano attraverso il blocco GATE And DELAY, a cui viene dato il valore 0 in ingresso sia al pin “GATE” che al pin “DELAY”. Il motivo è il seguente: i 128 impulsi servono sia all’indirizzamento dei canali del MUX che a temporizzare la serializzazione dati. Tuttavia, i 128 impulsi non possono arrivare nello stesso momento nel subdiagram “SelIndirizzi” e in “Serializzatore”, ma devono prima arrivare nel pin di selezione degli indirizzi del MUX, altrimenti la serializzazione comincerebbe prima che i dati siano stati mandati in uscita dal MUX. Inserendo 0 nel pin “DELAY”, come è stato detto nel Par. 8.2, si aggiunge un ritardo al segnale in ingresso al blocco di 30 ns. Con questo ritardo, grazie a test effettuati con l’oscilloscopio, si è osservato che le operazioni di indirizzamento dei canali del MUX e di inizio della serializzazione dati sono sincronizzate tra di loro. Per quanto riguarda il fatto che è stato messo 0 nel pin “GATE”, si può dire che non è importante per quanto tempo il segnale rimanga a livello logico alto, ma è importante che vengano viste le transizioni del segnale; per questo motivo la scelta del valore da inserire sul pin “GATE” è indifferente, ed è stato scelto il valore 0 che equivale ad un ciclo di clock. L’uscita dal pin “OUT” del SERIALIZER viene portata in ingresso alla Schematic Port Out “Serial” che sarebbe uno dei tre segnali che vengono condotti in uscita dalla scheda attraverso i LEMO.

L’uscita del GATE And DELAY, di cui si è già parlato, viene mandata in ingresso ad un secondo GATE And DELAY, che ha sempre il valore 0 in ingresso a “GATE” (assegnando, quindi, al segnale in questione un livello logico alto di durata pari a 6,25 ns), ma, questa volta, ha il valore 12 in ingresso a “DELAY”. L’uscita di questo secondo GATE And DELAY va in ingresso alla Schematic Port Out “Load”, il quale è il secondo segnale che esce dalla scheda. La necessità di questo segnale verrà esposta in maniera più chiara nel capitolo successivo, in quanto essenziale all’elettronica della scheda a cui si collegherà la DT5550W / A55CIT4, ossia la DD Low Int. Ciò che si può dire, ora, è che il segnale di Load è un impulso che deve avere un periodo di 200 ns e deve essere mandato in uscita dalla scheda il colpo di clock successivo all’ultimo bit dei 16 serializzati dal SERIALIZER. Affinché venisse rispettata questa specifica, si è sfruttato il fatto che il segnale in uscita dal primo GATE And DELAY sia il segnale che fa partire la serializzazione; quindi, esattamente 100 ns dopo questo impulso, tutti e 16 i bit vengono serializzati. Mandando questo segnale in ingresso ad un secondo GATE AND DELAY e dando a quest’ultimo un ritardo in cicli di clock pari a 100 ns, si ottiene un impulso appena dopo l’ultimo dei 16 bit serializzati, periodicamente ogni 200 ns. Tuttavia, c’è da considerare il fatto che il blocco GATE And DELAY induce di per sé un ritardo di circa 30 ns al segnale che riceve in ingresso; quindi, mettere in ingresso a “DELAY” il valore 16, che corrisponde esattamente a 100 ns, sarebbe errato. Facendo delle prove collegando un registro al pin “DELAY”, scrivendo da tastiera vari valori su tale registro con la scheda in esecuzione e visualizzando sull’oscilloscopio la traslazione orizzontale del segnale in questione al variare dei valori, si è visto che inserendo il valore 12 al pin “DELAY” si ottiene

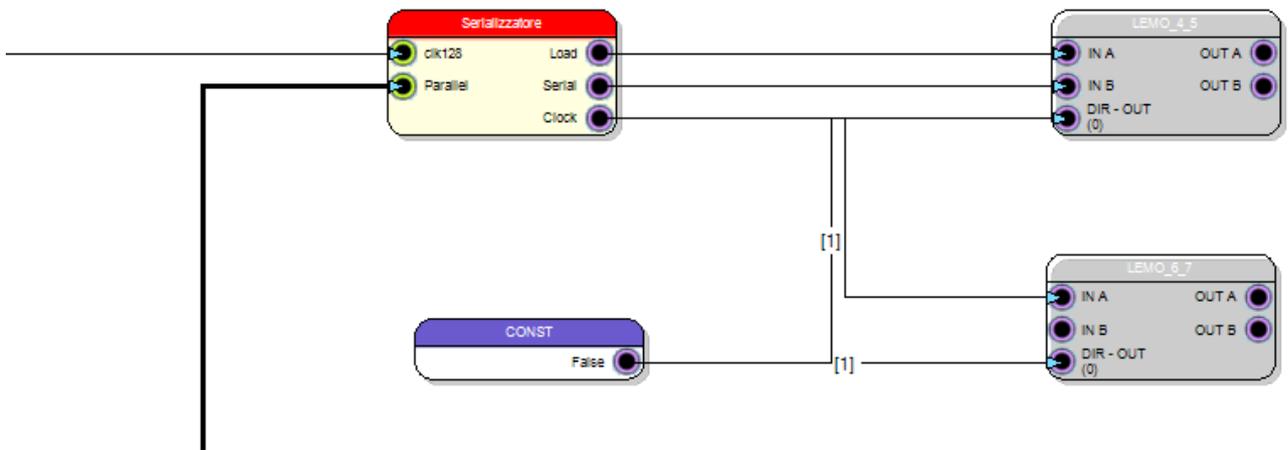


Fig.8.26. Parte del diagramma top dove viene mostrato il subdiagram "Serializzatore" con le sue uscite collegate ai LEMO.

l'impulso esattamente il ciclo di clock successivo all'ultimo bit serializzato.

Infine, viene generato il terzo segnale in uscita, chiamato "Clock", che, appunto, è un vero e proprio segnale di clock. Sempre per ragioni che verranno illustrate nel capitolo successivo, la scheda DD Low Int deve ricevere in ingresso un segnale di clock proveniente dalla DT5550W / A55CIT4. Il clock che è stato scelto, uno dei tanti messi a disposizione da SCI-Compiler per la DT5550W / A55CIT4, è quello da 160 MHz, il quale, come si può vedere dalla Fig. 8.24, viene collegato alla Schematic Port Out denominata "Clock".

In Fig. 8.26 è mostrata una parte del diagramma top dove viene richiamato il subdiagram "Serializzatore". Nell'ingresso "clk128" arriva l'uscita del subdiagram "Clk128" mentre nell'ingresso "Parallel" arriva il filo proveniente dal pin di uscita del MUX a 4 canali, anch'esso presente nel top. Le tre uscite Load, Serial e Clock vengono collegate, rispettivamente, ai LEMO 4, 5 e 6. In Fig. 8.27 è riportata una vista sull'oscilloscopio del corretto funzionamento della serializzazione e di tutti i segnali portati in uscita sui tre LEMO. L'impulso rosso (il quale non è un segnale che verrà trasferito nella DD Low Int, ma è stato utile visualizzarlo per verificare il corretto funzionamento di tutto il procedimento di serializzazione) è il segnale che va in ingresso al pin "LE" del SERIALIZER che fa partire la serializzazione. Il segnale giallo è la serie di 16 bit della parola binaria parallela 10101010101011. Si può notare in basso a destra che la griglia orizzontale dell'oscilloscopio è stata impostata a 20 ns per divisione. Sapendo questo, è possibile constatare dalla figura che tra il primo e l'ultimo bit del dato serializzato passano esattamente 100 ns. L'impulso in azzurro è il segnale di Load, che si trova esattamente il ciclo di clock successivo all'ultimo bit serializzato. Infine, in basso rispetto agli altri, si vede un segnale verde che sarebbe l'ultimo output della DT5550W / A55CIT4, ossia il segnale di Clock a 160 MHz.

8.7) Top

Il top del diagramma, che costituisce il firmware implementato, è la parte principale del codice dove vengono chiamati tutti i subdiagram fin qui descritti, i quali hanno i loro input e output definiti al loro interno. Nel top vengono inseriti i blocchi che simboleggiano degli elementi fisici della DT5550W / A55CIT4, ad esempio i canali dei Citroic, sui quali arrivano gli impulsi provenienti dai

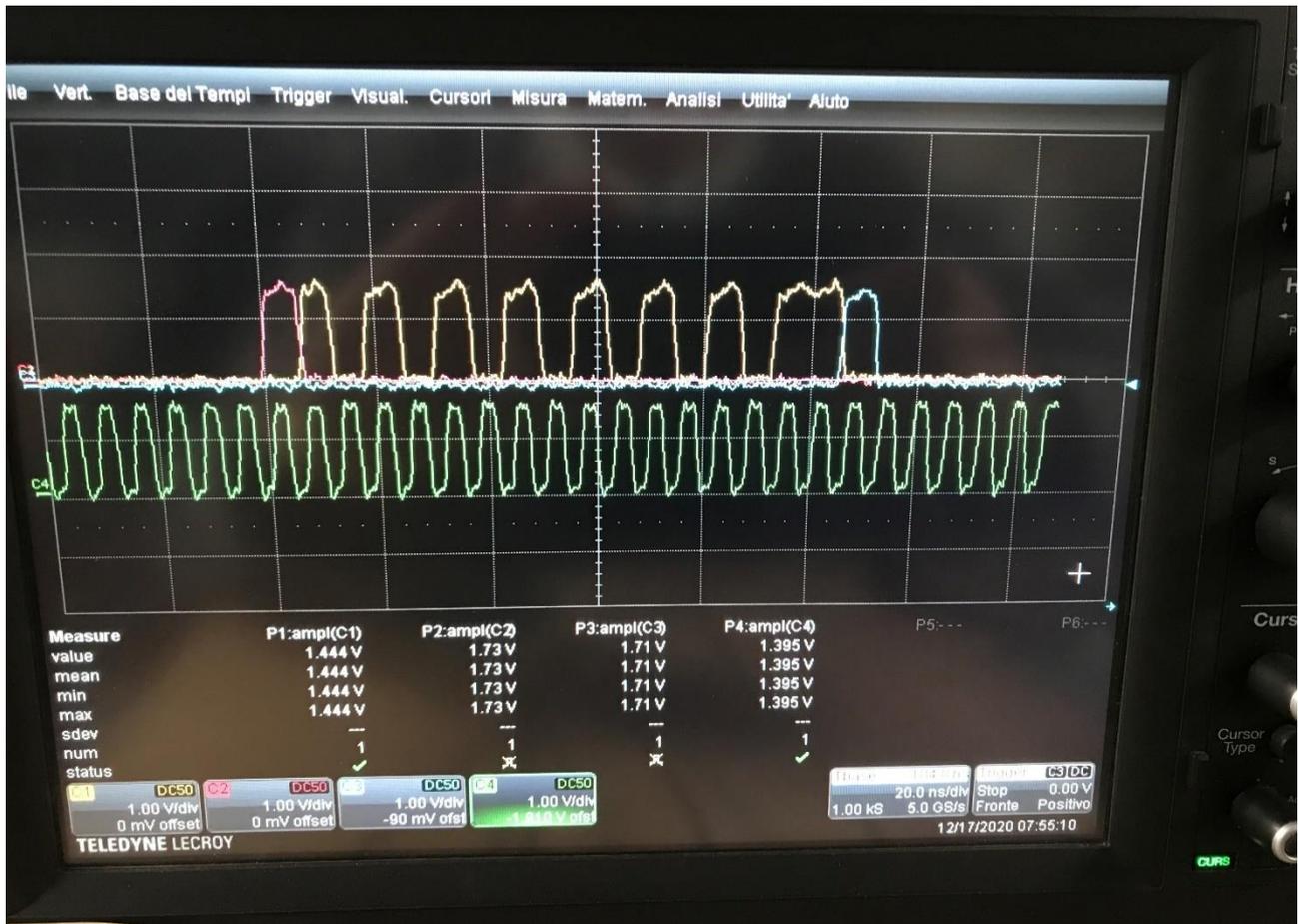


Fig.8.27. Visualizzazione su oscilloscopio dei segnali in uscita dal subdiagram "Serializzatore".

fotomoltiplicatori, oppure i LEMO; tutti componenti della scheda che non possono essere inseriti nei subdiagram, ma solo nel top del diagramma. In Fig. 8.28 è mostrata la sezione superiore della scheda "Project File" di SCI-Compiler con tutti i subdiagram illustrati nei paragrafi precedenti e il top in alto. In Fig. 8.29 è mostrato lo schema a blocchi del top. Essendo costituito da molti blocchi non è possibile vederlo bene dall'immagine. Tuttavia, durante la spiegazione dei singoli subdiagram, in alcuni casi, sono state mostrate alcune parti del top in maniera più ravvicinata. La Fig. 8.7, ad esempio, è la parte di blocchi connessi in alto a sinistra nella Fig. 8.29 e riguardano la generazione dei segnali di Latch e ResetD. Nella parte sull'estrema destra, si può notare ciò che è



Fig.8.28. Sezione superiore del "Project File" che mostra l'organizzazione del codice SCI-Compiler implementato.

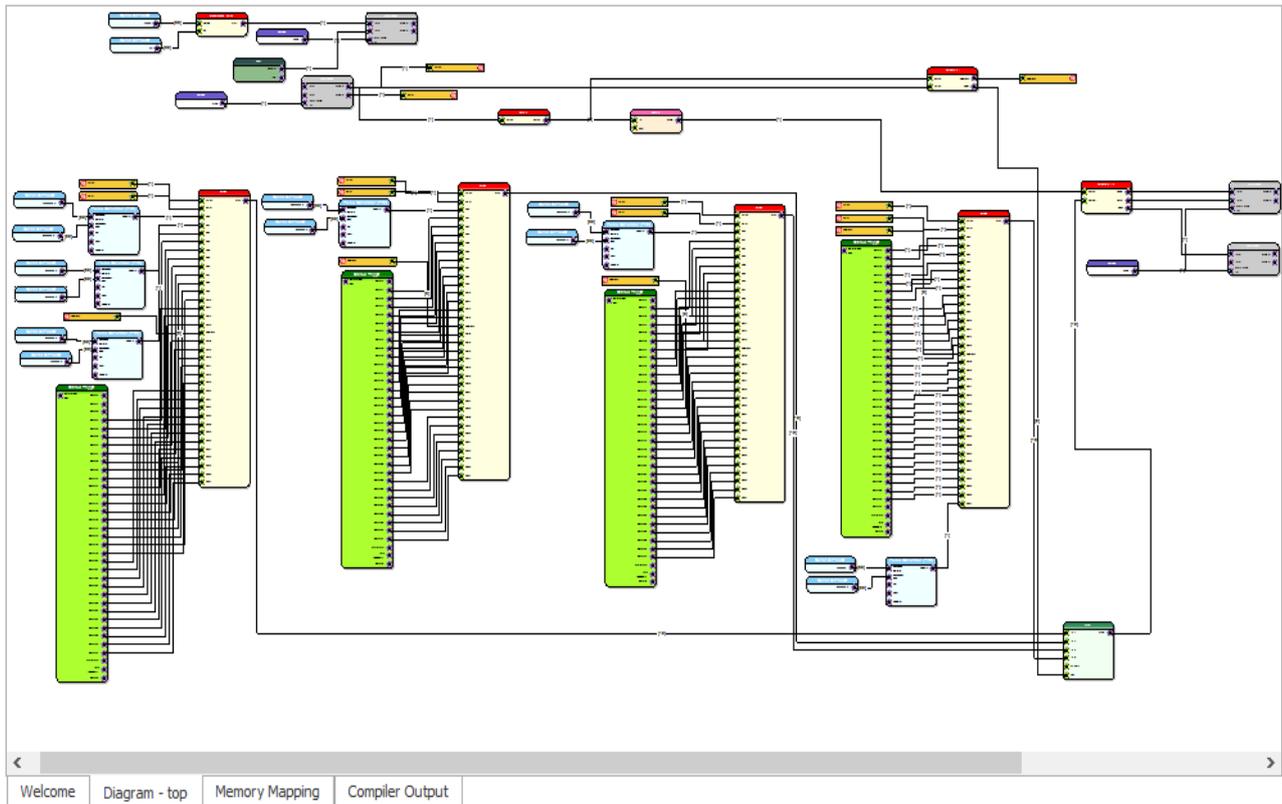


Fig.8.29. Top del diagramma.

rappresentato in Fig. 8.26, ossia le uscite del subdiagramma “Serializzatore” che si collegano ai LEMO. In basso a destra, invece, si vede il MUX a 4 canali; il blocchetto rosso in alto (verso destra) che collega un filo al pin di selezione degli indirizzi di questo MUX è il subdiagramma “SelIndirizzi”. Si possono osservare, tra l’insieme di blocchi dedicati alla generazione di Latch e ResetD e il blocco rosso “SelIndirizzi”, due blocchi, uno rosso e uno rosa. Il primo è il subdiagramma “Clk128” che ha un ingresso (il Latch) e un’uscita (i 128 impulsi). Dall’uscita di questo blocco si vede un filo che va in ingresso al blocco “SelIndirizzi” e un altro che entra nel blocco “Serializzatore” tutto a destra. Prima di arrivare a “Serializzatore”, però, i 128 impulsi vengono ritardati dal blocco rosa poco fa nominato, che sarebbe un blocco DELAY. Questo blocco DELAY è indicato anche nella Fig. 8.3, dove viene mostrato lo schema a blocchi del funzionamento del firmware, chiamato in quel caso “Ritardo” e posto, appunto, tra il blocco Clk128 e Serializzatore. Il motivo della presenza di questo blocco è dovuto alla necessità di rispettare un’altra specifica non ancora enunciata finora, ossia che tra il fronte di salita del segnale di Latch e quello del segnale di Load del primo canale, scorresse un intervallo di tempo multiplo di 100 ns. Un certo intervallo temporale tra i due fronti di salita dei due segnali esiste già, in quanto, tra l’arrivo del Latch nel subdiagramma “Clk128” e l’arrivo del primo dei 128 impulsi al subdiagramma “Serializzatore”, passa un certo arco temporale non controllabile dovuto a ritardi fisiologici di lavoro dei singoli blocchi dove passano i vari segnali; basti pensare, ad esempio, ai blocchi GATE And DELAY utilizzati, che di per sé introducono 30 ns di ritardo tra l’ingresso e l’uscita. In più, ci sono 200 ns tra l’impulso che fa partire la serializzazione e il Load. Visualizzando questo intervallo con l’oscilloscopio si è potuto rilevare che questo si aggirasse intorno ai 550 ns. Affinché si avesse un valore multiplo di 100 ns, come da specifica, si è aggiunto il DELAY in quel punto del diagramma, impostando

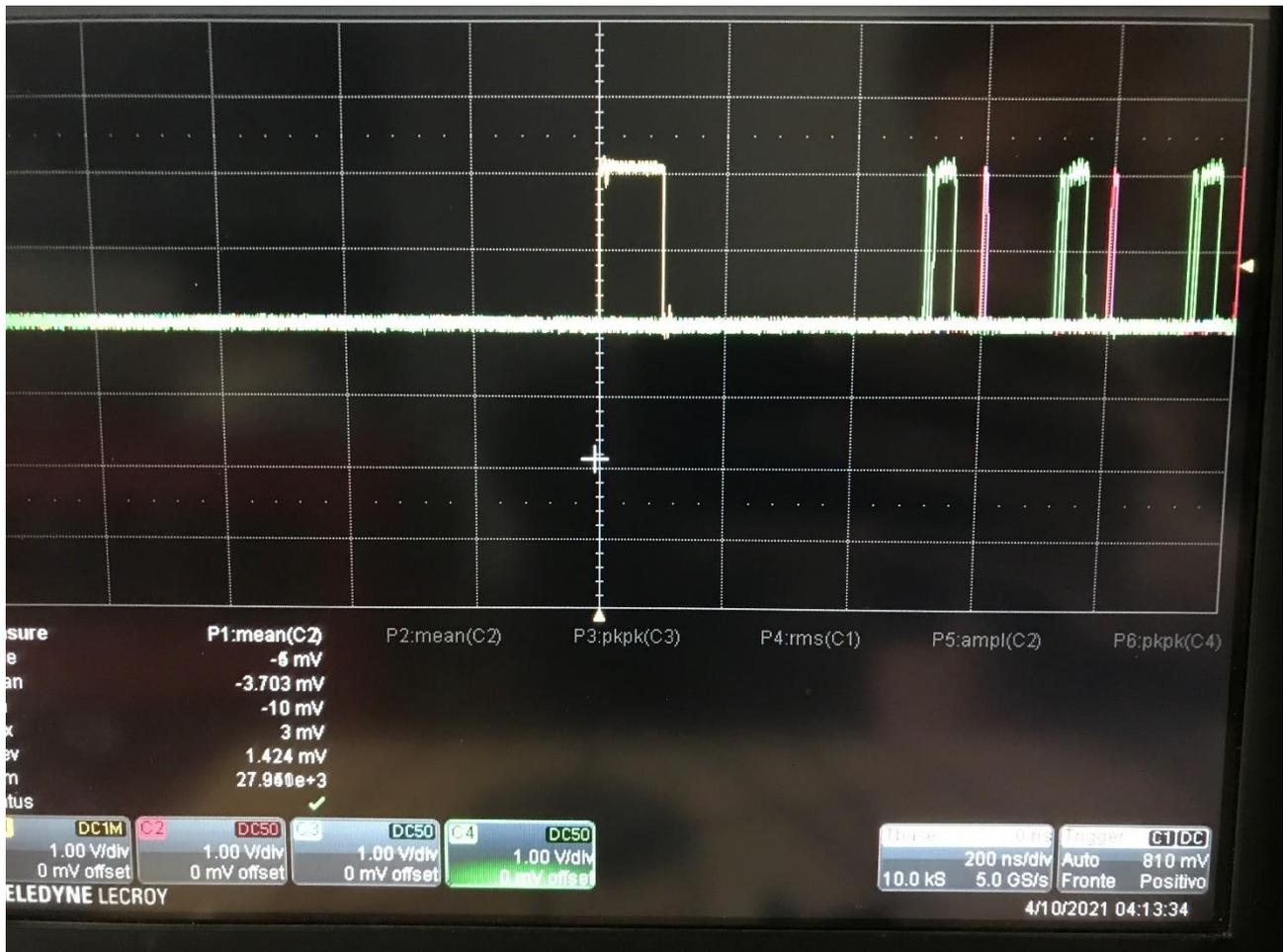


Fig.8.30. Intervallo di 600 ns tra fronte di salita del Latch e il Load.

l'aggiunta di 8 cicli di clock di ritardo che, con il solito clock da 160 MHz, sono 50 ns in più, per un totale di 600 ns tra i due fronti di salita. L'intervallo è mostrato in Fig. 8.30. Il segnale giallo è l'impulso di Latch mentre il rosso è il Load. Tra quest'ultimo e il fronte di salita del Latch c'è una distanza orizzontale di tre divisioni. In basso a destra nell'immagine si può notare che la griglia orizzontale dell'oscilloscopio è impostata su 200 ns per divisione; si deduce che la distanza temporale tra i due fronti di salita è effettivamente di 600 ns, ossia un multiplo di 100 ns.

L'ultima parte rimasta da esporre del diagramma top in Fig. 8.29 è l'insieme di blocchi già visti in Fig. 8.22. Questo insieme di blocchi è costituito dal blocco "Citiroc Trigger", avente 32 uscite, che sono collegate ai 32 ingressi del subdiagram "Citiroc", al quale sono collegati in ingresso anche i segnali di Latch e ResetD e il segnale a 5 bit di selezione di indirizzi del MUX a 32 canali, presente all'interno del suddetto subdiagram. Questa configurazione viene ripetuta quattro volte nel subdiagram per ottenere 128 canali che vengono indirizzati grazie al MUX a 4 canali come ampiamente esposto in precedenza.

Per testare la funzionalità dell'intero programma, e capire se fosse possibile procedere alla progettazione della scheda di interfaccia al DDS, la DD Low Int, sono stati simulati dei dati in ingresso al subdiagram "Citiroc". In Fig. 8.29, infatti, si notano dei blocchi azzurri in ingresso ad alcuni pin del blocco "Citiroc". Quei blocchi sono dei PULSE GENERATOR che sono stati utilizzati per simulare i dati da mandare in determinati canali. Sono state spedite tre onde quadre nei

primi tre canali del Citiroc 1, rispettivamente alla frequenza di 1 Hz, 2 Hz e 4 Hz. Un'onda quadra da 0,5 Hz è stata inviata al primo canale del Citiroc 2 e una da 0,25 Hz al primo canale del Citiroc 3. Infine, un'onda quadra da 160 Hz è stata mandata all'ultimo canale del Citiroc 4. In questo modo, sull'oscilloscopio, ci si aspetta di vedere, nei primi tre canali, nel 33esimo, nel 65esimo e nel 128esimo, i 16 bit serializzati che variano, contando in ordine crescente, alla frequenza delle varie forme d'onda date in ingresso nei rispettivi canali.

In Fig. 8.31 è mostrata una visione dell'oscilloscopio in cui si vede la variazione dei 16 bit alle frequenze di 1, 2 e 4 Hz dei primi tre canali. L'impulso giallo è il segnale di Latch, che dà inizio all'acquisizione; dopodiché si osservano gli impulsi azzurri che sono i conteggi provenienti dai vari canali alle frequenze fissate prima. Infine, in verde, l'impulso di Load che indica l'avvenuta serializzazione dei 16 bit.

