

UNIVERSITA' DI PADOVA

FACOLTA' DI INGEGNERIA



FAST WAVEFRONT SENSOR INTERFACE

Relatore : Prof.ssa Maria Guglielmina Pelizzo

Co-Relatore : Ing. Stefano Bonora

Tutor Aziendale : Ing. Cosmo Trestino

Laureando : Alberto Acciari

Corso di Laurea Specialistica in Ingegneria
Elettronica

Anno Accademico

2010-2011

Indice

Sommario	5
1 Introduzione.....	6
1.1 Ottica Adattiva	8
1.2 Aberrazioni.....	11
1.3 Sistemi di Ottica Adattiva.....	18
2 OBIETTIVO DELLA TESI	22
2.1 SENSORE SHACK HARTMANN.....	22
2.1.1 ALGORITMO CALCOLO CENTROIDI	27
2.1.2 Pregi e Limiti algoritmo.....	28
2.2 SVILUPPO ALGORITMO IN MATLAB	29
2.2.1 Inizializzazione	29
2.2.2 Analisi a Regime.....	29
3 ANALISI PRELIMINARI	31
3.1 PUNTI DI CRITICITA'	31
3.1.1 VELOCITA' DI ELABORAZIONE	31
3.1.2 FLESSIBILITA'	32
4 SCELTE PROGETTUALI	33
4.1 Camera Link.....	33
4.1.1 LVDS Descrizione Tecnica.....	33
4.1.2 Channel Link.....	33
4.1.3 Connettore e Cavo di Dimensioni Ridotte	34
4.1.4 Alta Velocità di Trasmissione Dati.....	34
4.1.5 Principio di Funzionamento	34
4.2 FPGA	36
4.2.1 Considerazioni generali di una FPGA	37
4.2.2 Blocchi e parametri caratteristici.....	38
4.2.3 Cluster Logic Block	38
4.2.4 Connection module.....	41

4.2.5	Switch module.....	41
4.2.6	Interconnessioni.....	42
5	SVILUPPO DEL PROGETTO	45
5.1	FASE 2	47
5.1.1	MS2.....	49
5.1.2	DIVs.....	59
5.1.3	RAM1_INTERFACE	65
5.2	FASE 1	75
5.2.1	STEP_1.....	87
5.2.2	CORR_SUB_X/Y.....	90
5.2.3	CONTROL.....	91
6	Conclusioni.....	92
7	Bibliografia	93
	APPENDICE A.....	94
	Ringraziamenti	99

Sommario

L'argomento trattato in questa tesi, si propone di risolvere una delle problematiche che attualmente si possono riscontrare nell'utilizzo dei sensori fronte d'onda, basati sulla tecnologia Shack-Hartmann.

Il sempre crescente sviluppo tecnologico che i suoi componenti hanno subito nel corso di questi ultimi anni, hanno portato alla richiesta di una potenza computazionale maggiore.

Durante la mia tesi che si è svolta presso l'azienda Adaptica s.r.l., ho sviluppato l'implementazione di un particolare algoritmo, per l'estrapolazione dei centroidi, su piattaforma FPGA.

Grazie a questa interfaccia, sarà possibile collegarsi direttamente ai sensori fronte d'onda, ricevendo, in uscita, solamente i dati utili; ossia, le coordinate dei centroidi e non più le immagini da dover elaborare come solitamente avviene.

L'intero progetto è stato sviluppato e testato su ambiente Xilinx e nei prossimi mesi verrà montato un primo prototipo sperimentale per testarne la validità.

La tesi è stata sviluppata nei seguenti capitoli:

nel cap.1 viene proposta un'ampia visione della materia di applicazione e degli utilizzi dei sensori fronte d'onda;

nel cap.2 viene illustrato il principio di funzionamento dei sensori Shack-Hartmann, inoltre si andrà a descrivere l'algoritmo di detection utilizzato, fornendo uno script in matlab utilizzato per validarne il funzionamento;

nel cap.3 viene fatta un'analisi preliminare del progetto focalizzando l'attenzione sulle scelte da fare per ottimizzare lo sviluppo dell'interfaccia;

nel cap.4 viene fatta una panoramica delle tecnologie scelte per lo sviluppo del progetto, illustrandone particolarità e caratteristiche;

nel cap.5 comincia la spiegazione sulla tipologia di architettura scelta per l'implementazione dell'algoritmo, descrivendone dettagliatamente le diverse fasi;

nel cap. 6 le conclusioni e le prospettive legate al progetto.