Per una più semplice visualizzazione dei canali, è stato impostato lo zero temporale dell'oscilloscopio in corrispondenza del fronte di salita del Latch, il quale è stato centrato, nel monitor dello strumento, facendo coincidere il suo fronte di salita con la linea verticale più marcata che si può notare nella griglia. In questo modo, è stato facile individuare se i bit che variavano negli altri canali, lo facessero effettivamente nei canali in cui sono state mandate le onde quadre con il PULSE GENERATOR nel codice. In Fig. 8.32 si vede un canale che conta, per il quale bisognava verificare se fosse il primo canale del Citiroc 2 come ci si aspettava. Il fronte di salita del Load è stato centrato in corrispondenza della linea verticale marcata del monitor. In basso a destra si può osservare il valore $-7,004 \mu\text{s}$ in corrispondenza del campo "Tbase". Esso indica che quel fronte di salita del Load si trova esattamente $7,004 \mu\text{s}$ dopo il fronte di salita del Latch dove si è definito l'origine temporale. Se a quel valore si sottraggono $0,6 \mu\text{s}$ (ovvero la distanza tra fronte di salita del Latch e il fronte di salita del primo Load) e il risultato viene diviso per $0,2 \mu\text{s}$ (che sarebbe il tempo in cui viene mandato un canale) si ottiene il numero del canale, relativo ai dati a 16 bit corrispondenti al segnale di Load, che si sta considerando. Facendo il calcolo si ottiene il valore 32,

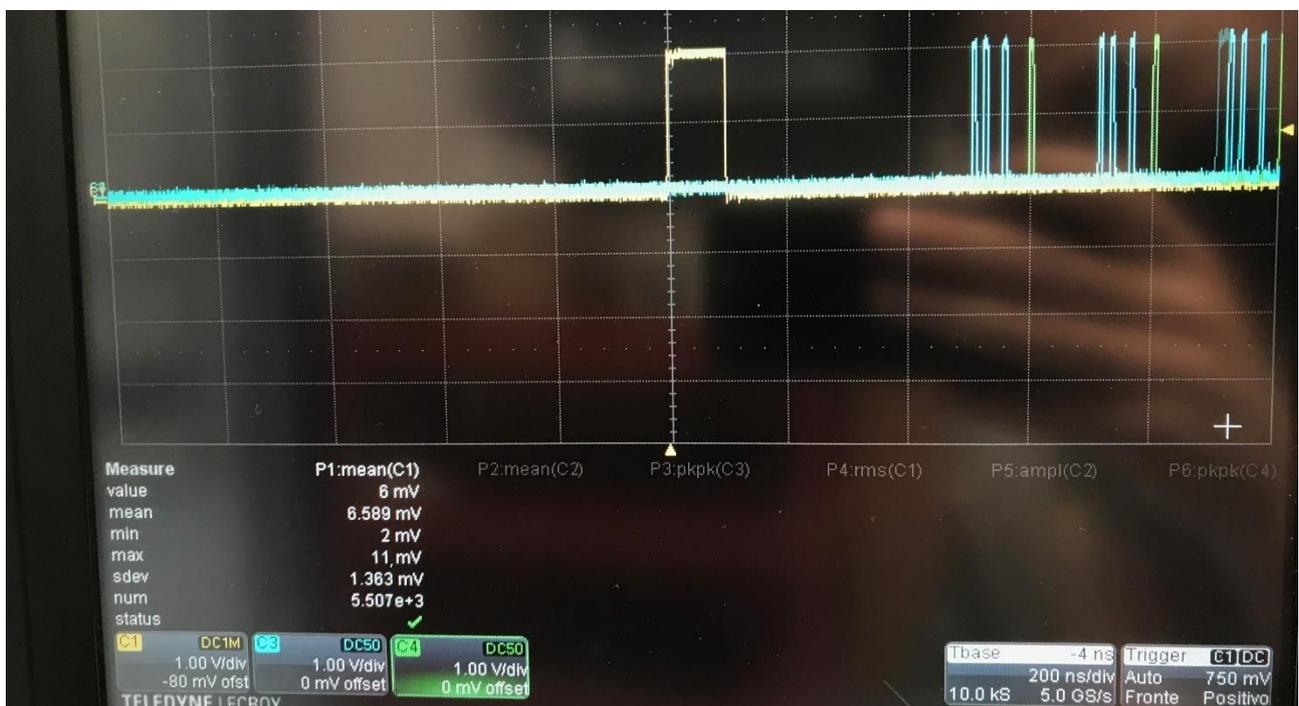


Fig.8.31. Conteggio dei primi tre canali del Citiroc 1.

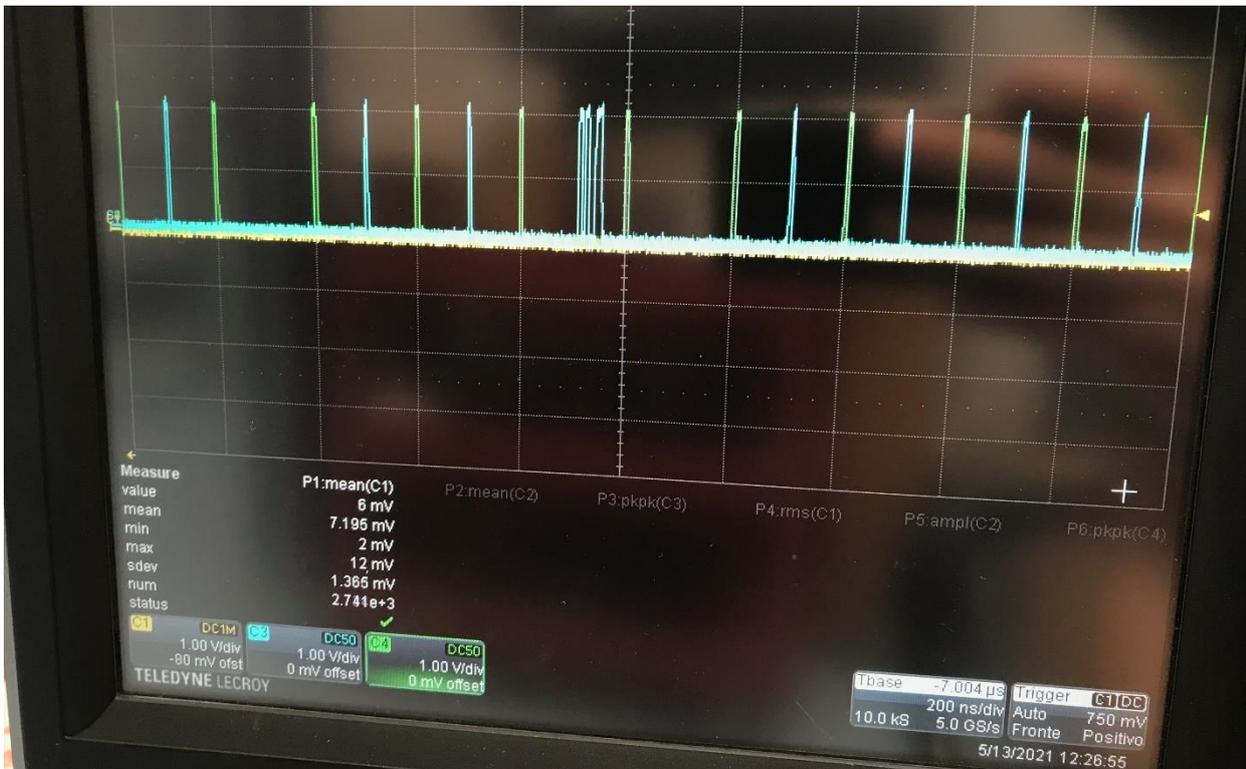


Fig.8.32. Conteggio del primo canale del Citiroc 2.

che sarebbe il 33esimo canale, considerando che si parte dal canale 0. Si è verificata, quindi, anche la correttezza dell'indirizzamento di questo canale, il quale, quando è stato osservato, variava alla frequenza di 0,5 Hz. Replicando questo ragionamento, è possibile verificare anche gli ultimi due canali visualizzati sull'oscilloscopio. In Fig. 8.33, il Load si trova 13,404 μ s dopo il fronte di salita

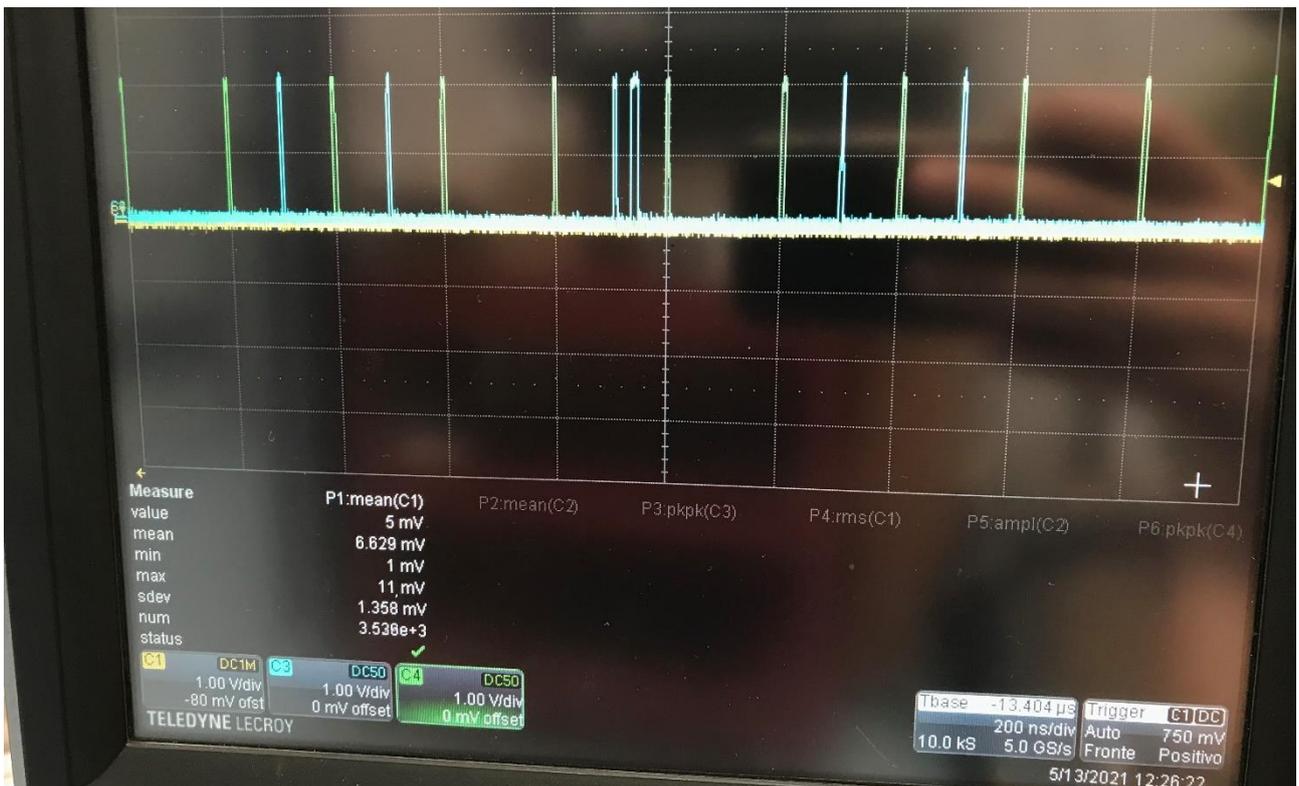


Fig.8.33. Conteggio del primo canale del Citiroc 3.

del Latch. Sottraendo $0,6 \mu\text{s}$ e dividendo per $0,2 \mu\text{s}$ si ottiene 64, ovvero il 65esimo canale. Esso variava con una frequenza più lenta di tutti ($0,25 \text{ Hz}$). L'ultimo canale è visibile in Fig. 8.34. Il calcolo potrebbe anche essere evitato come per i primi tre canali, in quanto è visibile ad occhio che si tratta dell'ultimo canale. Tuttavia, viene comunque svolto per verificare che, effettivamente, siano stati mandati tutti i canali, fino al 128esimo. Il Load si trova $26,004 \mu\text{s}$ dopo il fronte di salita del Latch. Effettuando il calcolo, viene fuori il valore 127 (128esimo canale). Inoltre, i bit variavano ad una velocità molto elevata, confermando il fatto che si trattasse dell'onda quadra a 160 Hz . Si può concludere, quindi, che i canali sono indirizzati nell'ordine giusto e con le tempistiche attese. Vengono trasferiti attraverso i LEMO 128 canali, con 16 bit per canale, in un tempo totale pari a $26,004 \mu\text{s}$.

Il segnale di Latch ha una periodicità di $100 \mu\text{s}$, per una frequenza di lettura dei dati pari a 10 kHz . Tuttavia, per ciò che è stato appena affermato, il segnale di Latch sarebbe potuto essere ripetuto anche ogni $50 \mu\text{s}$, dato che i 128 canali sono inviati fuori dalla scheda in $26 \mu\text{s}$, andando, quindi, a raddoppiare la frequenza di lettura dati portandola a 20 kHz . Ciò nonostante, la specifica da rispettare è quella di avere un'acquisizione a 10 kHz , pertanto la struttura dell'acquisizione è rimasta inalterata.

L'ultima cosa da precisare riguardo le prove fatte con l'oscilloscopio, riguarda il fatto che, come si può notare dalle figure precedenti, in quasi tutti i canali in cui non sono state mandate forme d'onda, si nota un singolo impulso azzurro. Quell'impulso simboleggia il livello logico alto del bit meno significativo dei 16; quindi, vuol dire che in quei canali è stato contato 1 dal contatore del subdiagramma "Acquisizione", in quanto ha visto una transizione proveniente dai Citiroc. Questo è un comportamento dovuto, molto probabilmente, al fatto che non è collegato alle linee dei Citiroc alcun SiPM. Non è stato ancora detto, infatti, che le schede CAEN utilizzate sono prive dei

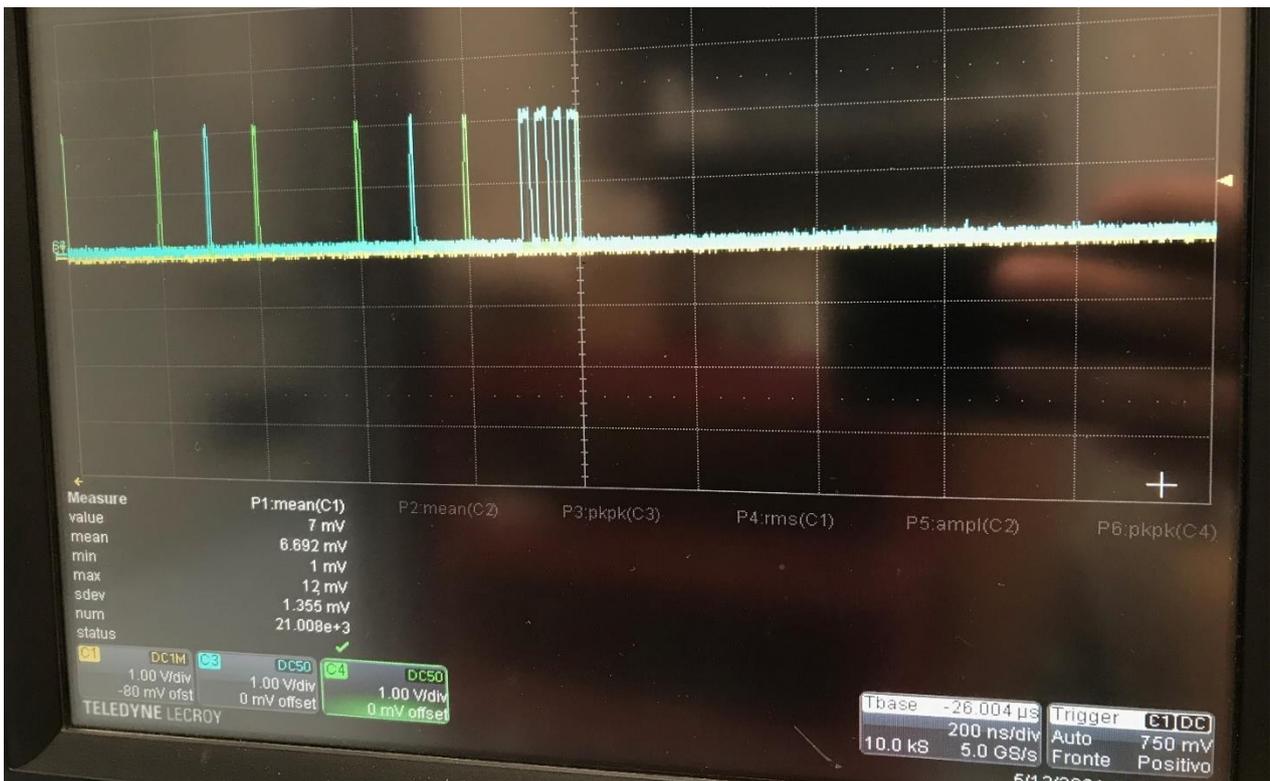


Fig.8.34. Conteggio dell'ultimo canale del Citiroc 4.

fotomoltiplicatori MPPC S13361-3050. Questi verranno inseriti nella scheda una volta terminata la costruzione del monitor da parte del team INFN. Infatti, è stata eseguita una prova inviando il comando ResetD da tastiera, attraverso il “Resource Explorer” di SCI-Compiler con la scheda in esecuzione. Quando si scrive 1 da tastiera, portando il ResetD a livello logico alto, tutti i contatori vengono resettati e non si nota nessun impulso azzurro in nessun canale. Appena viene scritto 0 sul registro che governa il ResetD, i contatori dei sei canali elencati prima ricominciano a contare alle frequenze fissate mentre, nuovamente, la maggior parte dei restanti canali (ma non tutti) contano un impulso iniziale, non vedendo più nulla in seguito. E’ un comportamento ripetitivo che non crea alcun tipo di problema e non influenza il conteggio dei canali che sono in funzione.

DD Low Int

La DD Low Int è una scheda elettronica progettata appositamente per il monitor a basse intensità, la cui funzione è quella di interfaccia tra la scheda CAEN DT5550W / A55CIT4 e il DDS. Questa, infatti, converte i dati forniti dalla scheda CAEN, i quali hanno un formato seriale, in un formato parallelo, in quanto è quello utilizzato dall'acquisizione dati del DDS. Per fare questo, si è deciso di utilizzare uno Shift Register SIPO (Serial In Parallel Output). La scheda si collega con la DT5550W / A55CIT4 attraverso cinque connettori LEMO i quali, attraverso l'uso di cinque cavi LEMO, saranno connessi ai LEMO 2,3,4,5 e 6 della scheda CAEN. Si ricorda che i LEMO 2 e 3 sono programmati come input per ricevere i segnali di Latch e ResetD mentre i LEMO 4,5 e 6 sono programmati come output per trasmettere i segnali Serial, Load e Clock. Quindi, nella DD Low Int, ci saranno due LEMO che invieranno dei segnali in uscita e tre LEMO che riceveranno dei segnali in ingresso. Dall'altra parte, la DD Low Int si collega al DDS attraverso la scheda denominata "Disaccoppiatore Ottico", già esistente nell'elettronica di acquisizione del DDS e in uso con gli attuali monitor del CNAO. La connessione avviene grazie ad un connettore SCSI tipo D femmina a 68 pin, il quale si conetterà allo SCSI a 68 pin maschio presente nel Disaccoppiatore Ottico.

9.1) Shift Register Serial In Parallel Out (SIPO)

Lo Shift Register SIPO è costituito da un certo numero di Flip-Flop di tipo D connessi in cascata, come mostrato in Fig. 9.1. I SIPO si classificano in base al numero di bit, che è uguale al numero di uscite, ovvero al numero di Flip-Flop da cui il sistema è costituito. Nell'esempio in figura si ha un SIPO a 4 bit le cui uscite sono Q_0 , Q_1 , Q_2 , e Q_3 . I Flip-Flop sono sensibili ai fronti di salita o di discesa del clock, ovvero il valore in ingresso al Flip-Flop si presenta in uscita dello stesso in corrispondenza di uno dei due fronti di transizione del clock. Per capire il funzionamento del SIPO viene illustrato un esempio con la parola binaria 1001. Inizialmente le quattro uscite sono a 0. In corrispondenza del primo fronte di clock l'1 va in uscita su Q_0 mentre le altre tre uscite successive rimangono sempre a 0. Si nota dalla figura che il clock è collegato a tutti i Flip-Flop, quindi, al fronte di clock successivo, l'1 che si aveva in uscita su Q_0 passa su Q_1 ; il secondo bit della parola seriale in ingresso, ossia 0, va in uscita su Q_0 . Al fronte successivo si avrà 0 su Q_0 , 0 su Q_1 e 1 su Q_2 . Infine, al quarto fronte di clock l'1 passa in uscita su Q_0 con relativo scorrimento degli altri tre bit come visto finora. Si ha in uscita la parola binaria 1001 in forma parallela, ottenuta dopo un

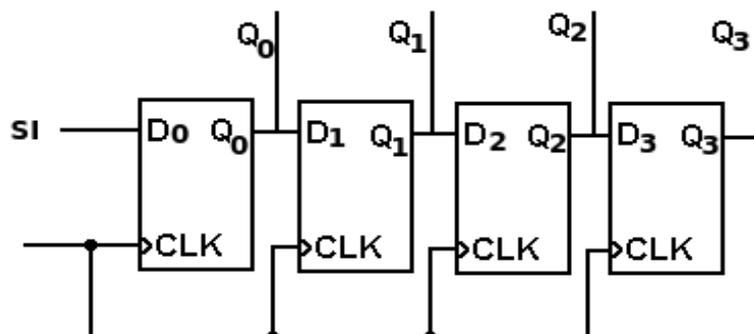


Fig.9.1. SIPO.

tempo, rispetto al primo bit della parola seriale in ingresso al primo Flip-Flop, pari a quattro volte il periodo del clock.

Nella DD Low Int è stato scelto come SIPO il 74LVC594A appartenente alla famiglia logica CMOS. Questo è costituito da uno Shift Register e uno Storage Register, entrambi ad 8 bit. Tra i vari SIPO considerati in fase di progettazione, si è deciso di scegliere questo, in quanto era l'unico capace di soddisfare le condizioni relative al parametro V_{IH} dei pin d'ingresso, mantenendo una frequenza di lavoro molto alta. Dal datasheet, infatti, risulta che quel circuito integrato alimentato a 3,3 V ha un valore di V_{IH} pari a 2 V. I segnali in uscita dalla scheda CAEN, esprimenti un livello logico alto, sono fatti arrivare a quei pin con un valore di tensione di 2,2 V. La frequenza di lavoro necessaria è di 160 MHz e, dal datasheet, si è osservato che con una tensione di alimentazione di 3,3 V, la frequenza di lavoro massima dell'integrato è di 180 MHz. Il 74LVC594A ha ingressi di clock e reset separati per lo Shift Register e Storage Register. Il dispositivo dispone di un ingresso seriale denominato DS. I bit vengono spostati tra i Flip-Flop in corrispondenza delle transizioni da 0 ad 1 portate in ingresso al pin SHCP (ossia il clock dello Shift Register) e i bit dallo Shift Register vengono trasferiti allo Storage Register in corrispondenza della transizione da 0 a 1 del segnale in ingresso al pin STCP (ossia il clock dello Storage Register). Un livello logico basso su uno dei due pin di reset del registro (\overline{SHR} e \overline{STR}) azzerava l'uscita del registro corrispondente. In Fig. 9.2 viene mostrato il diagramma funzionale del 74LVC594A. Gli 8 bit arrivano in serie in ingresso al pin DS. In quel pin verrà collegato il segnale Serial in uscita dalla scheda CAEN. In SHCP sarà collegato il segnale Clock a 160 MHz uscente sempre dalla scheda. \overline{SHR} viene collegato sempre a livello logico alto, in quanto non è necessario andare a resettare lo Shift Register. Q7S è un pin che dà in uscita i bit ricevuti in DS in formato seriale e serve per permettere il collegamento a cascata con un altro SIPO; questa operazione verrà eseguita in quanto il 74LVC594A è un SIPO ad 8 bit. Volendo parallelizzare un dato a 16 bit occorre collegare due SIPO ad 8 bit in cascata. Si nota, sempre dalla

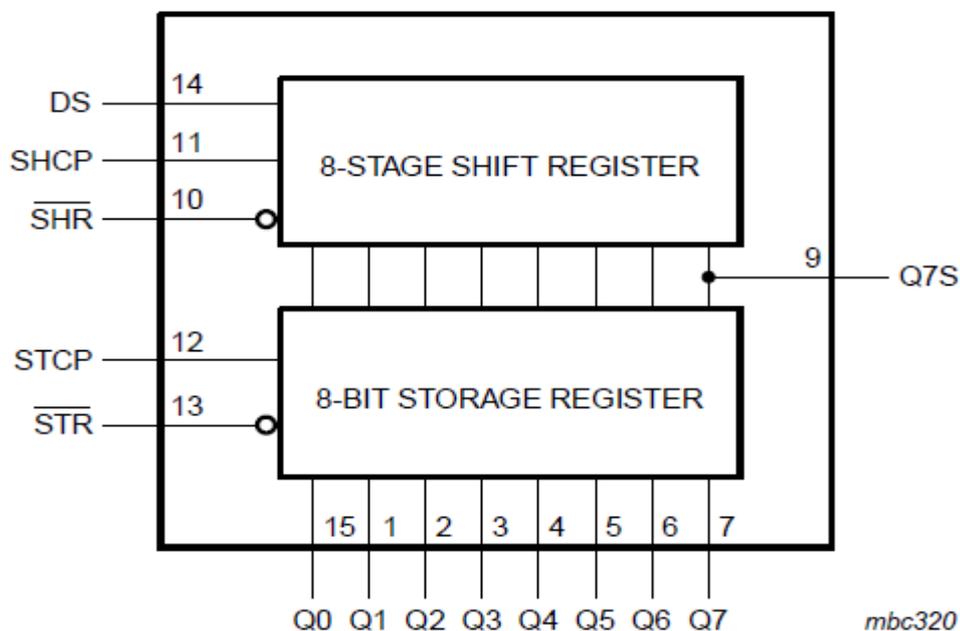


Fig.9.2. Diagramma funzionale [15].

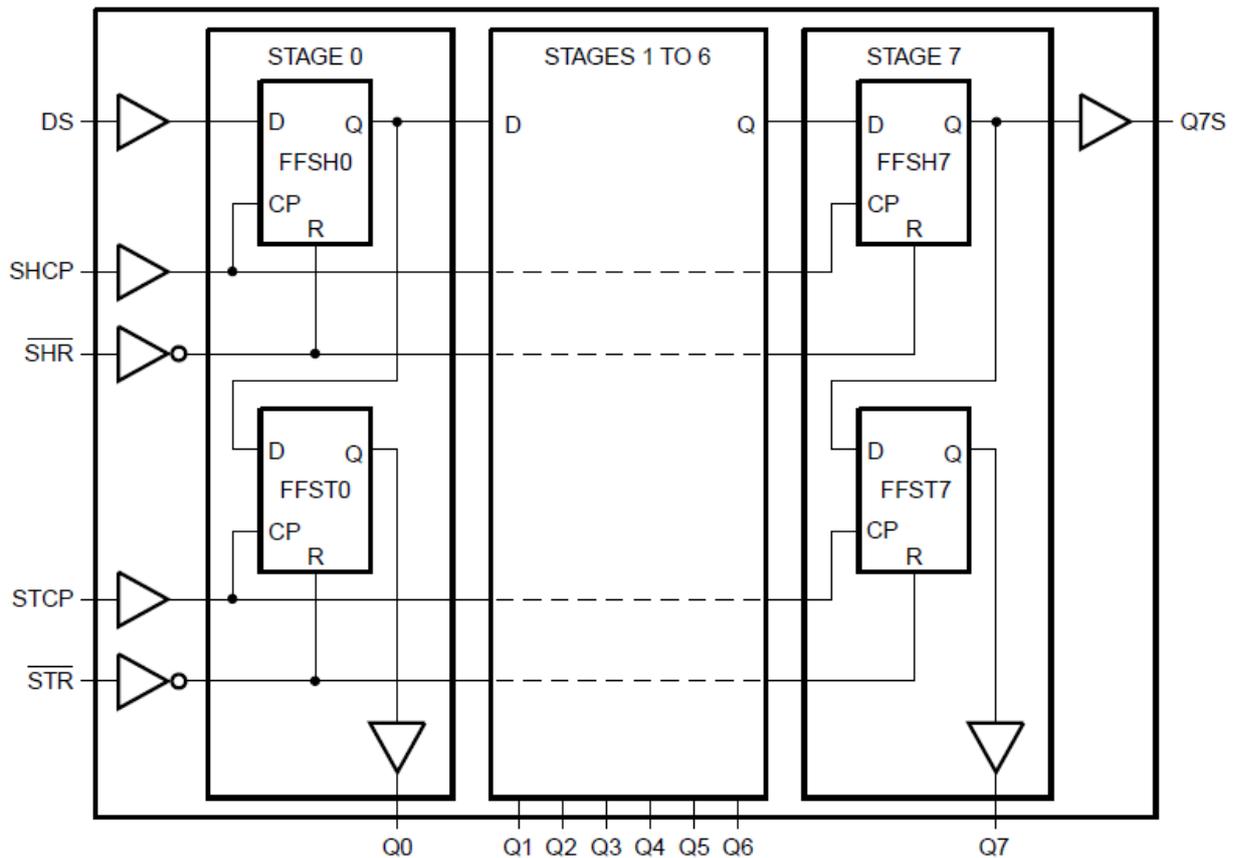


Fig.9.3. Diagramma logico [15].

figura, che lo Shift Register collega le sue uscite allo Storage Register. Questo ha il pin \overline{STR} che, come \overline{SHR} , viene lasciato a livello logico alto. STCP è il pin che funge da segnale di clock per lo Storage Register. In questo pin verrà collegato il segnale Load in uscita dalla scheda CAEN. In Fig. 9.3 è illustrato il diagramma logico del funzionamento del 74LVC594A. La parte superiore rappresenta lo Shift Register mentre quella inferiore lo Storage Register. Dal segnale Serial arrivano 16 bit in 100 ns che devono essere portati nelle uscite di due 74LVC594A collegati in cascata. Il collegamento in cascata avviene collegando l'uscita del pin Q7S del primo SIPO con il pin d'ingresso DS del secondo SIPO. Inoltre, il segnale Clock deve essere portato in ingresso nel pin SHCP di entrambi i SIPO, così come il segnale Load deve essere collegato in entrambi i pin STCP. Tutti i pin di Reset dei due circuiti integrati vengono collegati alla tensione di alimentazione per assegnare loro il livello logico alto. I bit in ingresso al primo SIPO vengono fatti scorrere fino all'ottavo Flip-Flop dello Shift Register, in seguito ad otto colpi di clock. Questi 8 bit escono da Q7S per entrare nel secondo SIPO e, negli 8 colpi di clock successivi, nel primo SIPO arrivano gli ultimi 8 bit dei 16, che scorrono fino all'ultimo Flip-Flop, mentre, nel secondo SIPO, gli 8 bit che erano stati portati in ingresso tramite Q7S sono fatti scorrere fino all'ultimo Flip-Flop. Tutto questo avviene contemporaneamente nei due SIPO avendo collegato nel loro pin SHCP lo stesso segnale di clock. A questo punto sono passati 100 ns, ovvero 16 colpi di clock. Come è stato descritto nell'implementazione del firmware della scheda CAEN, il colpo di clock successivo all'ultimo dei 16 bit serializzati corrisponde all'arrivo del segnale Load. Il segnale Load corrisponde al segnale di clock dello Storage Register il quale, ricevendo questo impulso, mostra in uscita i bit che gli sono stati portati in ingresso dalle uscite dei Flip-Flop dello Shift Register. Il processo di parallelizzazione dei dati è completato e i 16 bit rimangono in uscita nei pin dei due integrati per

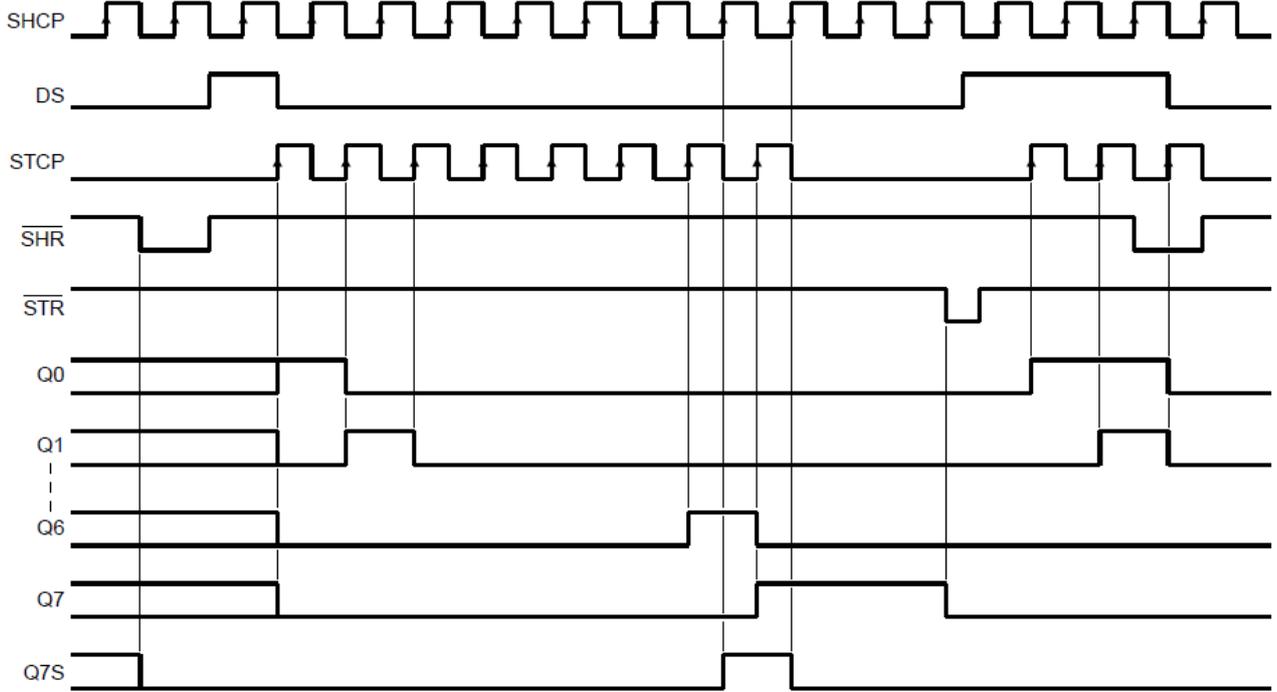


Fig.9.4. Diagramma temporale [15].

200 ns, ossia per il tempo equivalente al periodo del segnale Load. Dopo 200 ns, in corrispondenza del Load successivo, verranno mostrati in uscita dei nuovi dati, corrispondenti al canale successivo. In Fig. 9.4 è rappresentato lo schema temporale del funzionamento del SIPO appena descritto.

9.2) Schema elettrico

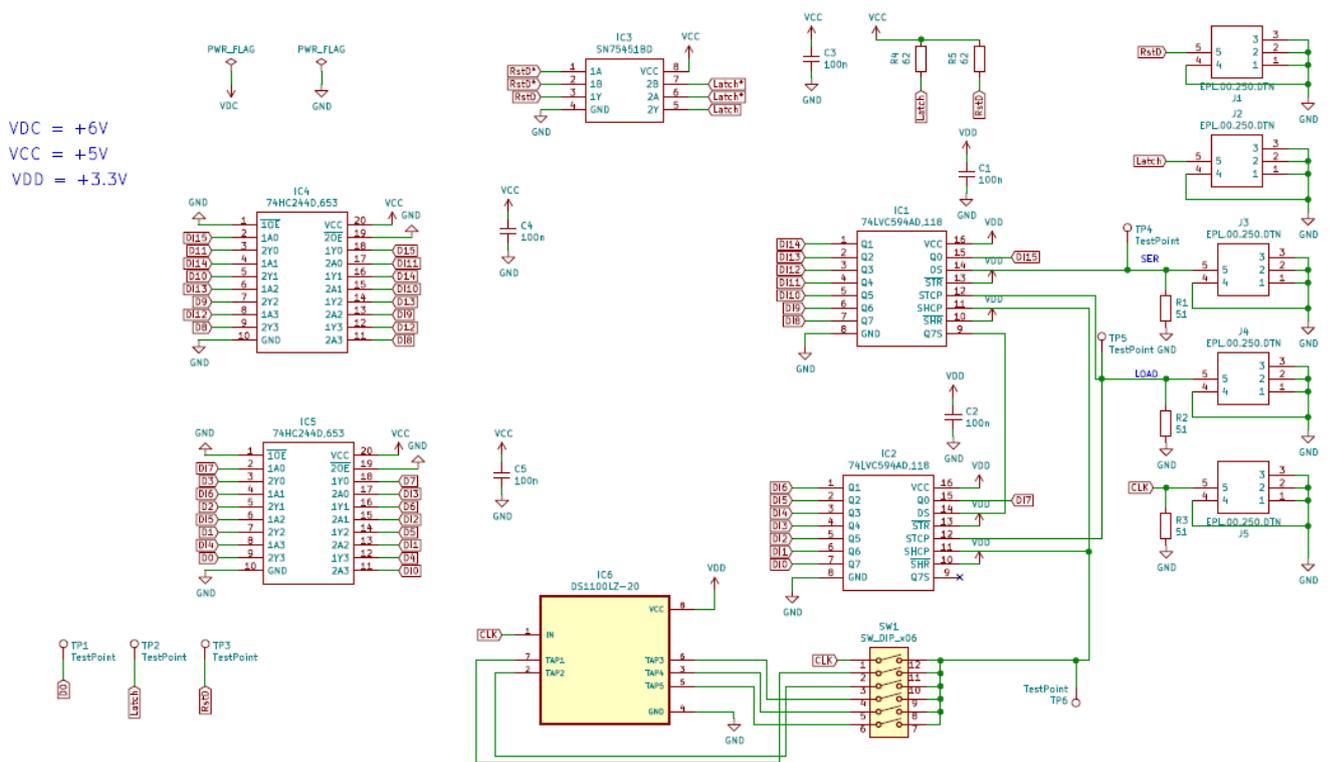


Fig.9.5. Schema elettrico (Foglio 1).

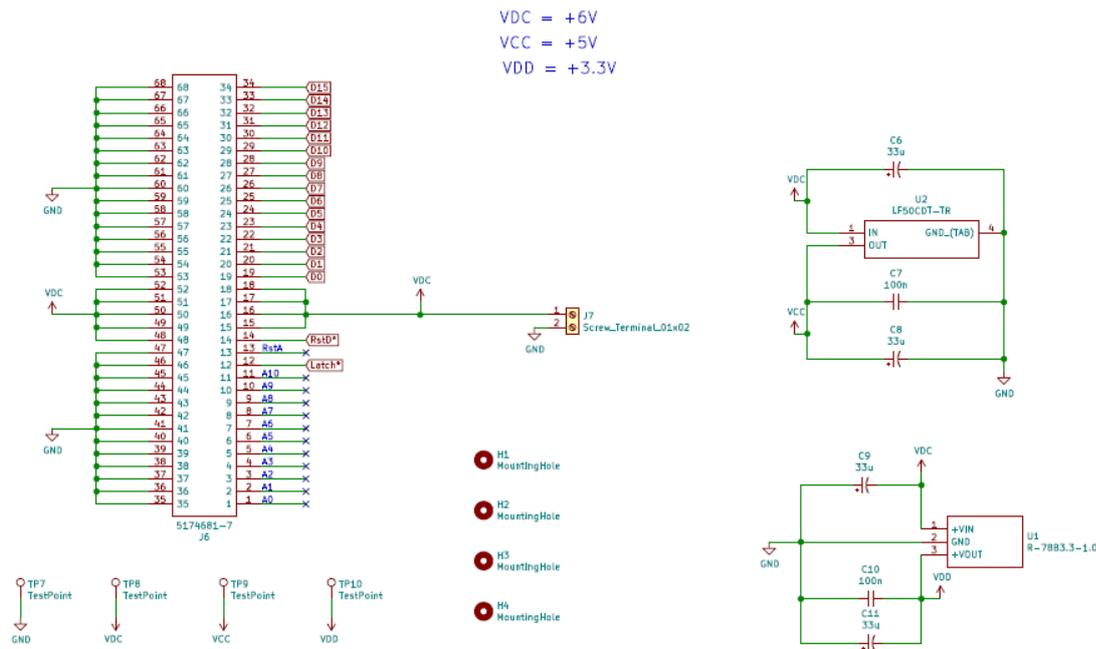


Fig.9.6. Schema elettrico (Foglio 2).

Nelle Fig. 9.5 e 9.6 è mostrato lo schema elettrico della DD Low Int, suddiviso in due fogli di lavoro realizzati con il software KiCad, specifico per il disegno di schemi elettrici e circuiti stampati. Questo schema verrà, di seguito, spiegato separando varie parti del circuito per una migliore comprensione dello stesso e visualizzazione dello schema.

La prima parte che viene descritta è raffigurata in Fig. 9.7. IC1 e IC2 sono i due 74LVC594A che di sono già stati commentati nel paragrafo precedente. Si può vedere che in questi due integrati, ma anche in altri componenti del circuito, alcuni pin non si collegano ad altri per mezzo del filo verde, ma terminano con una specie di etichetta. Questa serve proprio per evitare il collegamento con il filo ad un altro pin in modo tale che lo schema sia più ordinato. Nel caso di IC1 e IC2 tutti i pin di uscita che rappresentano i bit parallelizzati hanno un'etichetta e sono chiamati DIx (con x che va da 0 a 15). Per quanto riguarda i collegamenti nei pin di ingresso degli integrati, questi sono già stati esposti nel paragrafo precedente. Il pin di alimentazione è legato a V_{DD} , che nello schema indica una tensione pari a 3,3 V. Inoltre, sono presente i due condensatori di by-pass C1 e C2 da 100 nF collegati tra V_{DD} e ground. Sulla destra della figura è possibile osservare, con una disposizione in colonna, alcuni dei cinque LEMO. Quelli da cui arrivano i segnali che vanno in ingresso ai SIPO sono J3 (Serial), J4 (Load) e J5 (Clock). Il nome tecnico è EPL.00.250.DTN, ossia connettori LEMO femmina coassiali con impedenza caratteristica di 50 Ω . Sulla linea che porta il segnale dal connettore ai pin interessati sono collegati tre test point (TP4, TP5 e TP6). Questi sono utili in fase di test della scheda, quando questa verrà ultimata, per verificare la presenza e correttezza morfologica dei segnali in quelle linee, con l'ausilio di una sonda e di un oscilloscopio. Inoltre, in questi tre LEMO è possibile osservare la presenza di tre resistori di terminazione per l'adattamento delle porte logiche dei pin dei SIPO alla linea di trasmissione utilizzata, la quale ha un'impedenza caratteristica, come già detto, di 50 Ω . Nel caso in questione sono stati utilizzati tre resistori SMD da 51 Ω , (per motivi di tipo commerciale non è stato possibile acquistare resistori esattamente da 50 Ω) i quali soddisfano a pieno l'esigenza. E' stato verificato che, utilizzando dei resistori di

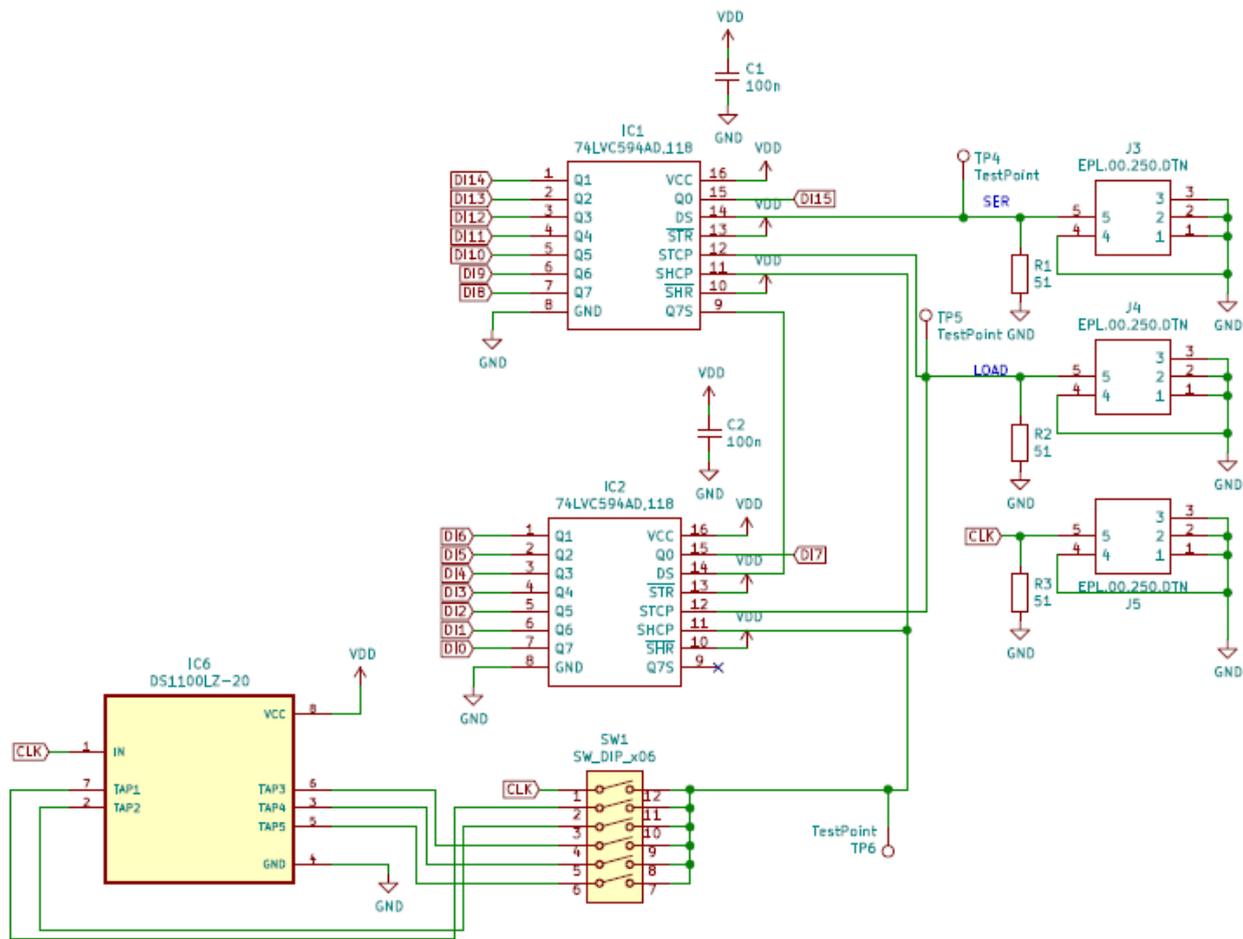


Fig.9.7. Parte di circuito comprendente i due SIPO, tre connettori LEMO e la rete di ritardo.

terminazione della linea di questo valore, si ottiene un valore di ampiezza delle onde quadre, in ingresso alla DD Low Int, di 2,2 V.

Si può notare come la linea in uscita da J5 non vada direttamente nei pin SHCP degli integrati IC1 e IC2. L'uscita termina con un'etichetta denominata "CLK" che si ritrova in ingresso al circuito integrato IC6. L'insieme costituito da IC6 e SW1 costituiscono una rete di ritardo per il segnale Clock proveniente da J5, dove viene preso in ingresso il segnale e riportato in uscita identico a come era in ingresso ma ritardato di una certa quantità temporale. Il motivo della presenza della rete di ritardo è il seguente: come è stato detto nella spiegazione del funzionamento del 74LVC594A, il singolo bit in ingresso al pin DS viene portato in uscita dal Flip-Flop e, quindi, fatto scorrere a quello successivo, in corrispondenza del fronte di salita del segnale di clock. Quando è stato testato il funzionamento del firmware SCI-Compiler, durante la visualizzazione sull'oscilloscopio dei tre segnali che dalla scheda CAEN entrano nella DD Low Int, si è osservato che il clock da 160 MHz aveva il proprio fronte di salita in corrispondenza, all'incirca, della metà dell'intervallo di 6,25 ns che costituisce ogni bit. Questa è la situazione che si vuole avere in quanto, se il fronte di salita fosse più spostato, ad esempio in corrispondenza delle transizioni dei bit trasmessi, potrebbero essere trasferiti valori dei bit sbagliati. Tuttavia, nonostante la situazione sia quella desiderata, è stata implementata questa rete di ritardo che agisce sul segnale di clock per avere una maggiore sicurezza. Il segnale, infatti, potrebbe subire eventuali ritardi nel passaggio da una scheda all'altra tramite i cavi, oppure per altre ragioni ignote.

Il nome tecnico di IC6 è DS1100LZ-20. E' un circuito integrato che riceve un segnale d'ingresso e lo fornisce in uscita su 5 pin con ritardi diversi tra loro, grazie alle linee di ritardo presenti internamente ad esso, con un ritardo che va da 4 ns a 20 ns, con un passo di 4 ns. Il circuito è alimentato a 3,3 V. SW1 è un interruttore Dip Switch a 6 pin. Come si vede dallo schema elettrico, in ingresso ai 6 pin dell'interruttore sono collegati le uscite di IC6 e il segnale di clock non ritardato nel pin 1. In questo modo, se non fosse necessario introdurre un ritardo al segnale, ad esempio se venisse verificato che i SIPO funzionano correttamente con il clock non ritardato, allora basterebbe chiudere l'interruttore della linea 1 di SW1 per dare in ingresso ai due SIPO il clock non ritardato. Al contrario, se fosse necessario inserire un ritardo al segnale, si andrebbe a chiudere una delle cinque linee di SW1 in base al ritardo che si vuole dare al segnale, considerando quelli messi a disposizione da IC6.

I segnali in uscita dai 16 pin dei due SIPO vengono mandati in ingresso a due driver, precisamente il 74HC244D (IC4 e IC5 in Fig. 9.8). Questo è un driver ad 8 bit con buffer tri-state sulle 8 linee. In generale, un'uscita tri-state è un'uscita che può assumere i livelli logici 0 e 1 e, in più, un terzo stato, detto ad alta impedenza. In quest'ultimo caso, l'uscita si comporta come se fosse elettricamente disconnessa dal resto del circuito, ovvero sull'uscita non è presente un valore di tensione preciso, ma la tensione dipende dal circuito al quale l'uscita stessa è collegata.

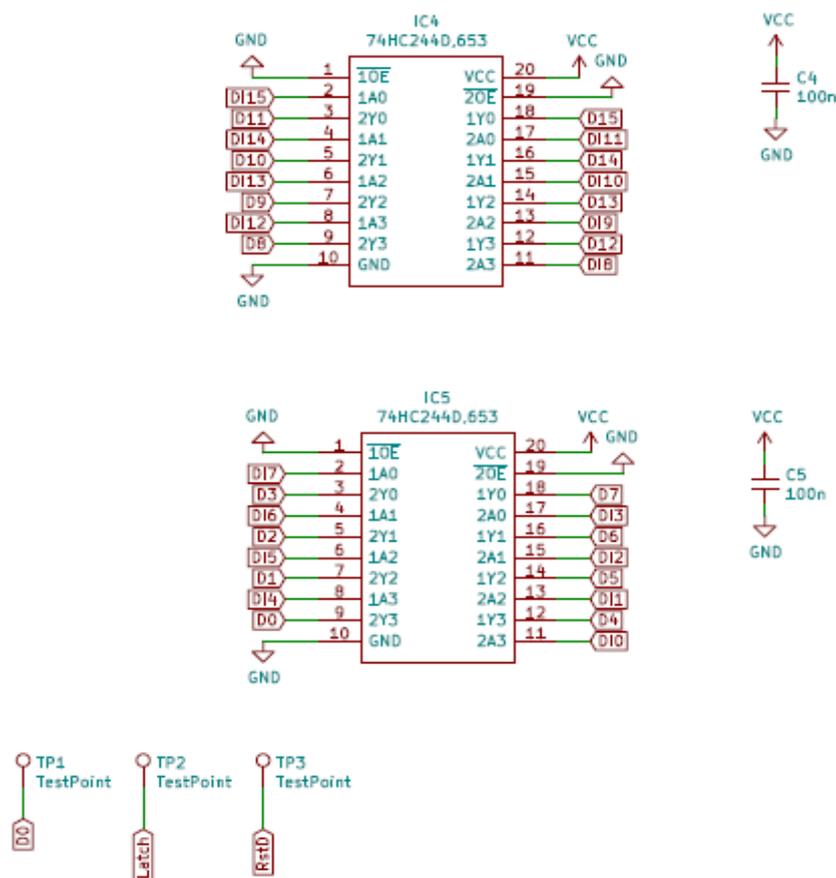


Fig.9.8. Parte di circuito comprendente i due driver 74HC244D.

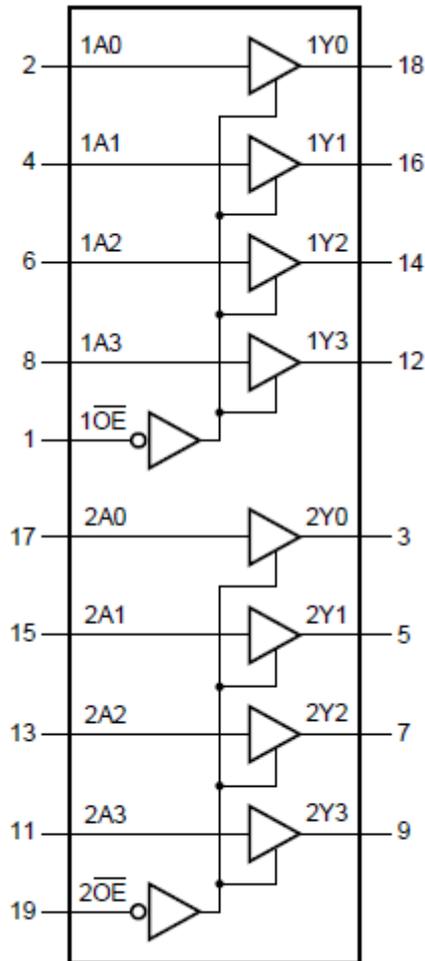


Fig.9.9. Diagramma funzionale 74HC244D [16].

Attraverso i pin $\overline{1OE}$ e $\overline{2OE}$ è possibile abilitare lo stato ad alta impedenza fornendogli un livello logico alto in ingresso. In Fig. 9.9 è raffigurato il diagramma funzionale del 74HC244D. E' costituito da 8 linee nelle quali è presente un buffer tri-state. Lo scopo principale del buffer è quello di separare o disaccoppiare la sorgente del segnale dal resto del circuito. Nel buffer la tensione di uscita segue la tensione di ingresso, ma, dato che la sua impedenza di ingresso è alta e quella di uscita è bassa, questa configurazione permette un adattamento di impedenza lungo la linea riducendone il valore nel collegamento tra circuiti. Quindi, inserire un driver nel circuito, è utile per "rafforzare" il segnale; in questo caso i segnali che passano attraverso i due driver sono i 16 bit di dati spediti ogni 200 ns. La scelta di questo tipo di driver è data da due motivazioni:

- i 16 bit verranno spediti, attraverso il connettore SCSI a 68 pin, al Disaccoppiatore Ottico. L'integrato che li riceverà in ingresso si aspetterà dei segnali in logica TTL a 5 V. Questo non sarebbe stato possibile mandando direttamente le uscite degli integrati IC1 e IC2, in quanto i due SIPO sono alimentati a 3,3 V. Il 74HC244D, invece, viene alimentato alla tensione $V_{CC} = 5 V$ nel circuito e fornisce un'uscita con quelle caratteristiche.
- La DD Low Int, collegata con la scheda CAEN DT5550W / A55CIT4, nel loro insieme, vanno a sostituire, nel monitor per basse intensità, quello che era il Tera06 per il monitor delle camere strip e pixel o il Tera07 per la camera integrale. Nella scheda contenente questi

chip, la quale si collega al Disaccoppiatore Ottico, il 74HC244D era utilizzato per il trasferimento dei segnali, per cui è una buona idea utilizzare, anche in questo caso, quello stesso stadio di uscita verso il Disaccoppiatore Ottico.

In Fig. 9.8 è possibile distinguere gli ingressi, indicati con la nomenclatura DIx (con x che varia da 0 a 15) dalle uscite indicate con la nomenclatura Dx (con x che varia da 0 a 15). I pin di abilitazione dello stato ad alta impedenza sono collegati a ground, in quanto quello stato non è da utilizzare nel caso in questione. Si possono notare i due condensatori di by-pass da 100 nF (C4 e C5) e altri tre test point (TP1, TP2 e TP3). TP1 viene messo sulla linea che collega il bit meno significativo dei 16 in uscita dal 74HC244D al connettore SCSI. In questo modo sarà possibile verificare sull'oscilloscopio la corretta parallelizzazione dei bit su quella linea. TP2 e TP3 sono messi sulle piste del circuito stampato dove passano i segnali di Latch e ResetD.