1 Introduzione

Nei secoli, gli strumenti per l'osservazione del cielo sono cambiati enormemente ... e se fino a Galilei l'occhio nudo era l'unico mezzo per osservare gli astri, oggi ben diversa è la situazione.

La scienza astronomica, come ogni altra, si appoggia costantemente alla tecnologia, è legata ad essa tramite un filo indissolubile: ogni progresso tecnologico può portare a moltitudini di piccole scoperte astronomiche, quando non addirittura a nuove teorie, o ad ulteriori conferme delle precedenti.

La nascita del primo cannocchiale rifrattore si fa comunemente risalire al 1608, ad opera di artigiani olandesi. Già a partire dall'anno successivo, in cui Galileo costruì il suo celebre strumento, il cannocchiale fu "ufficialmente" puntato verso la volta celeste. Pochi anni dopo, nel 1611, Keplero propose per la prima volta il telescopio riflettore con cui si potevano avere, allo stesso costo, aperture maggiori, seppur con alcuni inconvenienti sulla capacità risolutiva e sulla qualità dell'immagine. Nel corso dei secoli questi strumenti sono stati tecnicamente molto perfezionati, sia nella sezione propriamente ottica sia in quella meccanica, altrettanto importante in operazioni delicate come la messa a fuoco o lo spostamento preciso dello strumento.

Nell'ultimo secolo, nuovi passi da gigante sono stati fatti nella tecnologia astronomica: basti pensare allo sviluppo delle tecniche fotografiche, alla nascita della radioastronomia, ai telescopi orbitanti che nemmeno risentono della distorsione dovuta all'atmosfera.

Grazie all'analisi spettrale, è oggi possibile studiare il movimento degli astri, e perfino stabilirne la composizione ed ecco nascere, accanto all'astronomia, l'astrofisica. Tutte le radiazioni elettromagnetiche possono essere rilevate e studiate, anche al di fuori della banda del visibile.

Negli ultimi decenni, lo sviluppo dell'elettronica e dell'informatica ha rivoluzionato anche il mondo dell'astronomia. Enorme è stato lo sviluppo dei sistemi elettronici di ausilio ai grandi telescopi; calcolatori sempre più potenti permettono l'elaborazione di masse di dati sempre più consistenti; la stessa acquisizione delle immagini viene ormai fatta non più solo tramite l'occhio umano o la lastra fotografica, ma anche e soprattutto tramite sensori elettronici, sempre più spesso basati sulla tecnologia del silicio.

Vi sono scienziati per i quali "scrutare il cielo", oggi, significa soprattutto analizzare dati numerici, interpretarli con l'ausilio dei calcolatori, elaborare rigide strutture

matematiche; ma lo fanno mossi dallo stesso istinto del bambino affascinato che alza gli occhi al cielo stellato: la curiosità per l'ignoto, la gioia della scoperta.

Il lavoro da me svolto nell'ambito del progetto assegnatomi durante il mio tirocinio presso Adaptica s.r.l. s'inserisce proprio in tale contesto di sviluppo tecnologico rivolto ad applicazioni astronomiche.

In particolare il progetto è nato mirando allo sviluppo di un sistema di acquisizione e di prima elaborazione dati, del fronte d'onda, basato su sensori Shack-Hartmann, veloce, flessibile e a basso costo.

Capace di svincolare gli attuali limiti dati dall'enorme mole e dall'alta frequenza di acquisizione, necessari a migliorare la precisione e l'accuratezza dei sistemi a loop chiuso in cui questi sensori sono utilizzati.

1.1 Ottica Adattiva

Descrizione della materia nel corso degli ultimi decenni l'ingegneria ottica finalizzata all'imaging dei telescopi di grandi dimensioni è divenuta sempre più sofisticata. Con il trascorrere del tempo una gran parte di telescopi a terra di grandi dimensioni sono stati progettati e realizzati con caratteristiche talmente spinte da riuscire ad essere paragonabili in buone condizioni osservative ai telescopi spaziali, azzerando (o quasi) i difetti riscontrati sulle immagini scientifiche imputabili rispettivamente alla deformazione degli specchi indotta dal loro peso non indifferente e quelli, ancora più fastidiosi e difficili da rimuovere, causati dai moti turbolenti atmosferici. Tutti questi problemi in orbita, per definizione, non esistono, in quanto il peso risulta annullato dalla forza apparente centrifuga, la quale bilanciando la forza di gravità permette al telescopio (satellite) di rimanere in orbita senza decadere; inoltre un telescopio situato nello spazio, essendo fuori dall'atmosfera, non risente di tutte quelle turbolenze introdotte dalle correnti d'aria e dal movimento delle formazioni nuvolose, che affliggono inesorabilmente tutti telescopi terrestri. Come appena detto, per realizzare un telescopio a terra è necessario un certo quantitativo di finanziamenti, i quali nel caso di un telescopio spaziale crescono esponenzialmente; per non parlare delle missioni di riparazione e/o mantenimento indispensabili, ognuna con costi spropositati. Nonostante quindi fosse possibile escludere a priori tutti questi problemi osservativi, era preferibile cercare delle soluzioni tecnologiche per compensare ed azzerare tutti i difetti osservativi nocivi derivanti dal posizionamento di un telescopio a terra. Con il passare del tempo le più grandi menti si sono adoperate al fine di risolvere questi problemi e piano piano hanno ottenuto delle soluzioni sempre più perfezionate e funzionali. In questa carrellata di innovazioni tecnologiche basti pensare al primo telescopio con Ottica Attiva, il New Technology Telescope (NTT), che sfruttava la tecnologia degli specchi sottili, i quali, tramite degli attuatori piezo-elettronici, potevano compensare la deformazione dello specchio indotta dal suo peso, riproducendo una superficie ottimale teorica (paraboloide) a qualsiasi inclinazione dello specchio. Questo telescopio fu a tutti gli effetti una pietra miliare dell'astronomia da terra a cui tutti i successivi strumenti si conformarono (anche il TNG si basa su questa tecnologia); in particolare i suoi fratelli maggiori Very Large Telescopes. Nonostante l'ottica attiva permettesse di eseguire delle osservazioni con la migliore conformazione ottica possibile, le osservazioni da terra erano comunque affette dalle variazioni del tempo meteorologico che, anche quando è sereno e limpido può non essere adatto a delle osservazioni scientifiche. La prima cosa che si notò fu che il luogo fisico in cui i telescopi andavano costruiti doveva essere al riparo dagli agenti atmosferici poiché quando una corrente d'aria investe la cupola di un telescopio tende a creare dei mini-vortici sulla sua sommità e questo fenomeno degrada in maniera consistente l'immagine. Tra i siti privilegiati per