In Fig. 9.10 viene illustrata la parte di schema elettrico dedicata al trasferimento dei segnali di Latch e ResetD dal Disaccoppiatore Ottico alla scheda CAEN. IC3 è un driver nel quale arrivano in ingresso i due segnali. Il circuito integrato scelto è l'SN75451B (famiglia logica TTL). Questo driver è diverso dal precedente, in quanto quello aveva un'uscita tri-state mentre il driver corrente ha un'uscita open-collector. Lo schema logico dell'SN75451B è mostrato in Fig. 9.11. Il segnale che deve passare attraverso il driver viene mandato in input in entrambi i pin che costituiscono i due ingressi della porta logica in figura; nel caso dell'SN75451B si ha una porta logica NAND. Il funzionamento è il seguente: quando nei due pin d'ingresso della NAND arrivano due 0 logici, l'output della porta è 1. Avere una tensione alta su quell'uscita, vuol dire avere una tensione V_{BE} del BJT alta; il transistor è in saturazione. In questa situazione l'uscita dell'open collector è 0. Quando in ingresso alla porta NAND arrivano due 1 logici, l'output della porta è 0. Se $V_{BE} \approx 0$ il BJT è in interdizione e l'uscita dell'open collector è 1. Il BJT funge, praticamente, da porta NOT alla tensione in uscita dalla porta NAND. Per questo, in realtà, l'SN75451B, come si può notare anche dal datasheet, implementa una logica AND. Il collettore del BJT costituisce il pin di uscita dell'integrato; essendo una porta open-collector questa uscita dovrà essere collegata alla tensione di alimentazione tramite un resistore di pull-up all'esterno del circuito integrato, come visto nel Cap. 2. Questo driver presenta due linee del tipo appena descritto e risulta adatto, quindi, alla situazione attuale in cui si hanno esattamente due segnali, ovvero Latch e ResetD. In Fig. 9.10 si nota che gli ingressi sono caratterizzati dal nome del segnale d'ingresso seguito da un asterisco mentre le uscite hanno il nome del segnale normale. Si possono vedere, inoltre, le uscite collegate alla tensione di

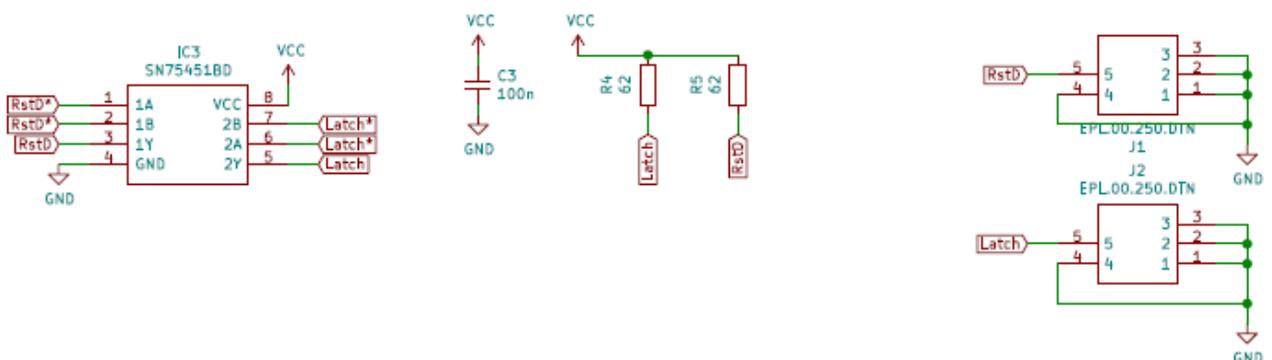


Fig.9.10. Parte di circuito comprendente il driver SN75451B, i resistori di pull-up e due connettori LEMO.

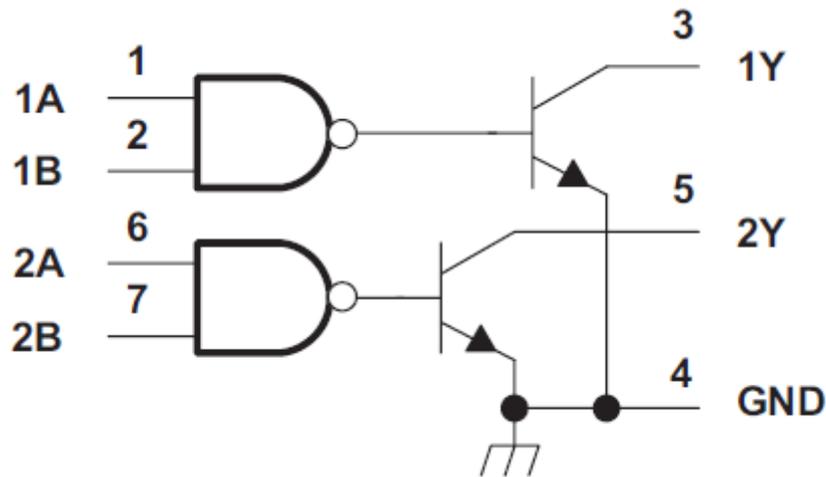


Fig.9.11. Diagramma logico SN75451B [17].

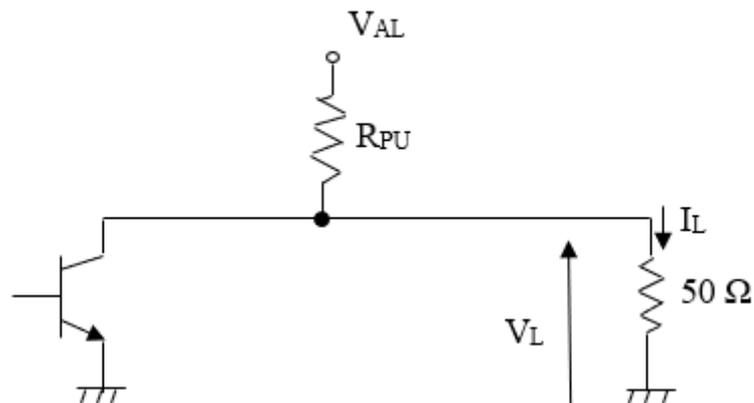


Fig.9.12. Circuito per il dimensionamento di R_{PU} sui pin d'uscita dell'SN75451B.

alimentazione V_{CC} (5 V) attraverso due resistor di pull-up da 62 Ω . Di seguito è mostrato il calcolo per il dimensionamento del resistore.

Si consideri come carico da pilotare il resistore di terminazione del valore di 50 Ω . Questo non si trova nel presente circuito, ma nella DT5550W / A55CIT4 grazie all'inserimento dei due jumper dietro i due connettori LEMO 2 e 3 della scheda, che sono programmati come input, pronti a ricevere i segnali Latch e ResetD in uscita dalla DD Low Int, tramite i due LEMO J1 e J2. Il circuito da considerare per il dimensionamento del resistore di pull-up è quello in Fig. 9.12. Questo è costituito dall'uscita open-collector di IC3, R_{PU} collegato alla tensione di alimentazione (pari a 5 V) e dal resistore di terminazione di 50 Ω . Come è stato illustrato nel Par. 2.2.1, bisogna considerare, per il dimensionamento, i due casi in cui il BJT è in interdizione e saturazione. Innanzitutto, bisogna considerare il fatto che la tensione V_L deve essere pari a 2,2 V, essendo questo il valore di tensione, utilizzato anche per i connettori LEMO di output della scheda CAEN, esprime il livello logico alto. Quindi, la corrente assorbita dal carico sarà:

$$I_L = \frac{V_L}{50\Omega} = \frac{2,2V}{50\Omega} = 44 \text{ mA}.$$

A questo punto è possibile calcolare l'estremo superiore dell'intervallo di valori che può assumere R_{PU} . Si considera la condizione di interdizione del transistor. La formula da applicare in questo caso (esposta ampiamente nel Par. 2.2.1) è la seguente

$$R_{PU} \leq \frac{V_{AL} - V_L}{I_L} = \frac{5 - 2,2}{0,044} = 63,63 \Omega$$

Per la condizione di saturazione del transistor si deve considerare il valore massimo di corrente che può scorrere nel collettore. Osservando il datasheet di IC3 si può notare che il valore di questa corrente è di 400 mA. Per cui, il valore dell'estremo inferiore dell'intervallo di valori per R_{PU} è

$$R_{PU} \geq \frac{V_{AL}}{I_C} = \frac{5}{0,4} = 12,5 \Omega$$

Si è scelto di attribuire a R_{PU} il valore di 62 Ω , ottenendo un livello di ampiezza della tensione che entra nei LEMO della DT5550W / A55CIT4 di circa 2,2 V. I resistori scelti, inoltre, sono resistori SMD (Surface Mounting Device), ossia a montaggio superficiale. Le dimensioni di questi tipi di resistori variano a seconda della potenza che riescono a dissipare; più sono grandi e maggiore è la potenza che riescono a dissipare. In questo caso è stato scelto un resistore SMD 1210. La sigla numerica indica una dimensione di 12 millesimi di pollice per 10 millesimi di pollice, ovvero 3,2 mm x 2,5 mm. Un resistore di questa dimensione, con un valore della resistenza di 62 Ω , ha una potenza massima dissipabile di 0,5 W. E' necessario verificare se, nel circuito già considerato nel dimensionamento di R_{PU} , quest'ultimo possa dissipare più di tale potenza.

Nel caso di interdizione, in cui l'uscita della porta logica è a livello logico alto, si considera il circuito in Fig. 9.13. La corrente che scorre nella maglia è la seguente

$$I = \frac{V_{AL}}{R_{PU} + 50} = \frac{5}{112} \approx 44 \text{ mA}$$

La potenza dissipata sul resistore di pull-up viene calcolata come

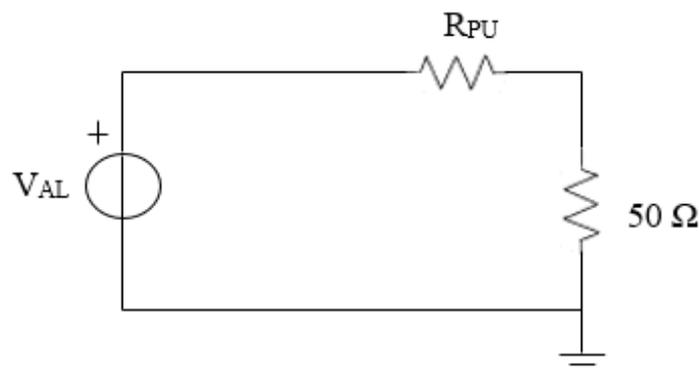


Fig.9.13. Circuito da considerare per il calcolo della potenza dissipata sul resistore di pull-up in condizione di interdizione del BJT della porta open-collector.

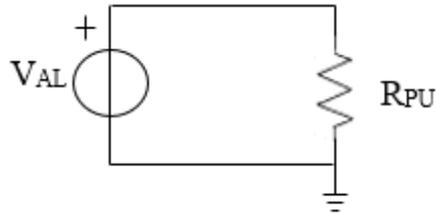


Fig.9.14. Circuito da considerare per il calcolo della potenza dissipata sul resistore di pull-up in condizione di saturazione del BJT della porta open-collector.

$$P_{R_{PU}} = R_{PU} I^2 \approx 0,12 \text{ W}$$

Quando il transistor lavora in interdizione la potenza dissipata su R_{PU} è minore della potenza massima dissipabile.

Nel caso di saturazione, in cui l'uscita della porta logica è a livello logico basso, si considera il circuito in Fig. 9.14. E' un circuito molto semplice, così come il calcolo per determinare la corrente che scorre nella maglia

$$I = \frac{V_{AL}}{R_{PU}} \approx 80 \text{ mA}$$

La potenza dissipata su R_{PU} vale

$$P_{R_{PU}} = R_{PU} I^2 \approx 0,4 \text{ W}$$

Anche quando il transistor è in saturazione non viene superato il limite di potenza dissipabile dato dal resistore SMD che si vuole utilizzare. Per cui si è giunti alla conclusione che è possibile utilizzare un resistore SMD 1210 da 62Ω da collegare all'uscita open collector dell'SN75451B.

Si notano, in Fig. 9.10, i suddetti resistori (R_4 e R_5) che collegano i due pin di uscita dell'SN75451B, dai quali fuoriescono i segnali di Latch e ResetD, alla tensione di alimentazione. C_3 è il condensatore di by.pass da 100 nF per il circuito integrato IC3. Infine, nella parte più a destra della Fig. 9.10, si riconoscono i due LEMO di uscita della DD Low Int. Tramite due cavi LEMO, questi si collegano ai connettori LEMO 2 e 3 della DT5550W / A55CIT4, i quali presentano la terminazione a 50Ω (come già riferito precedentemente) per la trasmissione del segnale, grazie ai due jumper inseriti dietro i due connettori.

Nella Fig. 9.15 è rappresentata la parte di schema elettrico in cui il componente principale è il connettore SCSI di tipo D femmina a 68 pin, rappresentato con la sigla J6 nello schema elettrico. Questo componente permette il collegamento della scheda DD Low Int con il Disaccoppiatore Ottico, che presenta un connettore SCSI di tipo D maschio a 68 pin, attraverso un cavo flat (Fig. 9.16). La parte laterale sinistra di J6, che comprende l'intervallo di pin 35-68, è costituita dai vari collegamenti al ground e alla tensione V_{DC} , la quale ha il valore di 6 V . Quest'ultima rappresenta la tensione di alimentazione dell'intero circuito che viene ricevuta dal Disaccoppiatore Ottico attraverso il connettore SCSI. Nella parte laterale destra di J6 vi sono i rimanenti collegamenti dei pin da 1 a 34. Quattro di questi sono ancora pin che si collegano alla tensione di alimentazione V_{DC} . Sopra questi quattro, dal pin 19 al pin 34, vi sono le etichette che si avevano in uscita dai pin di

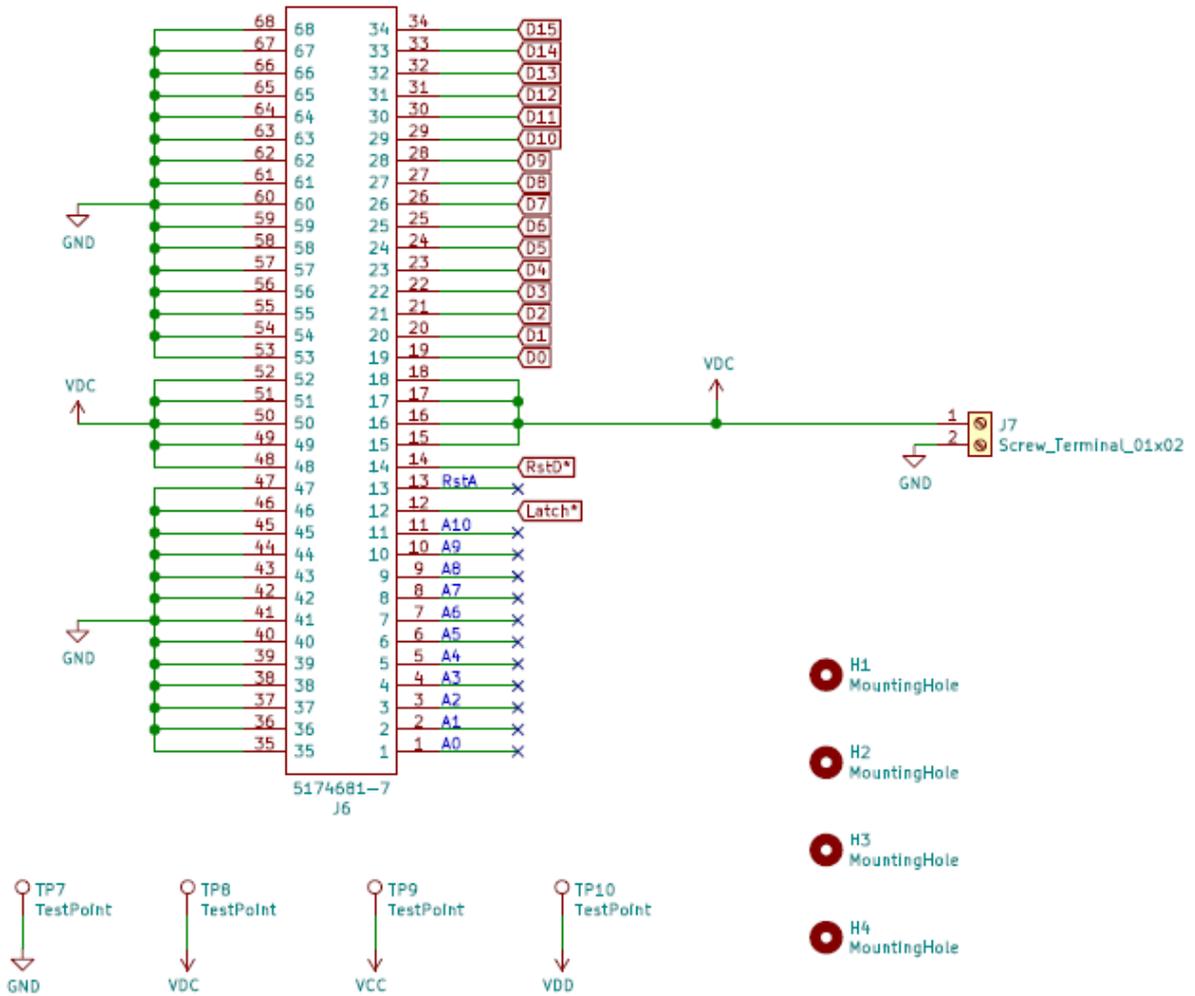


Fig.9.15. Parte di circuito comprendente il connettore SCSIID a 68 pin.



Fig.9.16. Cavo flat.

output di IC4 e IC5. Rappresentano i 16 bit parallelizzati dai SIPO che vengono portati, tramite il connettore SCSI, al Disaccoppiatore Ottico e, quindi, al DDS. Nei pin 12 e 14 si trovano, invece, i segnali di Latch e ResetD, rappresentati nelle etichette dal loro nome seguito dall'asterisco. Questi entrano nei pin d'ingresso di IC3. I due segnali provengono dal DDS passando per il Disaccoppiatore Ottico. Infine, vi sono una serie di pin che non hanno alcun collegamento e sono caratterizzati da una x blu alla fine. Questi sono pin del connettore non utilizzati che, però, hanno dei nomi lungo le linee, in quanto erano utilizzati nella configurazione elettronica per le camere strip, pixel e integrali. Il pin "ResetA" indica il comando digitale utilizzato nelle schede Tera per scaricare il condensatore di integrazione dell'OTA. Non essendo utilizzato il convertitore corrente-frequenza nel monitor per basse intensità, questo comando non è più necessario. Per quanto riguarda i pin da 1 a 11 il discorso è più complicato. Quei pin, nominati come Ax (con x che va da 0 a 10), erano utilizzati sempre nella configurazione per le camere strip, pixel e integrali, e servivano per l'indirizzamento dei canali di acquisizione dei chip Tera. In questo modo, l'ordine dei canali da acquisire era deciso dal DDS. Questo sarebbe stato possibile anche nella configurazione elettronica del monitor per basse intensità. Tuttavia, per renderlo possibile, si sarebbe dovuto aggiungere un PISO (Parallel In Serial Out) nella DD Low Int per serializzare i bit Ax necessari per l'indirizzamento dei 128 canali Citiroc, in modo che questi potessero entrare nella DT5550W / A55CIT4 attraverso un LEMO. Si è visto come, utilizzando i due SIPO per la parallelizzazione dei dati a 16 bit, è stato necessario usufruire di tre connettori LEMO della scheda CAEN. Avendo a disposizione solamente otto connettori LEMO, e sapendo che due di questi dovevano essere lasciati per il trasferimento di Latch e ResetD, non si aveva la sicurezza che i rimanenti sei LEMO sarebbero stati abbastanza per la parallelizzazione dei dati a 16 bit e per la serializzazione dei bit di indirizzo. Per tale motivo, in fase iniziale di progettazione (ancor prima della scrittura del firmware), è stata presa la decisione di non utilizzare tali bit di indirizzo, e, quindi, non lasciare il controllo dell'indirizzamento dei 128 canali Citiroc al DDS. Come si è visto nel Par. 8.5, nella descrizione del subdiagram "SelIndirizzi" del programma SCI-Compiler, gli indirizzi vengono generati a livello firmware grazie all'impiego di un contatore. In questo modo, però, si è vincolati ad acquisire sempre i canali in ordine crescente da 0 a 127.

Il componente J7 è un terminale a vite a 2 pin. E' utile per effettuare dei test sulla scheda e viene utilizzato per trattenere un filo tramite un serraggio a vite. I suoi due pin sono collegati a V_{DC} e a ground. Dato che, come è stato detto, la scheda prende l'alimentazione tramite il connettore SCSI dal Disaccoppiatore Ottico, il terminale a vite serve per andarla ad alimentare senza connetterla al Disaccoppiatore. Utilizzando un alimentatore la cui tensione viene impostata a 6 V, si usano due fili per collegare il potenziale a 6 V e quello a ground dell'alimentatore al terminale a vite, il quale ha i suoi due pin collegati uno alla pista che porta i 6 V e uno al piano di ground. In questo modo il circuito risulta correttamente alimentato e sarà possibile effettuare dei test sulla scheda una volta montata. Vi sono, infine, gli ultimi quattro test point della scheda (TP7, TP8, TP9 e TP10) che si troveranno in corrispondenza, rispettivamente, delle piste di ground, V_{DC} , V_{CC} e V_{DD} . H1, H2, H3 e H4 sono quattro fori di fissaggio che si trovano ai vertici della scheda.

Gli ultimi due componenti del circuito sono mostrati in Fig. 9.17 e sono due regolatori di tensioni. I regolatori di tensione servono per generare una tensione in uscita di un certo valore a partire da una determinata tensione d'ingresso. Nella DD Low Int occorrono due regolatori di tensione per generare le tensioni da 5 V e 3,3 V per l'alimentazione dei vari integrati presenti sulla scheda, a

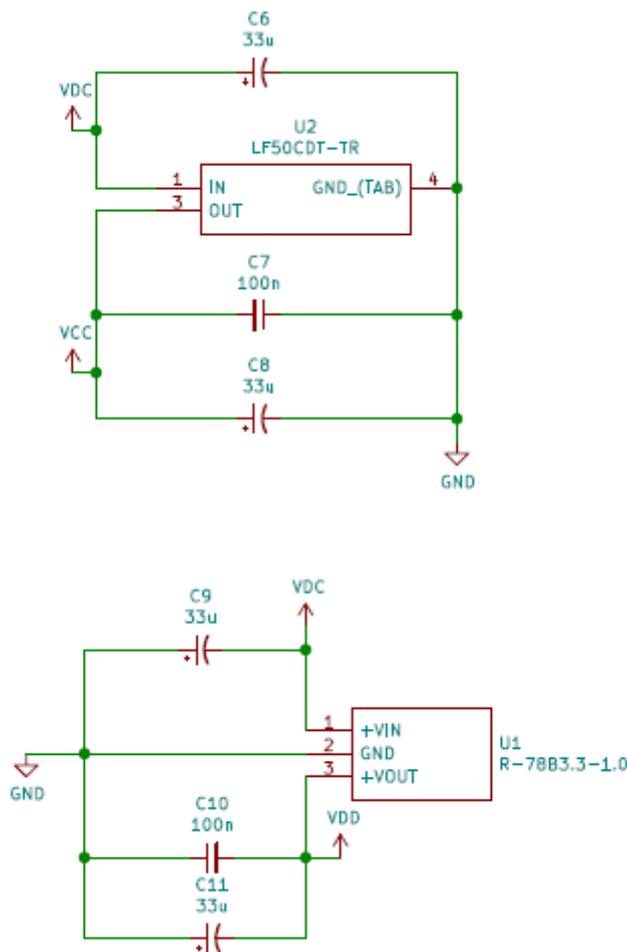


Fig.9.17. Parte di circuito comprendente i due regolatori di tensione.

partire dalla tensione di alimentazione della scheda V_{DC} pari a 6 V. Si è utilizzato l'LF50CDT-TR, denominato U2 nello schema elettrico, per convertire la tensione d'ingresso V_{DC} nella tensione d'uscita $V_{CC} = 5$ V. Questa serve per l'alimentazione dell'SN75451B e dei due 74HC244D. Per farlo si utilizza la configurazione visibile in Fig. 9.17, suggerita dal datasheet, costituita dai condensatori C6, C7 e C8. C6 e C8 sono due condensatori elettrolitici da 33 μ F mentre C7 è un condensatore ceramico multistrato da 100 nF (lo stesso utilizzato come condensatore di by-pass nei circuiti integrati visti precedentemente). L' R-78B3.3-1.0, denominato U1, è il regolatore di tensione necessario per convertire V_{DC} in $V_{DD} = 3,3$ V. Si ricorda che V_{DD} va ad alimentare i due 74LVC594A e il DS1100LZ-20. Anche qui si utilizza una certa configurazione di condensatori, suggerita dal datasheet del componente, la quale è identica a quella di U2. Vi sono due condensatori elettrolitici da 33 μ F (C9 e C11) e un condensatore ceramico multistrato da 100 nF (C10). Tutti i condensatori utilizzati nella DD Low Int sono condensatori SMD, ovvero a montaggio superficiale.

9.3) PCB DD Low Int

Il design del PCB (Printed Circuit Board) è stato sviluppato sempre con l'ausilio del software KiCad. Lo step successivo alla creazione dello schema elettrico è il disegno del circuito stampato.

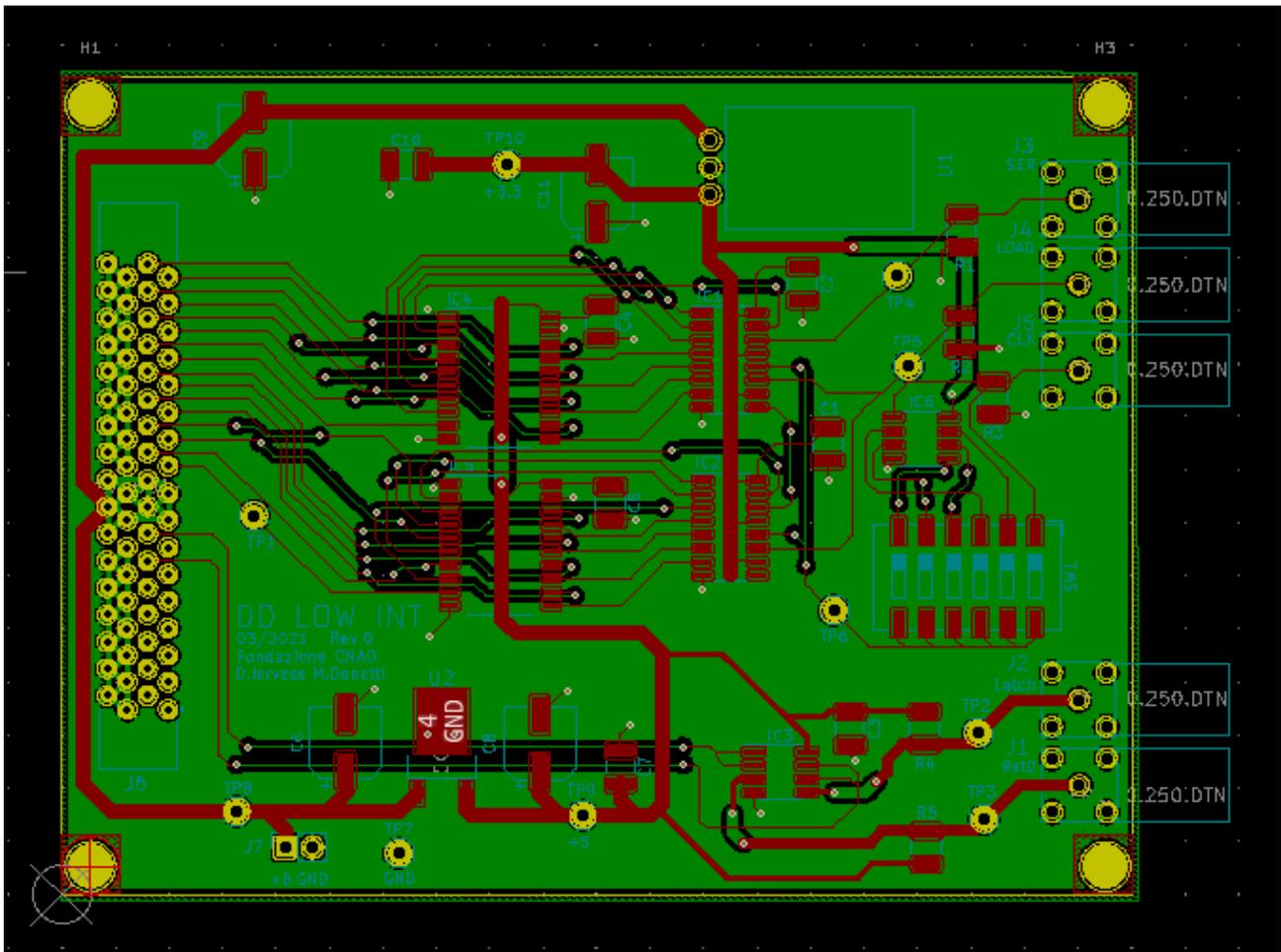


Fig.9.18. PCB terminato sul software KiCad.

In questo passo della progettazione della scheda viene deciso come disporre i vari componenti nel piano costituito dalla scheda stessa. Nella Fig. 9.18 è rappresentata la schermata del software KiCad relativa alla progettazione del PCB completata. La scheda ha uno spessore di 1,6 mm ed è costituita da due strati di rame. Quello inferiore costituisce il piano di ground, al quale vengono collegati tutti i pin di ground dei vari integrati e degli altri componenti della scheda, tramite un foro passante che permette di passare al piano inferiore di questa. Il piano di ground è quello rappresentato in verde nella Fig. 9.18. Il piano superiore è costituito da tutti i componenti della scheda e dalle piste di alimentazione e di collegamento tra i vari pin dei componenti, rappresentate in rosso nel disegno. Sono state utilizzate piste con diametri differenti. Le piste che trasportano le varie tensioni di alimentazione presenti nel circuito hanno un diametro di 1,5 mm o 1 mm. Nel resto del circuito, invece, per tutti gli altri collegamenti, sono state utilizzate piste con diametro di 0,5 mm e 0,25 mm. La scheda ha una forma rettangolare e una dimensione pari a [10 x 7,7] cm. Ai quattro vertici del rettangolo è possibile osservare i quattro fori H1, H2, H3 e H4 utili per il fissaggio della DD Low Int con il Disaccoppiatore Ottico tramite quattro torrette.

9.4) Scheda finale

Il PCB della DD Low Int è stato prodotto dall'azienda Baselectron S.r.l. alla quale è stato inviato l'insieme dei file prodotti su KiCad necessari alla realizzazione del PCB. Il PCB realizzato è raffigurato in Fig. 9.19. E' possibile vedere, grazie alla serigrafia presente nel PCB, tutte le sigle dei

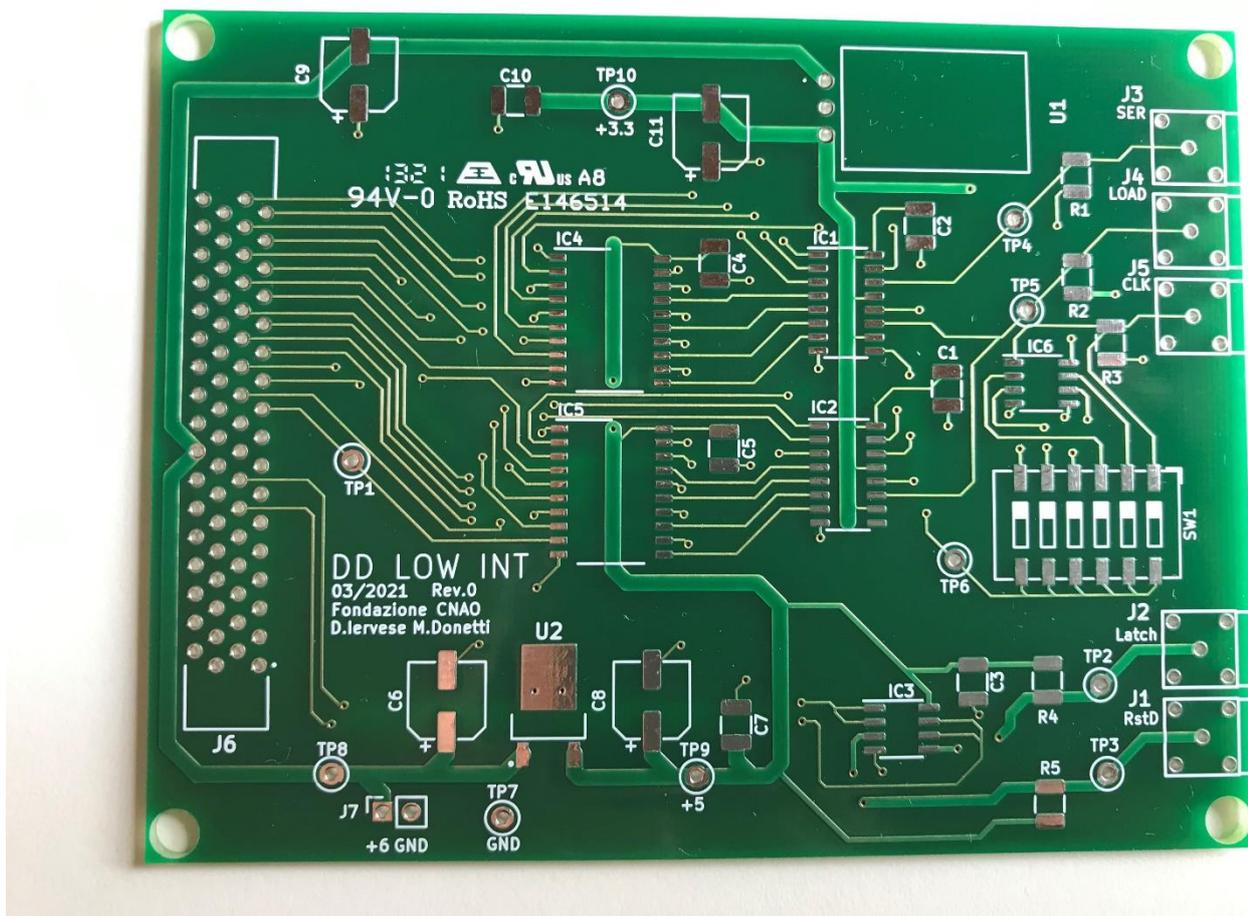


Fig.9.19. Circuito stampato DD Low Int.

componenti descritti durante la spiegazione dello schema elettrico (ad eccezione delle sigle riferite ai fori nei quattro vertici, in quanto non essendo componenti elettronici, non era necessario inserire la loro sigla nella scheda). Si può notare che, al di sotto delle sigle che fanno riferimento ai connettori LEMO, è presente il nome del segnale passante attraverso quel connettore.

La scheda DD Low Int finale, con tutti i componenti saldati, è illustrata in Fig. 9.20. Si ricorda che occorrono due di queste schede, in quanto è necessario collegarsi a due schede DT5550W / A55CIT4, una per il layer x e una per il layer y di fibre scintillanti.

Sia in Fig. 9.18 che in Fig. 9.19, è possibile notare che è presente un errore. Infatti, la pista che porta il segnale di Load, invece che collegarsi ai pin STCP dei due SIPO, è stata collegata ai pin SHCP. Lo stesso errore è stato commesso per la pista che porta il segnale di Clock, la quale, invece di collegarsi ai pin SHCP, si collega ai pin STCP. Questo è un errore di progettazione della scheda rilevato solamente dopo la produzione della stessa. Per questo è presente questa discordanza con lo schema elettrico illustrato nel Par. 9.2. In ogni caso l'errore è stato corretto in seguito eseguendo delle modifiche sul PCB.

Una volta saldati tutti i componenti della scheda si è potuto procedere con la verifica del suo corretto funzionamento. La scheda è stata alimentata usando un alimentatore da laboratorio attraverso il terminale a vite, nella maniera esposta nel Par. 9.2, in cui si è descritto il terminale a vite stesso. Già nella Fig. 9.20 si può intravedere, in basso, il suddetto componente, nel quale

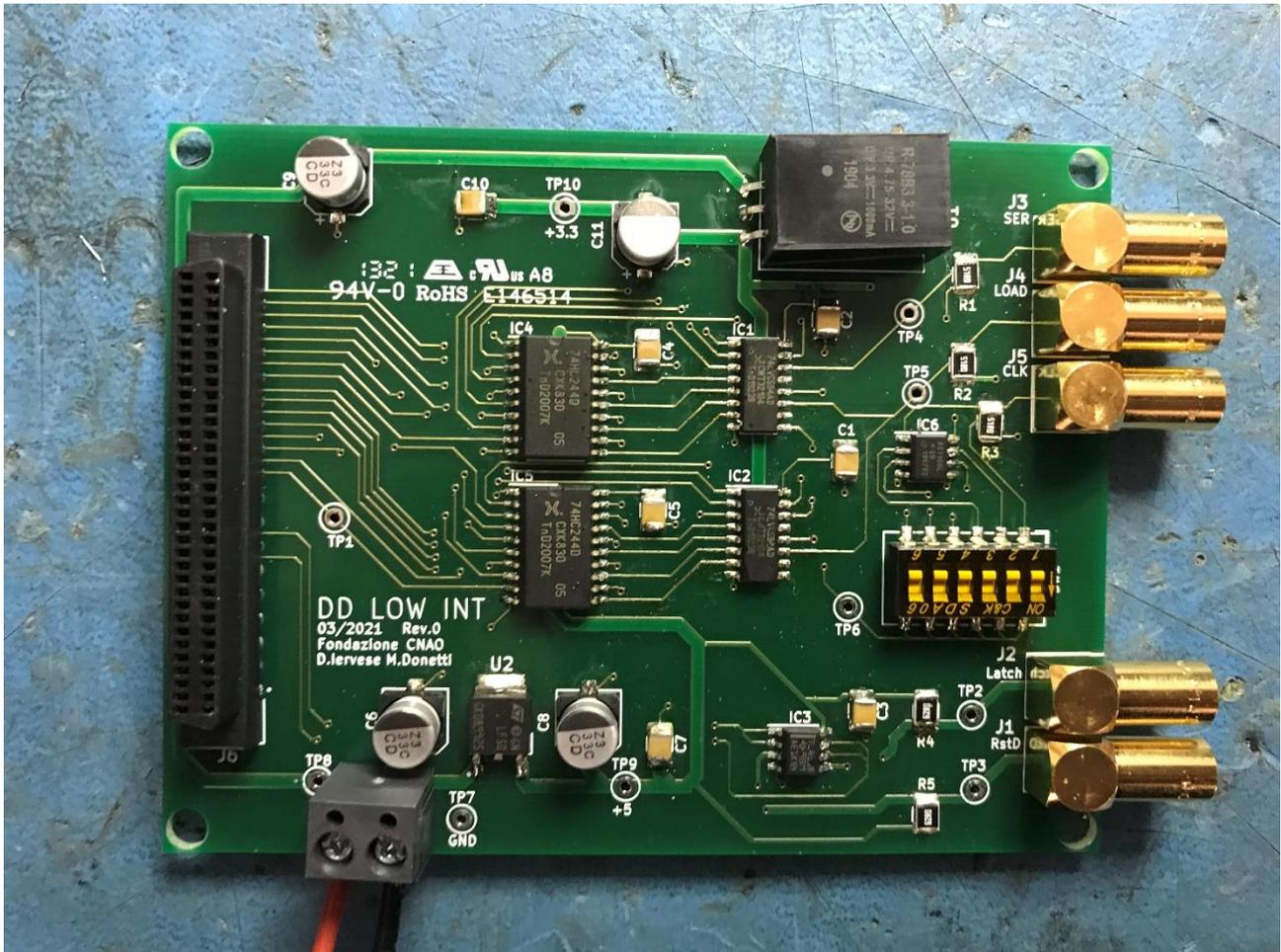


Fig.9.20. Scheda DD Low Int.

arrivano due fili, uno rosso e uno nero, che sono fissati al componente tramite serraggio a vite. Il primo filo porta i 6 V mentre il secondo la tensione di riferimento (ground). Per prima cosa sono stati utilizzati i test point TP7 e TP8 per verificare la corretta alimentazione della scheda. Con un tester si è verificato che TP7 fosse effettivamente 0 V e TP8 6 V. Dopodiché si è verificato il corretto funzionamento dei regolatori di tensione; quindi, tramite TP9 e TP10, si è osservato che i componenti U2 e U1 avessero in uscita le tensioni di 5 V e 3,3 V.

Terminata questa verifica, il passo successivo è stato constatare che i segnali Serial, Load e Clock giungessero correttamente nella scheda. Per fare questo è stato necessario collegare la DD Low Int con la DT5550W / A55CIT4. Nella Fig. 9.21 è illustrato il collegamento tra le due schede. La DD Low Int è correttamente alimentata a 6 V, come si può vedere nello schermo dell'alimentatore da laboratorio che segna quella tensione. Anche la DT5550W / A55CIT4 è alimentata e collegata al PC nel quale è aperto il software SCI-Compiler. La scheda CAEN viene accesa e, attraverso il PC, viene caricato nella sua memoria volatile dell'FPGA il programma descritto nel Cap.8. I tre connettori LEMO della scheda CAEN, dai quali fuoriescono i tre segnali (LEMO 4, 5, e 6), vengono collegati ai tre connettori LEMO J3, J4 e J5 della DD Low Int attraverso tre cavi LEMO della stessa lunghezza, come si può vedere nella Fig. 9.21. Tuttavia, affinché l'acquisizione dati sia avviata, è necessario far arrivare all'interno della scheda CAEN il segnale di Latch. Questo dovrebbe arrivare dal DDS che, però, non è presente, in quanto in questo test lo scopo è verificare il corretto interfacciamento tra queste due schede. Per questo motivo, il segnale di Latch, e anche di

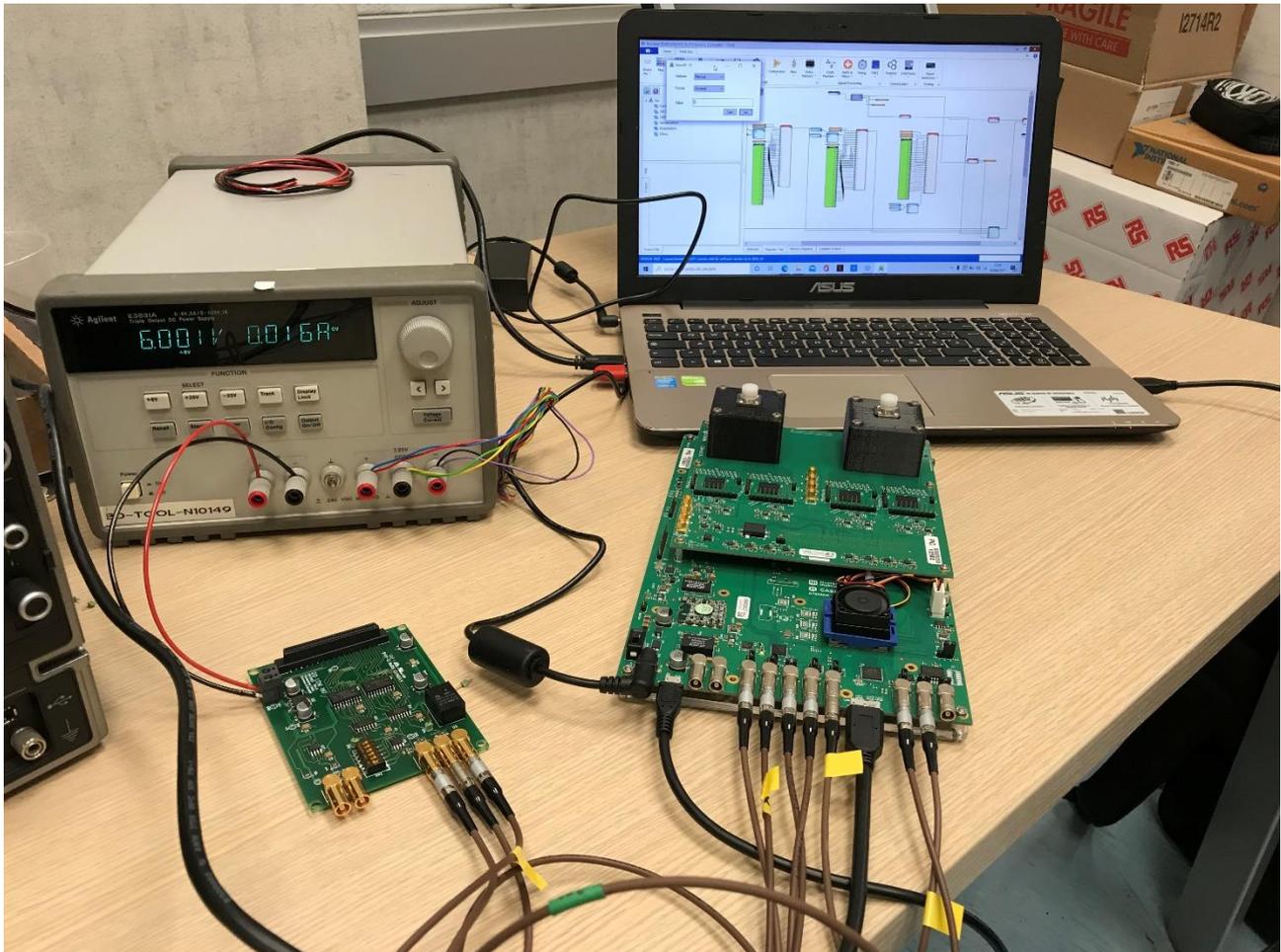


Fig. 9.21. Scheda DD Low Int correttamente alimentata e collegata alla DT5550W/A55CIT4.

ResetD, devono essere ancora una volta autogenerati dalla scheda CAEN stessa, cosa che il programma implementato fa. Nella Fig. 9.21, infatti, è possibile notare che non vi sono cavi LEMO collegati ai due connettori J1 e J2. Mentre, nella scheda CAEN, si può notare che ai LEMO 0, 1, 2 e 3 sono collegati dei cavi per far sì che, come da codice SCI-Compiler, i due segnali (Latch e ResetD) escano dai LEMO 0 e 1 e vengano portati indietro ai LEMO 2 e 3 grazie al connettore a T utilizzato nell'oscilloscopio, come già spiegato nel Par. 8.1. Questo è anche il motivo per cui, in questo test, non si potrà verificare la presenza dei segnali di Latch e ResetD nei test point TP2 e TP3, dato che, in quelle piste, non passa alcun segnale, non essendoci, di fatto, il collegamento della DD Low Int con il Disaccoppiatore Ottico e, quindi, col DDS.

Utilizzando una sonda per oscilloscopio sono stati visualizzati i segnali di interesse. In particolare, si è notato che sui pin di uscita di IC6 non vi era la presenza di alcun segnale. In effetti, quando è stato scelto questo componente, era stato considerato il fatto che questo sarebbe potuto accadere, in quanto il segnale che entra in input a IC6 è il clock a 160 MHz. Dal datasheet, si è visto che questo è un valore un po' al limite per quel circuito integrato e, infatti, una volta montato il componente sul circuito, il segnale non viene neanche visto avendo una frequenza troppo alta. Tuttavia, come spiegato anche nello schema elettrico, il segnale di Clock viene anche mandato direttamente da J5 nel primo pin dello Switch. Chiudendo quell'interruttore, si è verificato, come sarà visibile a breve, che con il clock non ritardato, i 16 bit sono parallelizzati correttamente. Per cui, anche se IC6 avesse

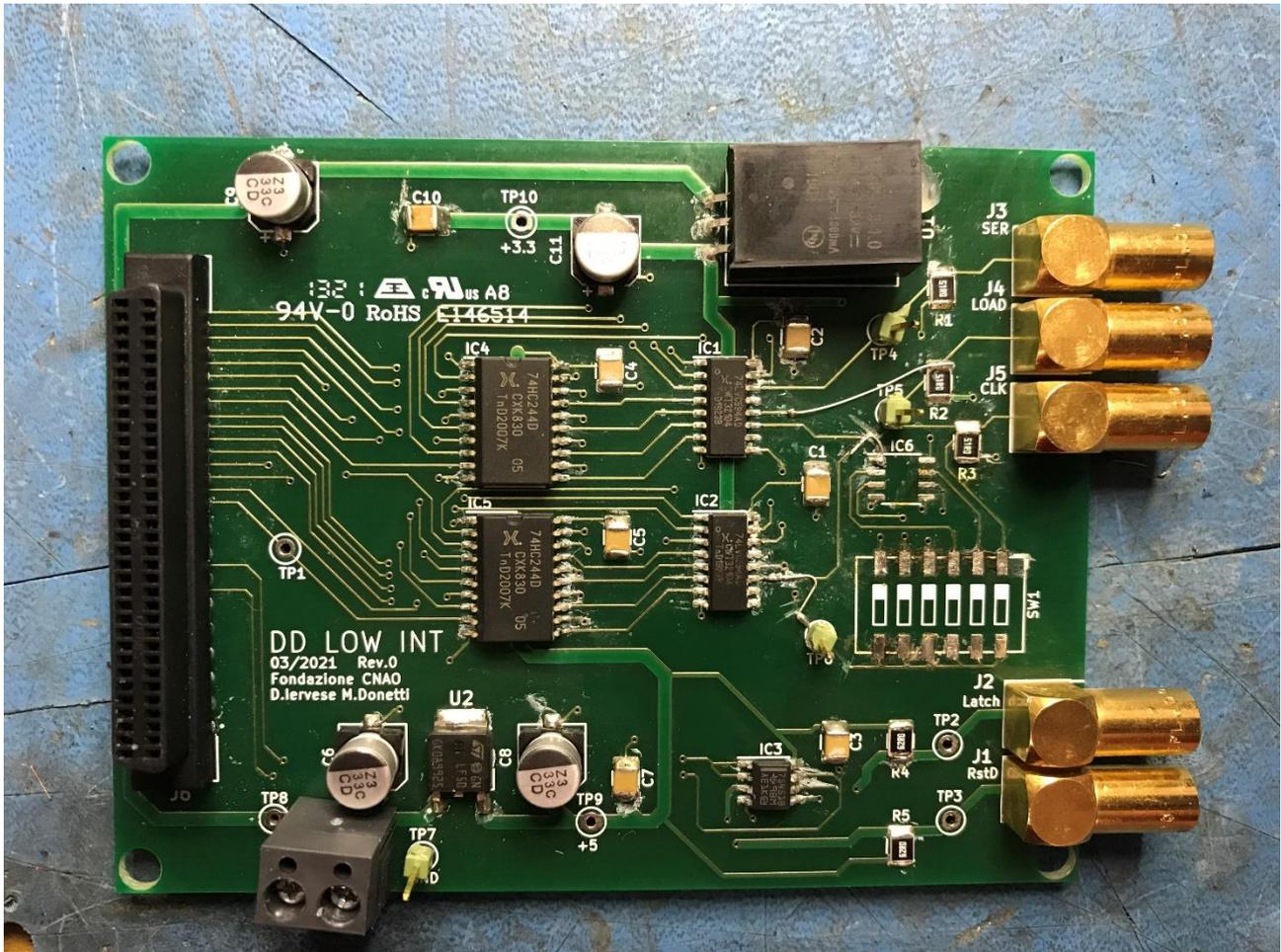


Fig. 9.22. Circuito DD Low Int a cui sono state applicate le modifiche di correzione.

funzionato correttamente, si sarebbe comunque utilizzato il clock non ritardato, ovvero non si avrebbe sfruttato nessuna uscita di IC6.

In Fig. 9.22 è visibile la DD Low Int a cui sono state applicate delle modifiche. Innanzitutto, è stato corretto l'errore riguardante il collegamento scambiato tra le piste di Load e Clock con i pin SHCP e STCP dei due SIPO. IC6, inoltre, è stato dissaldato, così come SW1. Infatti, con la prima modifica, il segnale di Clock è stato mandato direttamente nei pin SHCP dei SIPO (e a TP6 per preavarlo facilmente con la sonda), per cui risulta inutile il passaggio dello stesso segnale attraverso lo switch, la cui funzionalità era abbinata ad IC6.

Nella Fig. 9.23, si può osservare sull'oscilloscopio il segnale prelevato dal test point TP4, ovvero Serial. E' presente, in giallo, l'impulso di Latch autogenerato dal codice, seguito dai primi tre canali di acquisizione del Citiroc che variano alle frequenze prestabilite di 1, 2 e 4 Hz, come spiegato nel Par. 8.7. La distanza temporale tra ogni flusso di dati, così come le ampiezze degli impulsi sono quelle desiderate. In Fig. 9.24 è possibile notare il segnale prelevato con la sonda dal test point TP5, ovvero il segnale di Load. E' un impulso con periodo di 200 ns e ampiezza di 2,2 V, come ci si aspettava. Nella Fig. 9.25 si può osservare il clock da 160 MHz, collocando la sonda sul test point TP6. Infine, nella Fig. 9.26, si osserva sull'oscilloscopio il segnale prelevato dal test point TP1. Quest'ultimo si trova sulla pista del circuito nel quale passa il bit meno significativo dei 16 che

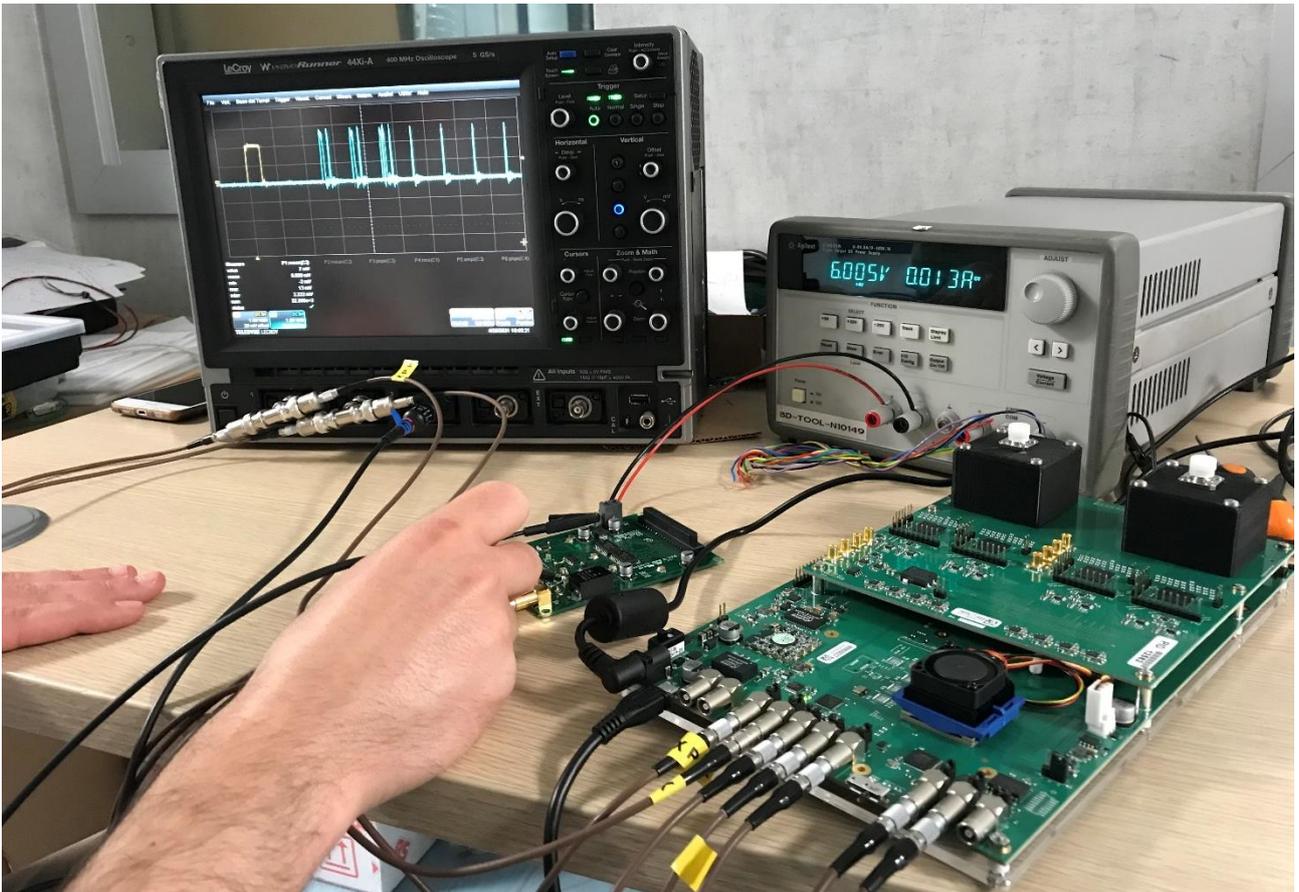


Fig. 9.23. Visione dei primi tre canali Citiroc dal testpoint TP4.

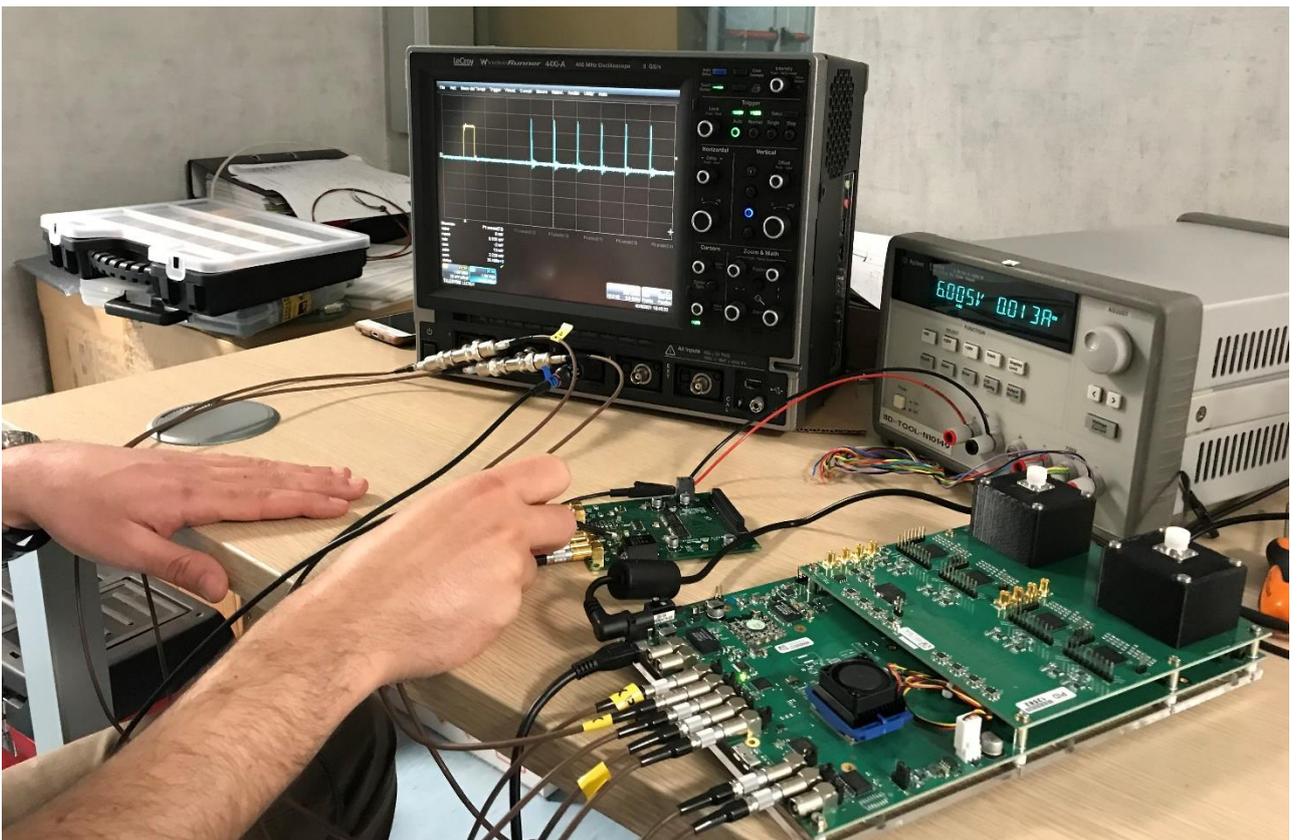


Fig. 9.24. Visione del segnale di Load dal testpoint TP5.

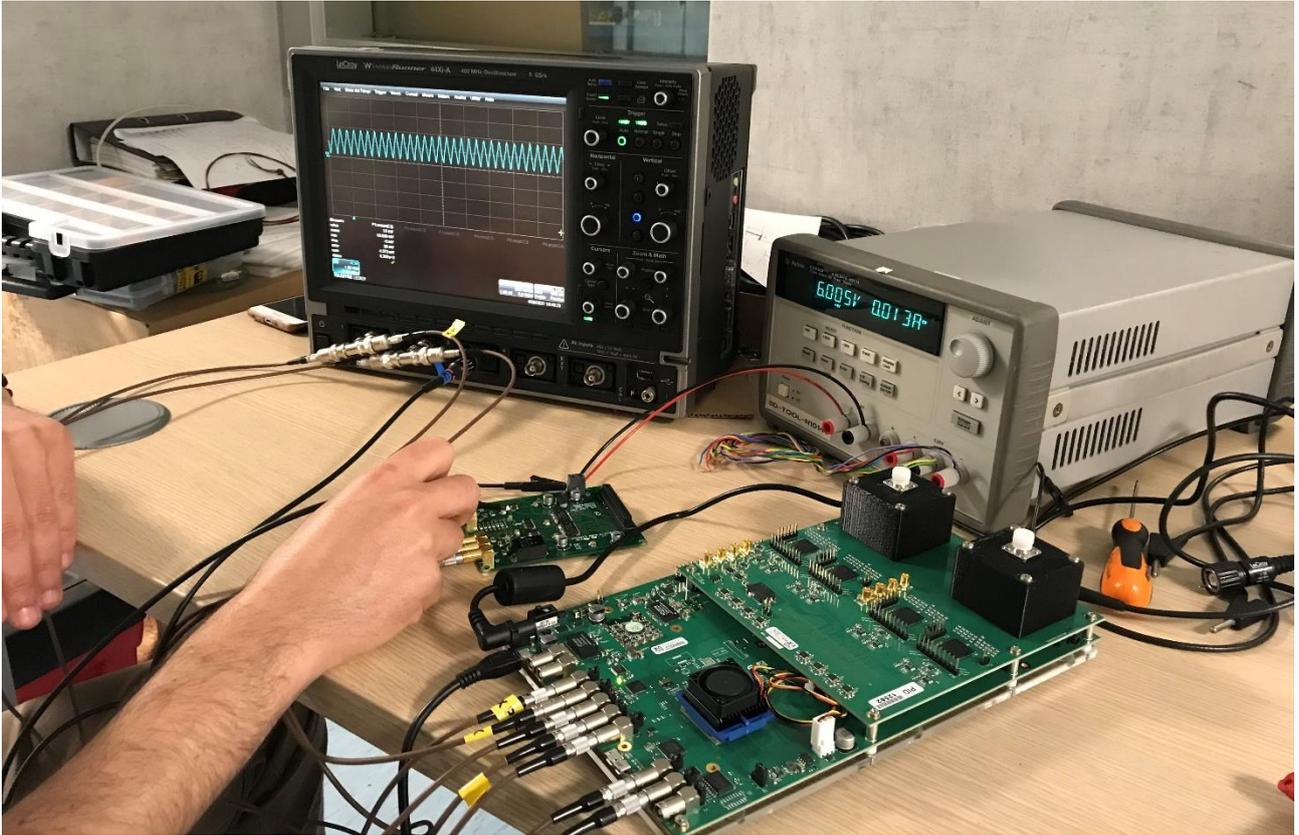


Fig. 9.25. Visione del segnale di clock dal test point TP6.

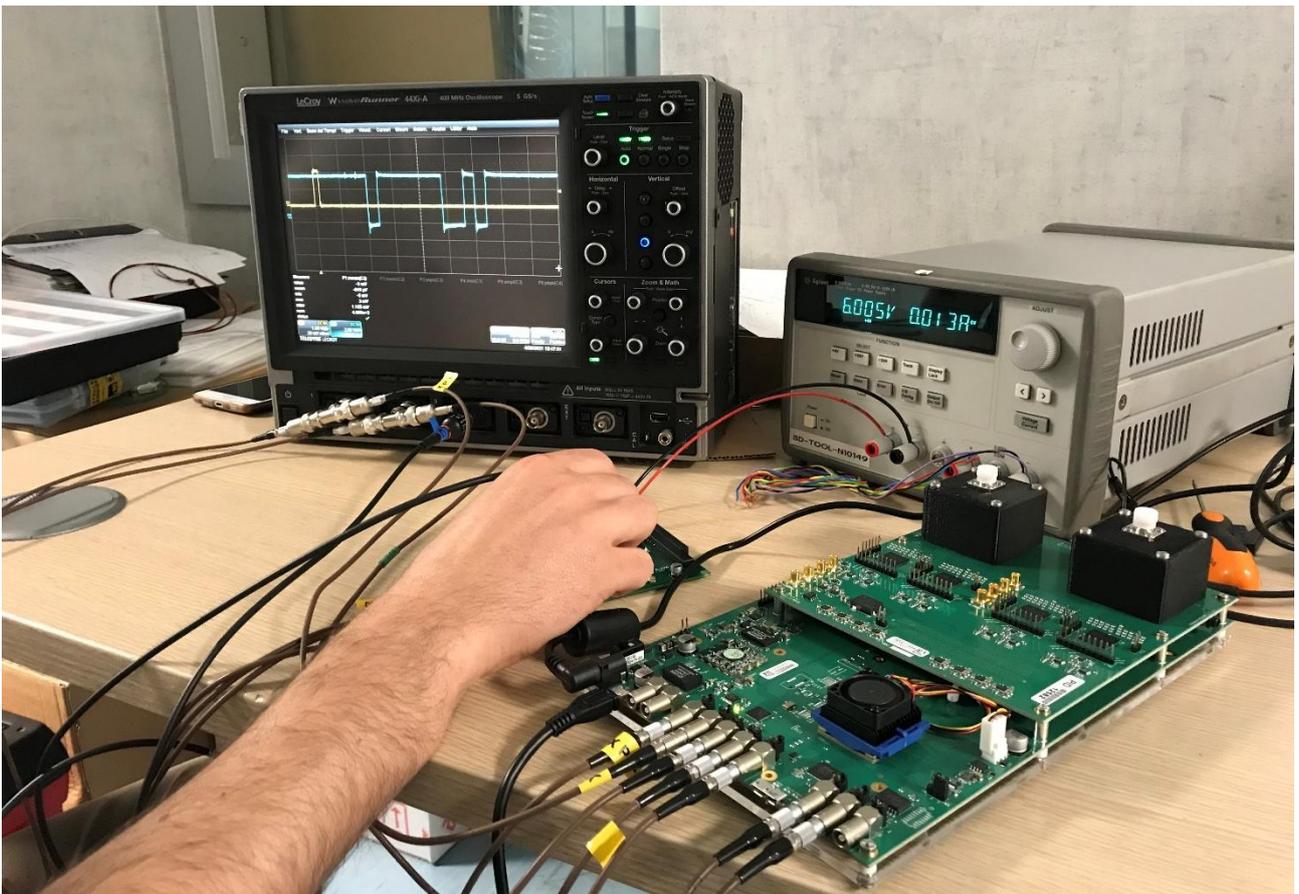


Fig. 9.26. Visione del bit meno significativo dei 16, che cambia nei vari canali, dal test point TP1.

vengono parallelizzati ogni 200 ns. Dall'oscilloscopio è stato possibile vedere un'onda quadra, che rappresenta il singolo bit, variare ogni secondo. Questo rappresenta il bit meno significativo del canale 0 del Citiroc, nel quale è stata mandata un'onda quadra di frequenza pari a 1 Hz tramite il firmware. Affianco a questo bit che variava, era presente un altro segnale che variava alla frequenza di 2 Hz. In questo caso si è osservato la variazione del bit meno significativo dei 16, in uscita dal canale 1. Lo stesso accadeva per ognuno dei canali in cui, all'interno del codice, è stata mandata un'onda quadra ad una certa frequenza, come illustrato nel Par. 8.7. Inoltre, i segnali visibili su TP1 hanno un'ampiezza di 5 V. Si ricorda, infatti, che questi segnali si trovano su un pin d'uscita di IC5, ossia il driver 74HC244D alimentato a 5 V ed inserito appositamente per dare un'uscita TTL a 5 V verso il Disaccoppiatore Ottico. Quest'ultima verifica su TP1 è stata necessaria per riscontrare il corretto funzionamento dei due SIPO per ottenere la conversione dei dati da seriale a parallelo come desiderato.

Test di acquisizione con il DDS

Una volta testato il corretto interfacciamento tra la scheda DT5550W / A55CIT4 e la DD Low Int, e aver verificato che l'operazione di parallelizzazione dei dati fosse effettuata, il passo successivo è stato quello di collegare il sistema costituito dalle due schede prima nominate al DDS. Presso la Fondazione CNAO sono presenti tre sale dedicate al trattamento dei pazienti, dove vi sono le BOX e i crate del sistema di controllo del DDS. Inoltre, è presente una quarta sala, nella quale vi sono ancora le BOX e il crate del DDS. Questa sala non è dedicata al trattamento dei pazienti ed è nominata Sala Sperimentale. Così come per le altre sale, anche in Sala Sperimentale il crate del DDS si trova in un'altra stanza rispetto alle BOX, per evitare danni da radiazione. Per testare il sistema di acquisizione del monitor per basse intensità si è utilizzata proprio la Sala Sperimentale.

Ciò che manca all'acquisizione dati, rispetto ai test eseguiti finora, sono i comandi provenienti dal DDS (Latch, ResetD e StrobeOut), il quale dovrà ricevere indietro il segnale di StrobeIn seguito dai dati di acquisizione, e un'interfaccia in grado di permettere la visualizzazione dei conteggi dei vari canali, che, per ora, sono sempre stati visualizzati sull'oscilloscopio. Per fare questo è stato utilizzato il software LabVIEW, il quale è il software adoperato al CNAO per la programmazione delle FPGA del DDS. Tramite LabVIEW è stato implementato un programma da trasferire ad una FPGA del DDS, in particolare quella che controlla la camera strip. In questo programma si definiscono i segnali che l'FPGA deve inviare e ricevere e le tempistiche con le quali deve farlo. Questo, quindi, è il programma con il quale vengono programmate le interconnessioni dell'FPGA del DDS. Inoltre, è stato implementato un ulteriore programma LabVIEW in modo tale da avere l'interfaccia per vedere e gestire l'acquisizione dati. In questo viene richiamato il programma LabVIEW FPGA, in modo tale che, andando ad effettuare il Run del codice, l'FPGA esegua le varie funzioni per il quale è stato programmato e l'acquisizione venga gestita dal PC su cui è installato LabVIEW, e visualizzata su di esso.

In LabVIEW i programmi sono chiamati "Virtual Instruments" (VI). La programmazione è di tipo grafico ad oggetti. Il codice sorgente è molto simile ad un diagramma di flusso formato da nodi, utili per eseguire funzioni specifiche, che connessi tra loro tramite fili, determineranno l'ordine di esecuzione del flusso dati. Il VI è composto dal Front Panel e dal Block Diagram. Il primo è la finestra che rappresenta l'interfaccia tra il programma e l'utilizzatore, dove all'interno si possono trovare tutti i controllori e gli indicatori relativi al VI. Con il termine controllore, si intende una variabile d'ingresso, introdotta dall'utente, che può essere modificata agendo sul Front Panel. Con indicatore si intende una variabile d'uscita il cui valore può essere modificato dal programma. Il Block Diagram è quella parte del VI che contiene il codice attraverso una rappresentazione del tipo diagramma a blocchi, ed è formato da:

- Nodi: elementi di elaborazione interconnessi tra loro da fili (collegamenti) che svolgono delle funzioni;
- Collegamenti: uniscono i nodi, permettono lo scambio di informazioni e corrispondono alle variabili.

Nel DDS, LabVIEW viene utilizzato per gestire un sistema NI ECM (Embedded Control & Monitoring). Questo sistema è raffigurato in Fig. 10.1 I prodotti hardware per sistemi NI ECM



Fig. 10.1. Sistema NIECM.

condividono un'architettura comune che comprende tre componenti: un controllore embedded per la comunicazione e l'elaborazione; FPGA per un controllo avanzato, temporizzazioni, processamento di segnali, filtraggio e protocolli personalizzati; moduli I / O industriali per la connessione con qualsiasi sensore o dispositivo [3].

Occorre precisare che, nel test che è stato effettuato, i dati che sono stati acquisiti sono sempre i dati simulati; i canali contano le transizioni date dalle forme d'onda generate nel firmware grazie ai blocchi PULSE GENERATOR di SCI-Compiler. Il motivo è dovuto al fatto che la costruzione del monitor non è ancora giunta al termine. Tuttavia, con questi test è possibile verificare la correttezza della catena elettronica di acquisizione dati a partire dalla scheda DT5550W / A55CIT4, sulla quale verranno montate le due matrici di SiPM appena sarà pronto il monitor, fino al DDS.

10.1) Programma LabVIEW

Di seguito sarà descritto il codice LabVIEW implementato per l'acquisizione. Questo è stato portato a termine sul crate del DDS della sala controllo della Sala Sperimentale del CNAO. Si descriverà prima il codice con il quale si è programmata l'FPGA del DDS. Come con SCI-Compiler, anche in LabVIEW, quando si implementa un codice con il quale programmare un'FPGA, occorre un compilatore che converta lo schema a blocchi LabVIEW in linguaggio VHDL. Dopodiché, si passerà alla descrizione del programma che viene eseguito sul controllore embedded, sul quale si richiama il file del codice precedente, grazie al quale sarà possibile ottenere un'interfaccia per visualizzare e gestire i dati acquisiti.

10.1.1) Block Diagram codice LabVIEW FPGA

Nella Fig.10.2 viene mostrato il Block Diagram del codice LabVIEW con il quale si programma l'FPGA del DDS. Si possono notare due rettangoli grandi. Quei rettangoli sono While Loop e,

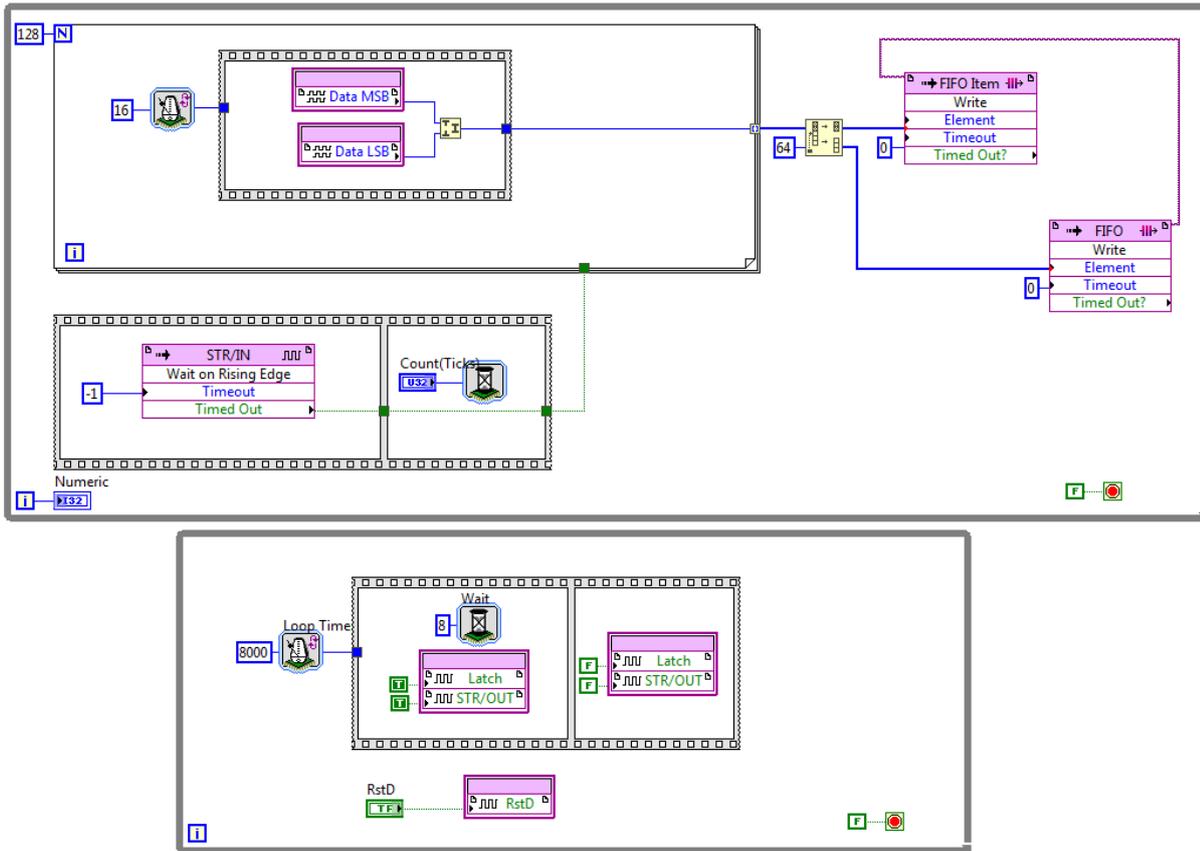


Fig. 10.2. BlockDiagram del programma LabVIEW FPGA.

all'interno dei due, vi sono specificate le azioni che l'FPGA svolge in lettura e in scrittura. La funzione della struttura While Loop è quella di eseguire il diagramma contenuto all'interno del rettangolo che lo delimita, purché al terminale di condizione non sia assegnato un valore booleano "false". Per come sono stati impostati, i due While Loop eseguono ciò che vi è al loro interno per un tempo infinito.

Il While Loop più in basso rappresenta la parte di scrittura dell'FPGA. Gli elementi di colore viola chiaro sono i cosiddetti FPGA I / O Node. Questi nodi possono essere inseriti nel diagramma e da questi selezionare le linee I / O dell'FPGA desiderate. All'interno sono state utilizzate le linee I / O dove passano i segnali di ResetD, Latch e StrobeOut. Si può notare che, nel nodo che simboleggia la linea ResetD, viene collegato un controllore booleano. Questo sarà rappresentato nel Front Panel come un interruttore o un pulsante, che potrà essere schiacciato per impostare un 1 logico o uno 0 logico su quella linea I / O. In questo modo si è implementata la funzione di ResetD sul DDS. E' possibile notare la presenza di un altro rettangolo all'interno di quello rappresentante il While Loop: la Sequence Structure. Questa viene utilizzata per controllare il flusso di dati nel diagramma a blocchi. Nel caso in questione si è utilizzata una Sequence Structure di tipo Flat. Queste sono costituite da frame; nella parte del diagramma a blocchi che si sta descrivendo vi sono due frame. Ciò che è presente all'interno di un frame viene eseguito in maniera sequenziale a partire dal frame di sinistra. E' presente un simbolo collegato alla Flat Sequence Structure denominato Loop Timer. Quest'ultimo ha la funzione di impostare il tempo con il quale ripetere le operazioni all'interno della Flat Sequence Structure. Questo tempo viene deciso impostando un valore temporale nello spazio alla sinistra del nodo Loop Timer. In questo caso si è utilizzato il valore 8000. Questo valore è riferito in termini di cicli di clock, in relazione al clock di 80 MHz utilizzato nell'intero DDS. Con

quel valore l'intervallo temporale impostato è di 100 μ s. Nel primo frame sono presenti i nodi I / O FPGA di Latch e StrobeOut. Questi due segnali hanno la stessa forma e temporizzazione e devono essere mandati dal DDS verso il sistema di acquisizione dati del monitor per basse intensità nello stesso momento. A questi due nodi viene collegato in ingresso il valore booleano 1. Inoltre, nello stesso frame, è presente il nodo Wait. Questo serve per inserire un ritardo tra due operazioni sull'FPGA. In questo caso il ritardo è pari al valore specificato sulla sinistra del nodo stesso, ossia 8 cicli di clock. Questo valore, in termini temporali, equivale a 100 ns. Dopo 100 ns l'esecuzione passa al secondo frame della Flat Sequence Structure, dove ci sono i due stessi nodi I / O FPGA, sui quali è collegato in ingresso il valore booleano 0. Dato che l'intera Flat Sequence Structure viene ripetuta periodicamente per 100 μ s, e il primo frame occupa un intervallo temporale di esecuzione di 100 ns, il secondo frame farà sì che, in quelle linee I / O, vi sia un livello logico basso per i restanti 99,9 μ s. In poche parole, si sono generati i segnali di Latch e StrobeOut in modo che abbiano una periodicità di 100 μ s e rimangano a livello logico alto per 100 ns.

Prima di passare alla spiegazione del diagramma a blocchi all'interno del While Loop più in alto, occorre descrivere meglio la funzione dei due segnali StrobeOut e StrobeIn, e come questi vengono gestiti per l'acquisizione per il monitor per basse intensità. Lo StrobeOut è un impulso che entra nella scheda elettronica già nominata nei capitoli precedenti, ovvero il Disaccoppiatore Ottico, e fuoriesce con un certo ritardo, sotto il nome di StrobeIn. Questi due impulsi servono per la sincronizzazione del DDS con l'arrivo dei dati, dato che tra il DDS e le BOX c'è una distanza che non è fissa, ma può variare tra le varie Sale Trattamento e la Sala Sperimentale. Nell'elettronica di acquisizione del DDS per le BOX1 e BOX2, i dati dei canali di acquisizione non sono acquisiti come nel monitor per basse intensità. Come già enunciato nel Par. 9.2, il DDS manda anche gli indirizzi dei canali che devono essere acquisiti. Per questo motivo occorre un segnale di StrobeIn, che funge da trigger per informare il DDS che occorre leggere il dato che si sta ricevendo. Nel caso del monitor per basse intensità questo non si verifica dato che gli indirizzi sono autogenerati all'interno del firmware ad una frequenza fissa, per cui, l'FPGA che riceve i dati, sa che questi arriveranno ogni 200 ns. Inoltre, il Disaccoppiatore Ottico è stato progettato in modo tale da inserire un ritardo tra StrobeOut, che la scheda riceve in ingresso, e StrobeIn, che manda in uscita, il quale serve a compensare il ritardo dovuto alla distanza percorsa dai segnali (in termini di cavi di connessione) e al tempo di elaborazione dati. Questo ritardo, essendo stato tarato appositamente a livello hardware per il tempo di elaborazione dei chip Tera, non avrà nulla a che vedere con il tempo di elaborazione del sistema DT5550W / A55CIT4 + DD Low Int. Per questo motivo, lo StrobeIn non giunge al DDS in sincronia con i dati e sarà necessario regolare un ritardo in LabVIEW per ottenere questa sincronia. Inoltre, lo StrobeIn non arriva al DDS ogni volta che quest'ultimo riceve un dato (quindi un canale), come accade per l'acquisizione delle BOX1 e BOX2, ma riceverà un impulso di StrobeIn che verrà usato come trigger per l'acquisizione in sequenza di tutti i 128 canali.

Fatta questa precisazione si può passare alla descrizione del contenuto del secondo While Loop presente in alto in Fig. 10.2. Questa parte di codice è dedicata alla gestione dell'arrivo dei dati dei canali di acquisizione e del segnale di StrobeIn al DDS; si tratta, quindi, della parte di lettura dell'FPGA. Anche questo While Loop è eseguito per un tempo infinito e, inoltre, nell'angolo in basso a sinistra del rettangolo, si può vedere un indicatore collegato all'indice di scorrimento dei vari cicli che verranno eseguiti nel While Loop. In questo modo, nel Front Panel, ci sarà un

indicatore denominato “Numeric” che indica il numero dei cicli del While Loop, permettendo, inoltre, di verificare l’esecuzione del programma. La prima sezione del diagramma a blocchi da commentare è la Flat Sequence Structure costituita da due frame. Nel primo frame è presente un FPGA I / O Method. Questo permette l’invocazione di un determinato metodo su un elemento FPGA I / O, in questo caso sulla linea I / O StrobeIn. Il metodo utilizzato si chiama “Wait on Rising Edge”. Consiste nell’attendere che su quella linea I / O arrivi un fronte di salita, il quale viene atteso per un certo tempo, il cui valore viene dato in ingresso alla voce “Timeout”. Quando il fronte di salita viene rilevato, dalla voce “Timed Out” viene fuori un trigger che, nel caso in Fig. 10.2, permette l’esecuzione del secondo frame della Sequence Structure. Nel caso in descrizione, in ingresso a “Timeout” si mette il valore -1, il quale indica che il fronte di salita su quella linea viene atteso all’infinito. Il trigger viene collegato al secondo frame, che viene eseguito ogni volta che lo riceve. Nel secondo frame è presente semplicemente il nodo Wait, al cui input è collegato un controllore. Nell’input bisognerebbe inserire un valore che indica il ritardo che deve essere aggiunto dal Wait. Questo ritardo è necessario per il motivo esposto nella spiegazione precedente relativa ai segnali StrobeOut e StrobeIn. Non essendoci sincronia tra l’arrivo dello StrobeIn e dei 128 canali, non inserendo alcun ritardo, con molta probabilità il primo canale che viene acquisito non è effettivamente il primo che viene mandato, ma potrebbe corrispondere a uno o più canali dopo o prima, in base allo sfasamento che c’è tra l’impulso StrobeIn e il primo canale mandato. Tuttavia, non si può conoscere a priori il valore necessario, per cui si inserisce il controllore, in modo tale da poter regolare il ritardo durante l’esecuzione del programma. Dopo che passerà un certo tempo, che verrà impostato in fase di esecuzione del programma, il secondo frame termina la sua funzione e, il segnale di trigger proveniente dal primo frame, come si può vedere in Fig. 10.2, viene collegato al rettangolo che si trova sopra, il quale è un For Loop. La sua funzione è quella di ripetere l’esecuzione della porzione di codice in esso contenuta un certo numero di volte. All’interno del For Loop è presente una Flat Sequence Structure da un frame. In questa Structure vi sono due FPGA I / O Node che rappresentano le linee I / O nelle quali passano i dati a 16 bit da acquisire. In particolare, vi sono due nodi: Data MSB e Data LSB. Nel primo vi sono gli 8 bit più significativi mentre nel secondo vi sono gli 8 bit meno significativi dei 16. I fili in uscita da questi due nodi entrano nell’elemento “Join Numbers”, il quale mette insieme i due input, ovvero i due insiemi da 8 bit, dando in uscita i due ingressi uniti. Quindi, in uscita da questo elemento si hanno i 16 bit. La Sequence Structure viene temporizzata con un Loop Timer, inserendo il valore 16 che simboleggia 200 ns (perché i dati sono acquisiti ogni 200 ns). Ogni 200 ns si hanno i 16 bit in uscita dall’elemento “Join Numbers”. Questi sono portati fuori dalla Structure e, quindi, dal For Loop. Si può notare, nell’angolo in alto a sinistra del rettangolo For Loop, che il loop viene ripetuto 128 volte, essendo questo il numero di canali da acquisire. L’uscita del For Loop corrisponde ad un array da 128 elementi.

A questo punto, per completare l’acquisizione dei dati, occorre utilizzare una DMA (Direct Memory Access) FIFO. In generale la FIFO, acronimo che indica First In First Out, è un elemento che si utilizza per immagazzinare dati, dove il primo dato che entra è il primo a uscire da essa. La frequenza di ingresso dei dati, però, è differente dalla frequenza di uscita di questi. Il motivo di necessità della DMA FIFO è il seguente. Come è stato detto inizialmente, il Block Diagram che si sta descrivendo è quello con il quale si programma l’FPGA del DDS. Qui vengono acquisiti i dati provenienti dal sistema scheda CAEN + DD Low Int. Questi dati saranno mostrati sul PC della sala controllo della Sala Sperimentale grazie all’implementazione di un VI LabVIEW eseguito sul PC

stesso. Questo VI interagisce col VI FPGA ricevendo i dati provenienti da quest'ultimo, permettendo la loro visualizzazione sul Front Panel del secondo VI. Il problema deriva dal fatto che il sistema operativo Windows è un sistema general-purpose. Le applicazioni LabVIEW in esecuzione su Windows non garantiscono l'esecuzione in tempo reale perché Windows non è un sistema operativo real-time. Windows non può garantire che il codice finisca sempre entro limiti di tempo specifici. Il tempo necessario per l'esecuzione del codice in Windows dipende da molti fattori, inclusi altri programmi in esecuzione in background. Windows deve anche gestire gli interrupt da dispositivi come una porta USB, tastiera, mouse e altre periferiche che possono ritardare l'esecuzione del codice [3]. Il trasferimento di dati tra il VI FPGA e il VI Windows utilizzando controllori e indicatori richiede l'utilizzo del processore del PC. Per cui la velocità di trasferimento dati è dipendente dalla velocità e dalla disponibilità del processore. Si potrebbero, però, perdere dei dati se il VI FPGA scrive dei dati ad un indicatore più velocemente della capacità di lettura da parte del VI Windows. La DMA FIFO trasferisce i dati dall'FPGA direttamente alla memoria del sistema operativo target. Le DMA FIFO possono trasmettere grandi quantità di dati tra il VI FPGA e il VI Windows. Questo consente all'FPGA di utilizzare la RAM del sistema operativo come se fosse la propria, offrendo vantaggi significativi in termini di prestazioni [3]. Una DMA FIFO è costituita da due parti, come mostrato in Fig. 10.3. Una parte FPGA e una parte PC host. La parte FPGA della DMA FIFO è implementata nell'FPGA. L'FPGA VI scrive nella FIFO un elemento alla volta con il metodo DMA FIFO Write. La parte PC host della DMA FIFO viene immagazzinata in memoria sul PC target. LabVIEW utilizza un controller DMA per collegare le due parti [3].

Nel programma LabVIEW della Fig. 10.2, l'array da 128 elementi, nella quale sono presenti i dati, vengono mandati in un elemento denominato "Split 1D Array" prima di entrare in due DMA FIFO. Questo elemento va a dividere l'array in corrispondenza della sua 64-esima cella, come si può

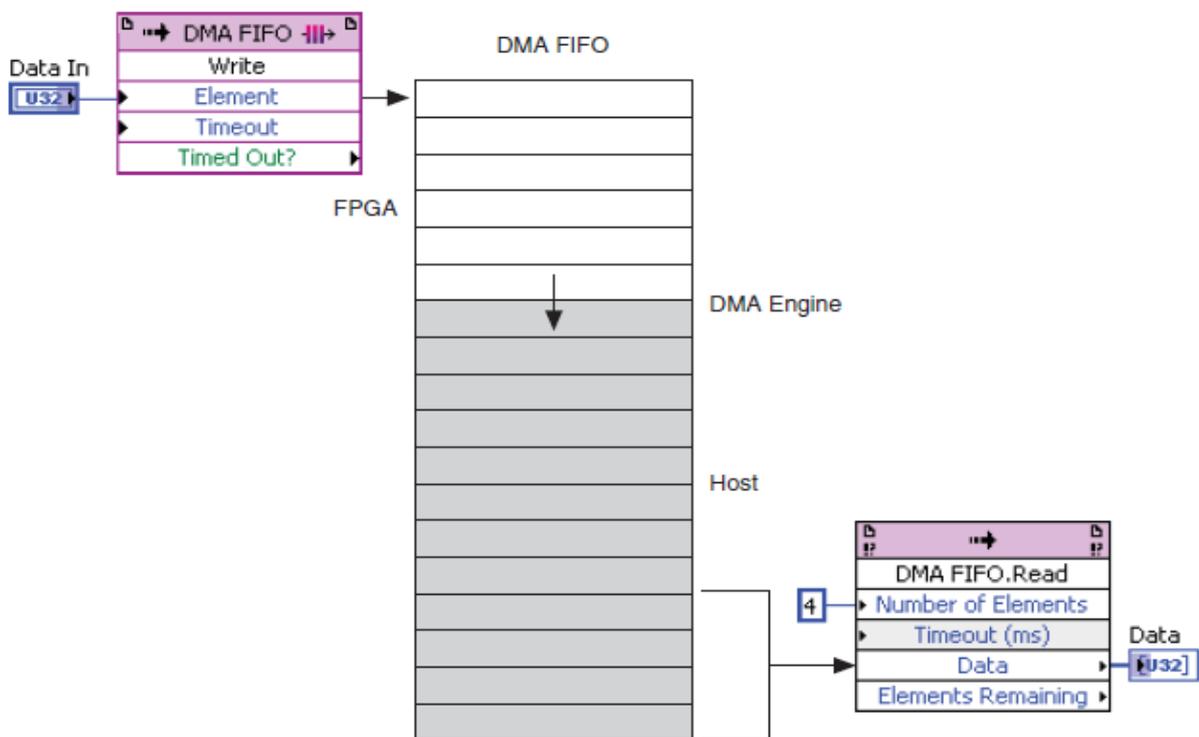


Fig. 10.3. Le due parti della DMA FIFO [3].

vedere dal valore 64 presente in ingresso ad esso. La suddivisione dell'array è stata effettuata perché la DMA FIFO è stata creata come un blocco di trasferimento di dimensione massima pari a 64 dati. L'ingresso "Element" della FIFO indica i dati o gli elementi che la FIFO prende in ingresso per essere immagazzinati. L'ingresso "Timeout", invece, specifica l'intervallo di tempo, in cicli di clock, che il metodo utilizzato nella FIFO (nel caso del codice in questione si utilizza il Write Method) attende affinché si abbia spazio disponibile nella FIFO nel caso in cui questa sia piena. Collegando 0 vuol dire che non si aspetta nessun intervallo di tempo.

Si hanno due FIFO identiche. In una entrano i dati dei primi 64 canali, nell'altra quelli degli ultimi 64 canali. Dalla FIFO più in basso esce un filo viola che si collega in ingresso all'altra FIFO. Quel collegamento indica che la FIFO più in basso è la prima ad essere eseguita; quindi, in questa, vi entrano i primi 64 canali mentre nella FIFO più in alto subentrano gli ultimi 64. Con questo accorgimento il trasferimento dei dati è più veloce.

10.1.2) Block Diagram codice LabVIEW PC host

In Fig. 10.4 è rappresentato il Block Diagram del codice LabVIEW che riceve i dati provenienti dalla DMA FIFO descritta prima. Con questo programma viene generata un'interfaccia grazie alla quale è possibile osservare il conteggio sui vari canali di acquisizione.

Il primo simbolo in basso a sinistra nella figura indica il riferimento all'FPGA VI, chiamato "Open FPGA VI Reference". Utilizzando questo elemento è resa possibile la comunicazione con l'FPGA VI. I due blocchi successivi sono due DMA FIFO utilizzate in lettura (Read Method). La voce "Number of Elements" specifica il numero di elementi che vengono letti dalla DMA FIFO. "Timeout" indica un tempo di attesa per la lettura dei dati. Il tempo di default è 5000 ms. Se il tempo di attesa scade prima che si raggiunga il valore in "Number of Elements", l'uscita della DMA FIFO sarà vuota. "Data" porta in uscita i dati letti dalla DMA FIFO. Infine, la voce "Elements Remaining" riporta il numero di elementi rimasti nella memoria della parte PC host della DMA FIFO. Il modo in cui queste due DMA FIFO sono collegate nel codice, ha il fine di svuotare i

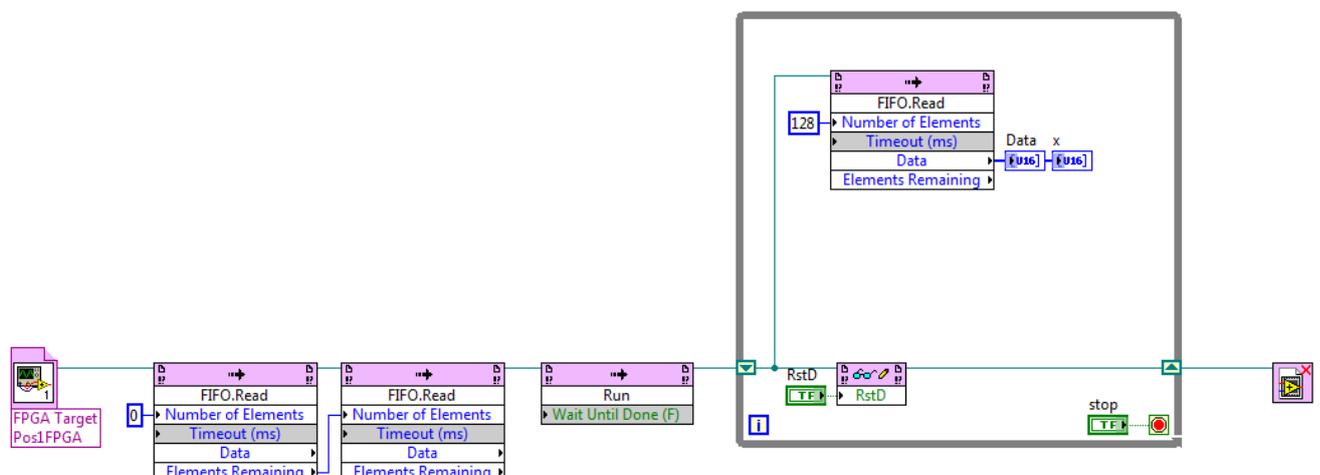


Fig. 10.4. Block Diagram del programma LabVIEW PC host.

dati che possono essere presenti nella FIFO. Ad esempio, nel caso in cui avvenga una brusca interruzione del programma potrebbero rimanere dei dati. Dopo le due DMA FIFO appena descritte, è presente il Run Method. Questo ha il compito di far eseguire il codice LabVIEW dell'FPGA VI. In questo modo, quando si effettua il Run del codice della Fig. 10.4, viene eseguito prima il codice LabVIEW dell'FPGA VI che scrive i dati e, poi, il codice LabVIEW PC host che li legge.

Il terminale di uscita del Run Method, dal quale arrivano i dati provenienti dall'FPGA VI, viene portato all'interno di un While Loop e, qui, in ingresso a una DMA FIFO in Read Method. A differenza dei While Loop usati nel codice precedente, dove al terminale di condizione era presente un valore booleano costante, in questo è collegato un controllore booleano al terminale di condizione, denominato "stop". Con quel controllore è possibile fermare l'acquisizione dati dal Front Panel. La DMA FIFO nel While Loop è impostata in modo tale che legga i 128 elementi che provengono dalle DMA FIFO dell'FPGA VI, come si può vedere dal valore 128 in "Number of Elements". All'uscita "Data" vengono collegati due indicatori chiamati "Data" e "x". Il primo permetterà la visualizzazione in colonna dei conteggi dei 128 canali mentre il secondo mostra l'andamento dei conteggi in un grafico. Il terminale di uscita del Run Method viene collegato, sempre all'interno del While Loop, ad un altro elemento, ossia il Read/Write Control. Questo ha il compito di leggere un valore da un indicatore o scrivere un valore a un controllore sull'FPGA VI. In questo metodo viene selezionata la linea I/O ResetD, in modo che, nel Front Panel, sarà presente il controllore booleano per il reset dei contatori del firmware della DT5550W / A55CIT4. Infine, fuori dal While Loop, è presente il "Close FPGA VI Reference", il quale chiude, appunto, il riferimento all'FPGA VI.

10.2) Front Panel dei due codici LabVIEW

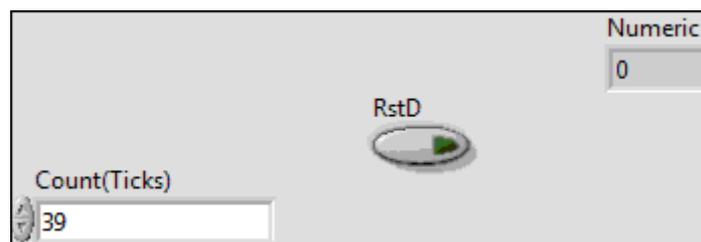


Fig.10.5. Front Panel dell'FPGA VI.

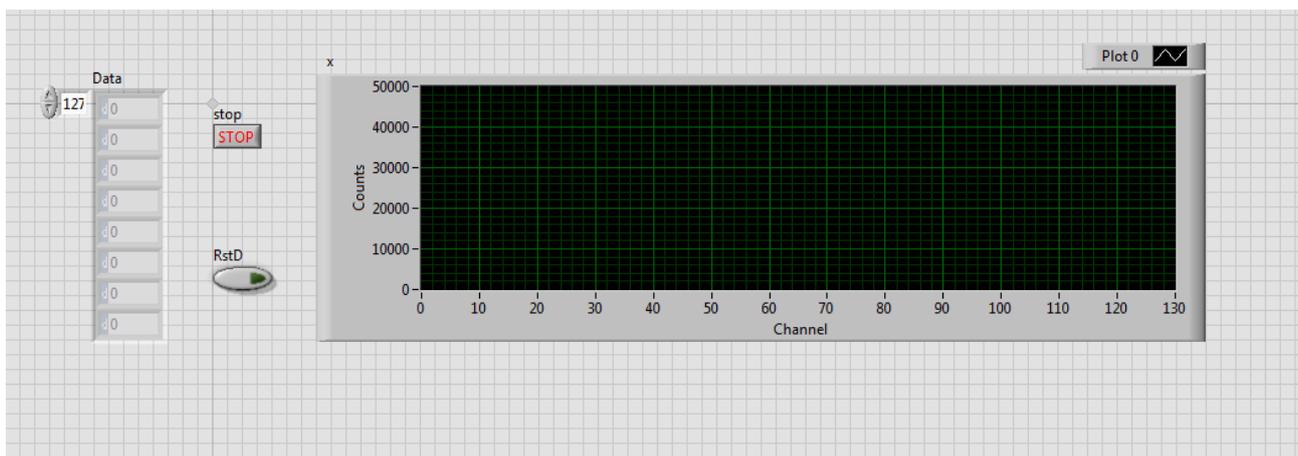


Fig.10.6. Front Panel del VI Windows.

In questo paragrafo vengono illustrati i Front Panel dei due codici LabVIEW descritti. In Fig. 10.5 è rappresentato il Front Panel dell'FPGA VI: questo comprende l'indicatore "Numeric", il controllore "Count" e il controllore "RstD". Il primo indica l'avanzamento dei cicli del While Loop in cui è presente la parte di lettura dell'FPGA. "Count" è il controllore collegato al nodo "Wait" che serve per impostare il tempo di attesa tra l'arrivo del segnale di StrobeIn e l'arrivo dei 128 canali in sequenza al DDS. Impostando, inizialmente, il valore su quel controllore pari a 0, è stato possibile verificare che l'impulso di StrobeIn fosse troppo in anticipo rispetto all'arrivo del primo dato. Infatti, il primo canale che veniva acquisito non corrispondeva al primo canale dei 128, ma al 127esimo del flusso di dati precedente, a causa della mancanza di sincronia tra lo StrobeIn e i dati. Cambiando manualmente il valore in "Count", è stato inserito un valore pari a 39 cicli di clock, corrispondente a circa 490 ns. Questo è il tempo in cui il programma, dopo aver mandato lo StrobeIn al DDS, attende prima di ricevere i 128 canali, in modo tale che il primo canale visualizzato corrisponda al canale 0. Infine, è presente il pulsante "RstD" per azzerare il conteggio di tutti i canali.

In Fig. 10.6 è presente il Front Panel del VI Windows. Questa è l'interfaccia con la quale si gestisce l'acquisizione dati. Si può notare il pulsante di stop per fermare l'acquisizione dei dati e, ancora, il ResetD per l'azzeramento dei canali. Sulla sinistra è visibile l'indicatore "Data" rappresentato da una colonna di caselle, nelle quali si vedrà il conteggio dei vari canali. Nell'immagine si possono vedere solo otto caselle ma il loro numero può essere diminuito o aumentato. Inoltre, è possibile selezionare i canali da visualizzare, cliccando sulle frecce visibili alla sinistra della colonna, provocando lo scorrimento dei canali, oppure inserendo direttamente il numero del canale che si vuole visualizzare. L'ultimo indicatore presente è il grafico, il quale permette una visualizzazione generale dei conteggi su tutti i canali. Sull'asse delle ascisse è rappresentato il numero di canali mentre nelle ordinate il conteggio relativo ad ogni canale.

10.3) Test di acquisizione dati

Come è stato affermato all'inizio di questo capitolo, i test per l'acquisizione dati per il monitor per basse intensità sono stati effettuati nella Sala Sperimentale del CNAO. La scheda CAEN DT5550 / A55CIT4 con il PC, per la programmazione della sua FPGA, e la scheda DD Low Int sono state trasferite nella stanza dove sono presente le BOX del DDS della Sala Sperimentale. Essendo stata programmata in LabVIEW l'FPGA che controlla la camera strip, è stato necessario aprire il supporto della BOX1, contenente la camera in questione. Qui è stata rimossa la connessione, attraverso il cavo flat, tra il Disaccoppiatore Ottico della camera e la scheda contenente il chip Tera06. Il cavo è stato utilizzato per collegare il Disaccoppiatore Ottico alla DD Low Int come è possibile vedere nella Fig. 10.7. Nella figura si può distinguere il Disaccoppiatore fissato al supporto della BOX1, il cui connettore SCSI viene collegato a quello della DD Low Int per mezzo del cavo flat. Al di sotto del Disaccoppiatore Ottico per la camera strip si può notare, invece, il Disaccoppiatore Ottico per la camera integrale e il suo collegamento alla scheda contenente il chip Tera07 (quest'ultima non visibile trovandosi all'interno della BOX).

La Fig. 10.8 mostra il sistema completo (a meno del crate del DDS e del controllore embedded che si trovano nella sala controllo della Sala Sperimentale). Si nota il PC sul quale è presente SCI-Compiler collegato alla scheda CAEN correttamente alimentata, per la programmazione dell'FPGA della scheda stessa. Dopodiché si osservano le connessioni tramite i cavi LEMO tra la scheda

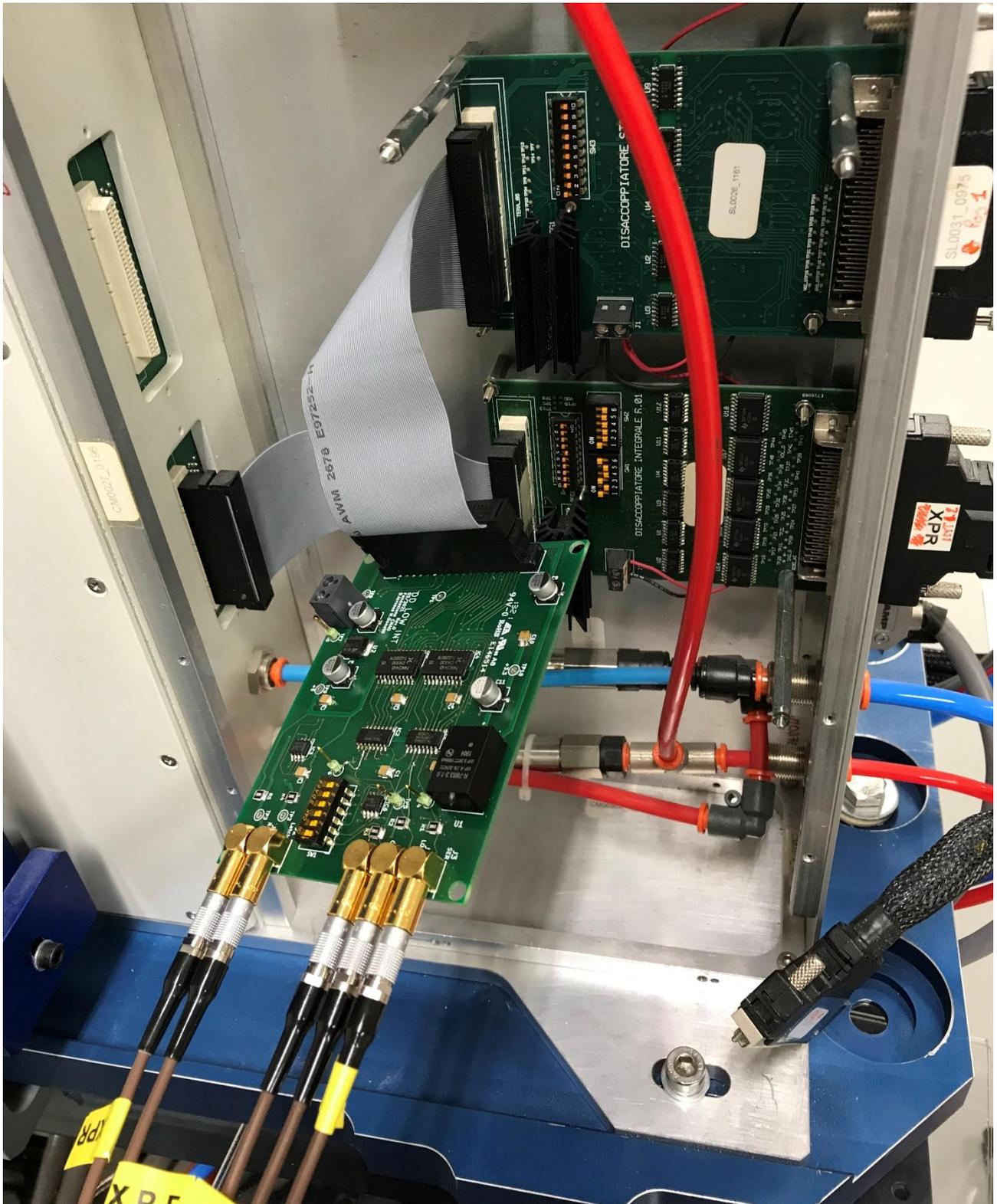


Fig.10.7. Connessione tra la scheda DD Low Int e il Disaccoppiatore Ottico della BOX1.

CAEN e la DD Low Int. Quest'ultima è connessa al Disaccoppiatore Ottico, come è stato precedentemente illustrato. Inoltre, si può vedere da una migliore prospettiva la BOX1.

Il primo passo per effettuare l'acquisizione dei dati, a partire dalla configurazione in Fig. 10.8, (quindi con le connessioni tra le schede elettroniche in gioco già ultimate) è l'accensione della scheda CAEN tramite il suo interruttore apposito. Successivamente, si procede alla



Fig.10.8. Collegamento del sistema DT5550W/A55CIT4 + DD Low Int alla BOX1.

programmazione dell’FPGA della scheda stessa dal PC con SCI-Compiler. La programmazione dell’FPGA richiede alcuni secondi. Questo passaggio deve essere ripetuto ogni volta che la scheda CAEN viene spenta, in quanto si va a programmare la memoria volatile dell’FPGA. Una volta avvenuta la programmazione dell’FPGA, la scheda CAEN manda ogni 100 μ s i conteggi dei 128 canali dei Citiroc, i quali ricevono i dati simulati nel firmware stesso solo in determinati canali. Si ricorda che sono mandate una serie di onde quadre in sei canali dei quattro Citiroc presenti sulla DT5550W / A55CIT4. Nei canali 0, 1 e 2 sono mandate onde quadre alla frequenza, rispettivamente, di 1, 2 e 4 Hz. Nel canale 32 è inviata un’onda quadra di frequenza pari a 0,5 Hz mentre nel canale 64 l’onda quadra varia a una frequenza di 0,25 Hz. Infine, nel canale 127 l’onda quadra ha una frequenza di 160 Hz. A questo punto, l’ultimo passo è eseguire il Run del VI Windows in LabVIEW descritto nel Par. 10.1.2 nella sala controllo e osservare l’acquisizione dati sul Front Panel del VI.

Le schermate dell’acquisizione sono mostrate nelle figure successive. In Fig. 10.9 viene mostrato il Front Panel dell’acquisizione con il programma LabVIEW in esecuzione. E’ un’istantanea nella quale si nota il conteggio dei primi tre canali di acquisizione. Da questa immagine si può notare la concordanza tra ciò che viene mandato e ciò che viene acquisito. I tre canali, infatti, contano alle frequenze prestabilite e lo si può notare andando a confrontare i tre valori nell’immagine. Il secondo canale mostra un valore doppio rispetto al primo, a certificazione del fatto che nel canale 1 è mandata una forma d’onda con frequenza doppia rispetto al canale 0. A sua volta, il canale 2 ha una

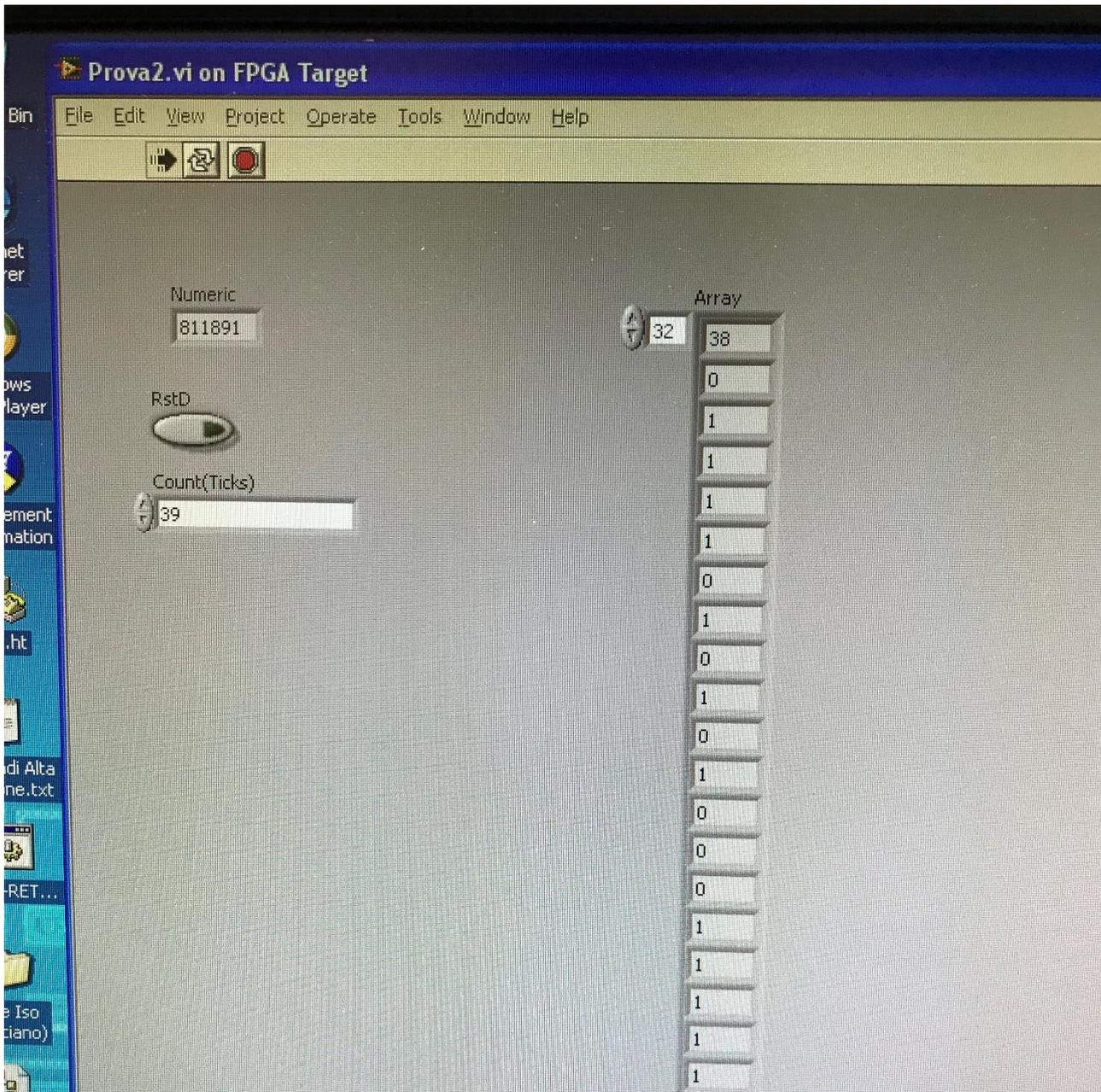


Fig.10.10. Conteggio del canale 32.

Un'altra cosa che si può notare in tutte le immagini riguardanti l'acquisizione è il valore presente negli altri canali nei quali, a livello firmware, non è stata mandato alcun segnale. Come era già stato espresso nel Cap. 8, in quei canali si nota un valore pari a 1 o 0. E' un comportamento dovuto, con molta probabilità, alla mancanza delle due matrici di SiPM sulla scheda DT5550W / A55CIT4.

Inoltre, in Fig. 10.13, si può osservare la schermata dell'acquisizione dati in cui il pulsante RstD è attivato (è illuminato di verde). Con il ResetD a livello logico alto è possibile notare che tutti i canali hanno azzerato il loro conteggio, dimostrando il corretto funzionamento della funzione di reset.

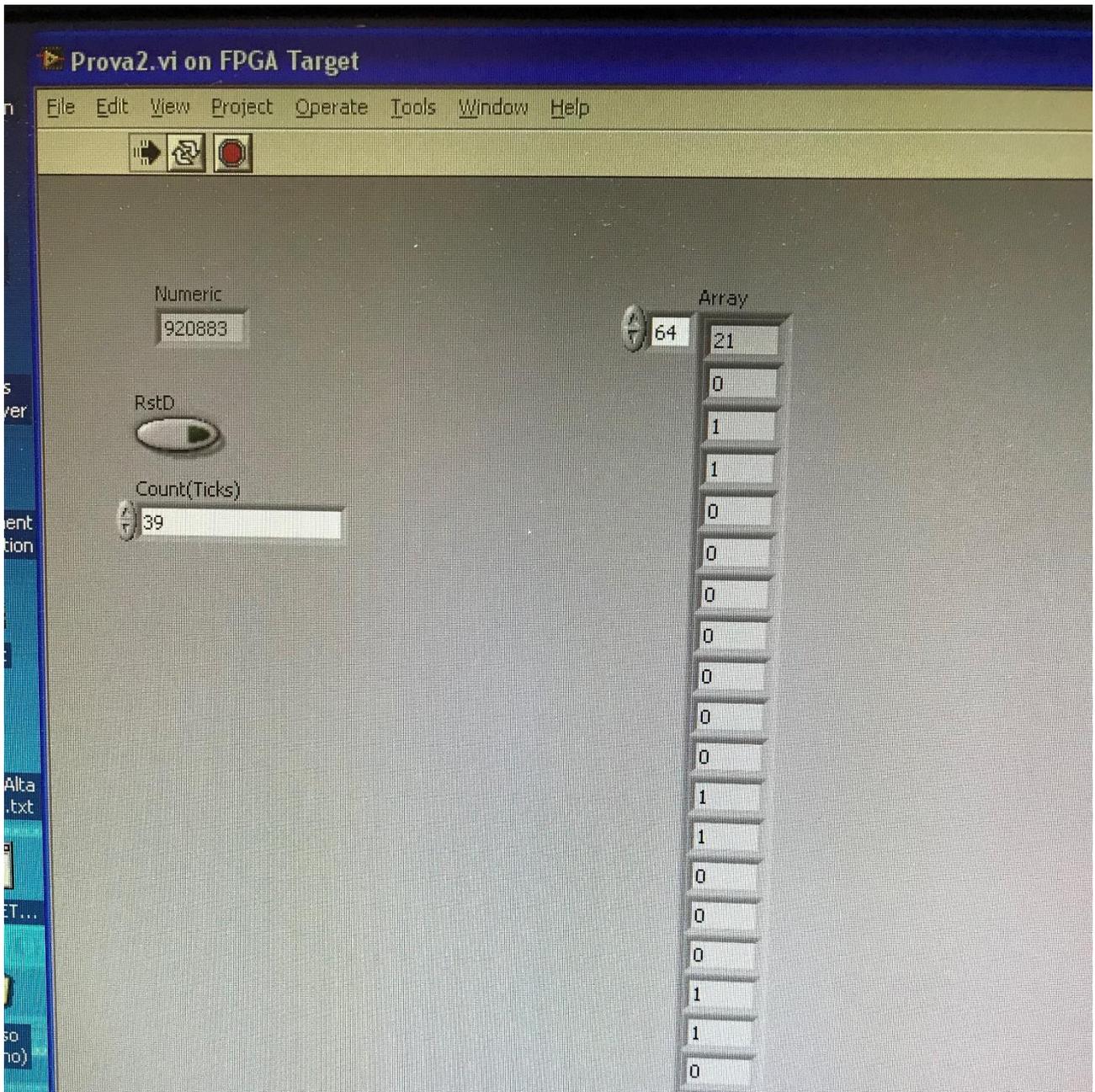


Fig.10.11. Conteggio del canale 64.

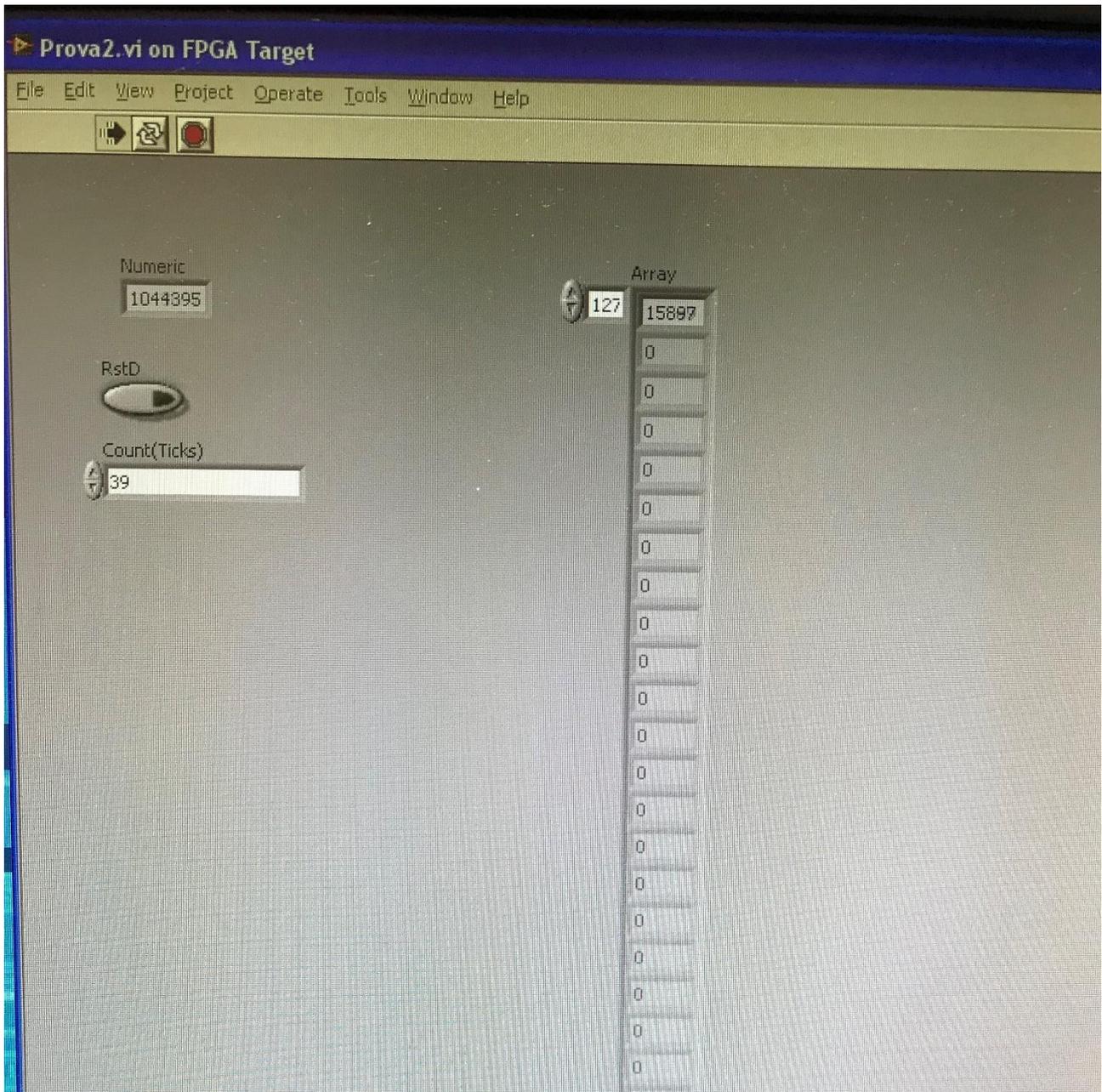


Fig.10.12. Conteggio del canale 127.

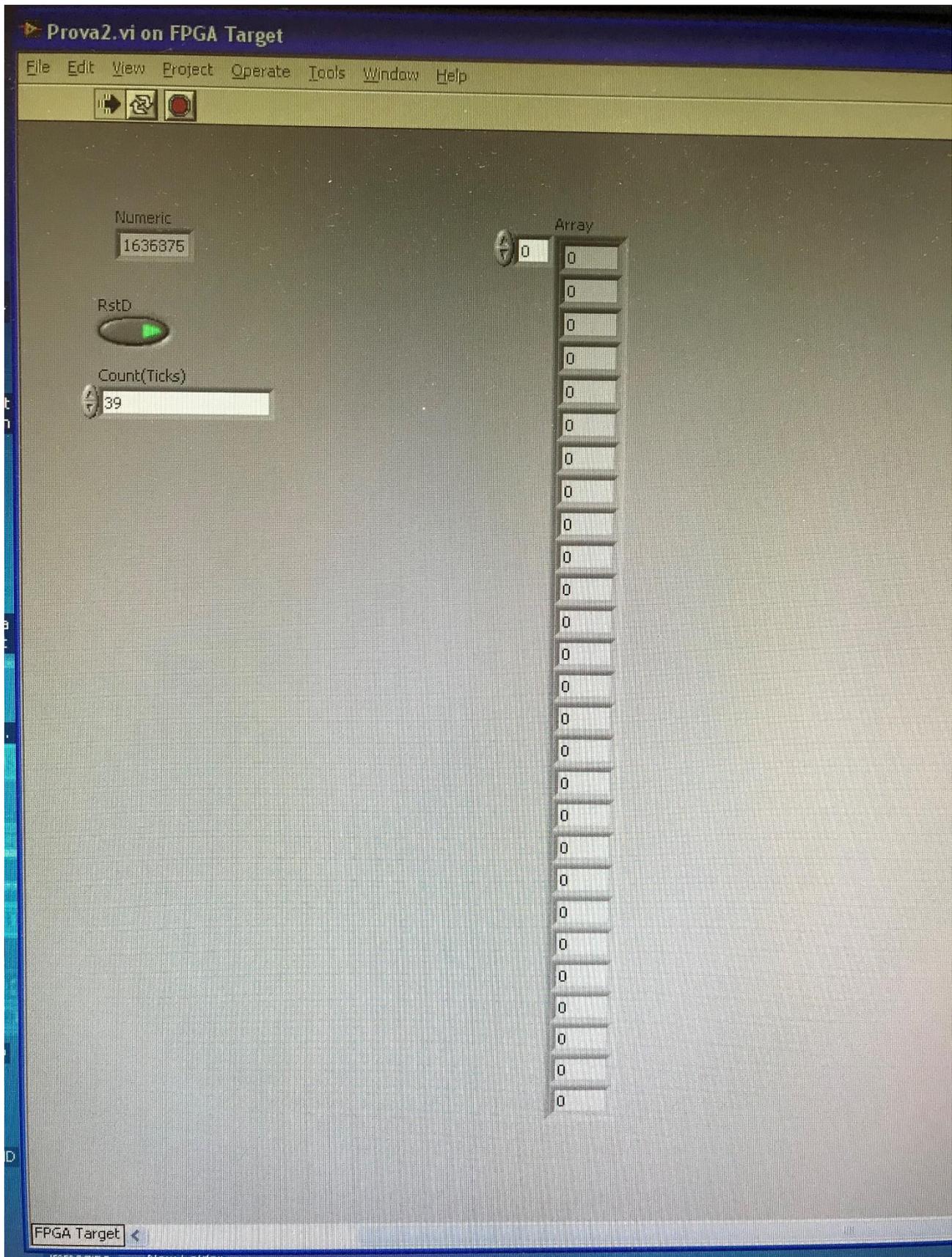


Fig.10.13. Azzeramento del conteggio dei canali in seguito all'accensione del pulsante RstD.

Conclusione

La tesi esposta aspira, dunque, a spiegare ed illustrare il progetto di realizzazione di un monitor fornito di fibre scintillanti per la misurazione di fasci di particelle a bassa intensità. Questo è stato condotto dalla collaborazione dell'INFN della sezione di Roma e della Fondazione CNAO, dove si è concentrato il lavoro svolto in prima persona, che pone come obiettivo la realizzazione del sistema di acquisizione del monitor e del suo interfacciamento con il DDS.

La programmazione del firmware della scheda CAEN – DT5550W / A55CIT4 ha costituito il punto di partenza del lavoro presso il CNAO. La scheda è stata scelta appositamente per gestire l'acquisizione dei dati grazie alla possibilità di montare su essa due matrici di SiPM. È stato utilizzato il software SCI-Compiler, la cui licenza è stata concessa in dotazione con la scheda, per la programmazione dell'FPGA programmabile Xilinx XC7K160T, situata nella scheda. La programmazione della DT5550W / A55CIT4 è avvenuta con successo e ha permesso di avere un'architettura di acquisizione dati molto simile a quella utilizzata storicamente presso il CNAO nel DDS. In particolare, il codice implementato permette di prelevare tutti i 128 canali dei quattro ASIC Citiroc presenti sulla scheda, i quali leggono le informazioni dai SiPM, e presentarli in uscita con un certo criterio e precisione temporale ben definite, sulla base di comandi provenienti dal DDS.

Il secondo passo si è basato sulla progettazione di una scheda elettronica da inserire tra la scheda di acquisizione dati e il DDS, per la conversione del formato dei dati in uscita dalla scheda CAEN da seriale a parallelo (utilizzato nell'elettronica del DDS). I vari step di progettazione della scheda hanno incluso: la scelta dei componenti, il dimensionamento di alcuni di questi, il disegno dello schema elettrico e il disegno del PCB; gli ultimi due passi compiuti con l'ausilio del software KiCad. Dopo la realizzazione del PCB da parte dell'azienda Baselectron S.r.l. e l'acquisto di tutti i componenti, questi ultimi sono stati montati presso il laboratorio di elettronica del CNAO. Una volta che la scheda, il cui nome assegnato in fase di progettazione è DD Low Int, è stata assemblata, il passo successivo è stato testarla singolarmente e in accoppiamento con la DT5550W / A55CIT4.

A seguito dell'esito positivo dei test, il sistema DT5550W / A55CIT4 + DD Low Int è stato collegato al DDS presso la Sala Sperimentale del CNAO. Prima di effettuare il test finale, è stato necessario programmare un'FPGA del DDS per inviare i comandi verso il sistema di acquisizione realizzato e leggere i dati inviati da quest'ultimo al DDS. LabVIEW è stato il software utilizzato in questo step, il quale, oltre a programmare l'FPGA, fornisce anche un'interfaccia per visualizzare e gestire l'acquisizione dei dati.

Il test finale effettuato certifica il corretto funzionamento della catena di acquisizione dei dati. Al termine del mio periodo di permanenza al CNAO, il monitor, la cui progettazione e realizzazione è gestita esclusivamente dal team INFN, non era ancora terminato. Per questo motivo, il sistema di acquisizione realizzato in prima persona, in collaborazione con il Correlatore della presente tesi, non è stato ancora testato con il monitor, anche se è stata effettuata un'acquisizione di dati simulati. Sono state inviate, a livello del firmware della scheda CAEN, una serie di forme d'onda a frequenze fisse diverse in alcuni dei 128 canali di acquisizione, che simulano il passaggio di particelle attraverso il monitor e la loro rilevazione da parte dei SiPM.

In conclusione, il lavoro svolto in prima persona è dedicato ed incentrato sull'elettronica di acquisizione del rivelatore per fasci di bassa intensità e al suo accoppiamento con il DDS: è possibile, quindi, affermare che è stata progettata, realizzata e testata con successo tutta la catena elettronica, partendo dai SiPM della scheda DT5550W / A55CIT4 fino all'acquisizione del profilo del rivelatore dentro il DDS.

Il monitor per basse intensità permetterà alla Fondazione CNAO di possedere nella Sala Sperimentale un sistema in grado di controllare e monitorare erogazioni di fasci di particelle a bassa intensità. Un sistema, quindi, utile ed efficace a livello sperimentale non solo per il CNAO ma anche per soddisfare le richieste di enti terzi che desiderano effettuare test e misure che coinvolgono le basse intensità, presso la Fondazione stessa; un traguardo che sarebbe stato impossibile raggiungere con gli attuali monitor presenti, "ciechi" nei confronti di fasci con dette caratteristiche.

Bibliografia

- [1] Digital Electronics. F.Zappa. Società Editrice Esculapio. (2019; 44-48)
- [2] Fondamenti di campi elettromagnetici. Teoria e applicazioni. Fawwaz T. Ulabi. McGraw -Hill. (2006; Edizione italiana a cura di Stefano Selleri: 33-58)
- [3] Embedded Control and Monitoring Using LabVIEW Course Manual. ni.com. (2015: 14-188)
- [4] FPGAs Fundamentals, Advanced Features, and Applications in Industrial Electronics. Juan José Rodríguez Andina, Eduardo de la Torre Aranz, and María Dolores Valdés Peña. CRC Press. (2017: 21-44)
- [5] Performances of a VLSI wide dynamic range current-to-frequency converter for strip ionization chambers. G.C. Bonazzola, R. Cirio, M. Donetti, F. Marchetto, G. Mazza. C. Peroni, A. Zampieri. Nuclear Instruments and Methods in Physics Research A. (Marzo 1998; 405(1): 111-117).
- [6] Design and test of a 64-channel charge measurement ASIC developed in CMOS 0.35 μm technology. A. La Rosa, G. Mazza, M. Donetti, F. Marchetto, L. Luetto, A. Attili, F. Bourhaleb, R. Cirio, M.A. Garella, S. Giordanengo, N. Givehchi, S. Iliescu, J. Pardo, A. Pecka, C. Peroni, G. Pittà. Nuclear Instruments and Methods in Physics Research A. (Dicembre 2007; 583(2-3): 461-465).
- [7] Characterization of a front-end electronics for the monitoring and control of hadrontherapy beams. A. La Rosa, M. Donetti, M. Borri, F. Rivero, A. Attili, F. Bourhaleb, R. Cirio, M.A. Garella, S. Giordanengo, N. Givehchi, G. Mazza, F. Marchetto, J. Pardo, A. Pecka, C. Peroni. Nuclear Instruments and Methods in Physics Research A. (Febbraio 2008; 586(2): 270-272).
- [8] Design and characterization of the beam monitor detectors of the Italian National Center of Oncological Hadron-therapy (CNAO). S. Giordanengo, M. Donetti, M.A. Garella, F. Marchetto, G. Alampi, A. Ansarinejad, V. Monaco, M. Mucchi, I.A. Pecka, C. Peroni, R. Sacchi, M. Scalise, C. Tomba, R. Cirio. Nuclear Instruments and Methods in Physics Research A. (Gennaio 2013; 698: 202-204).
- [9] The CNAO dose delivery system for modulated scanning ion beam radiotherapy. S. Giordanengo, M. A. Garella, F. Marchetto, F. Bourhaleb, M. Ciocca, A. Mirandola, V. Monaco, M. A. Hosseini, C. Peroni, R. Sacchi, R. Cirio, M. Donetti. Medical Physics. (Gennaio 2015; 42(1): 263-274).
- [10] Dose Delivery Concept and Instrumentation. S.Giordanengo, M.Donetti. CERN Yellow Reports: School Proceedings. (2017; 1: 16-20).
- [11] SiPM for fast Photon-Counting and Multiphoton Detection. P. Eraerds, M. Legre', A. Rochas, H. Zbinden, N. Gisin. Optics Express. (Ottobre 2007; 15 (22): 14540,14541).
- [12] https://www.hamamatsu.com/resources/pdf/ssd/s13361-3050_series_kapd1054e.pdf
- [13] <https://www.caen.it/products/dt5550w/>

- [14] <https://www.caen.it/products/sci-compiler/>
- [15] <https://assets.nexperia.com/documents/data-sheet/74LVC594A.pdf>
- [16] https://assets.nexperia.com/documents/data-sheet/74HC_HCT244.pdf
- [17] <https://www.ti.com/product/SN75451B>
- [18] <https://www.st.com/resource/en/datasheet/cd00000546.pdf>
- [19] <https://recom-power.com/pdf/Innoline/R-78B-1.0.pdf>