costruire dei telescopi c'è sempre stata l'alta montagna poiché se è vero che l'atmosfera produce gran parte dei problemi osservativi, posizionare un telescopio a 2000-3000 metri di fatto esclude una parte di atmosfera all'osservazione e la stessa percentuale di aria risulta molto meno densa e rarefatta, quindi gran parte dei fenomeni connessi all'assorbimento atmosferico di certe lunghezze d'onda vengono meno. Purtroppo sulla scia di questo ragionamento, molto spesso furono realizzati dei telescopi sulla sommità di montagne, pensando di poter accedere ad un maggiore angolo solido osservativo, ma questo, oltre ad avere un disastroso impatto visivo ed ecologico, comportava una continua turbolenza sistematica causata dal profilo frastagliato delle montagne ed anche dalle cupole dei telescopi in questione, che solitamente erano a forma sferica. Dopo questi continui fallimenti si comprese che i migliori siti osservativi non erano le montagne, ma gli altipiani (possibilmente molto ampi, regolari e desertici) dove le correnti d'aria si potevano disporre in maniera laminare senza creare turbolenze e vortici. Inoltre già dal NTT in poi le cupole cominciarono a non essere più di forma sferica, ma a forma di grandi scatole squadrate con le cupole completamente aperte e le pareti dotate di lamelle di ventilazione in modo da ridurre al minimo l'ostacolo del telescopio e della cupola al vento e riprodurre nel modo più efficace possibile l'andamento laminare delle correnti d'aria. Il sito ottimale per le osservazioni scientifiche quindi doveva essere un altipiano riparato da una cordigliera montuosa da una parte (in modo da bloccare sia le perturbazioni sia le correnti d'aria umide) e dalla parte opposta doveva essere aperto verso grandi praterie sconfinite (avevano il ruolo di prosciugare le correnti d'aria (alisei), privandole del vapor d'acqua). Fu da uno studio apposito che l'ESO (European Southern Observatory), scoprì La Silla prima e Cerro Paranal poi, come i migliori siti osservativi sul pianeta situati su un altipiano a 4000/5000 metri di altitudine sul livello del mare. Entrambi i siti protetti dalle grande cordigliera delle Ande cilene ad Ovest e disposti ad EST verso le grandi pianure (Pampas) argentine. In questi siti le precipitazioni sono scarsissime ed il cielo è quasi sempre adatto alle osservazioni astronomiche. Nonostante si utilizzasse il miglior telescopio (VLT) nel miglior sito osservativo (Cerro Paranal appunto), i telescopi spaziali erano ancora un gradino avanti come prestazioni e potere risolutivo ai migliori telescopi da terra. Gli scienziati non si persero d'animo e svilupparono delle tecnologie via via più sofisticate fino a che nacque il primo prototipo di Ottica Adattiva. L'ottica adattiva funziona grosso modo come l'ottica attiva in cui lo specchio viene deformato dalla forma teorica (garantita dai cicli di ottica attiva) sempre tramite degli attuatori piezo-elettronici, disposti su una rete più fitta. A differenza dell'ottica attiva, il cui tasso temporale di correzione è minimo, l'ottica adattiva agisce introducendo le deformazioni sugli specchi secondari ad intervalli temporali tipici della variazione della forma degli oggetti (variabilità del seeing). Tramite la combinazione di ottica attiva ed ottica adattiva è possibile modificare in tempo reale il fronte d'onda incidente in modo da compensare le deformazioni introdotte dalla turbolenza

atmosferica. La realizzazione fisica di questo apparato scientifico è semplice ma al contempo complessa: si proietta una stella artificiale sulla volta celeste e si analizza il profilo di luce con lo scorrere del tempo. Successivamente paragonando la forma d'onda (sfericità, elongazione, concentrazione ecc...) osservata in cielo con quella proveniente dalle osservazioni a cupola chiusa, è possibile introdurre delle micro variazioni sulla forma dello specchio secondario per modificare interattivamente il profilo di luce risultante ed ottenere sempre le stelle di guida più puntiformi possibile.

1.2 Aberrazioni

La trattazione delle aberrazioni e le modalità per la loro correzione assumono notevole importanza nell'ottica moderna.

L'aberrazione di un sistema ottico può essere definita come la differenza tra l'immagine effettiva, reale o virtuale, formata dal sistema e l'immagine che si voleva ottenere, immagine che di solito è bidimensionale e consiste in una proiezione geometrica della scena reale sul piano focale del sistema secondo i principi dell'ottica geometrica ideale. Le aberrazioni possono dare (di solito più sulla periferia dell'immagine che al suo centro) scarsa nitidezza, deformazioni dell'immagine, differenze tra le immagini corrispondenti ai diversi colori, non uniformità della luminosità.

Oltre che a imperfezioni dei componenti o dei materiali che costituiscono il sistema ottico, le aberrazioni possono esser dovute, in maniera più essenziale, al fatto che il sistema si allontana dalle condizioni dell'ottica parassiale, quando raccoglie raggi ad angoli grandi o li devia di molto (cosa che succede per esempio in ottiche grandangolari o con grande rapporto diametro/distanza focale). Possono essere ridotte, a volte quasi completamente eliminate, utilizzando materiali migliori, lavorando in modo particolare le ottiche, accoppiando molti componenti, ecc. In generale la correzione comporta un aumento dei costi di produzione.

In pratica un'immagine formata da un insieme di onde elettromagnetiche visibili può essere deviata, ridotta o aumentata o altro, senza modificare le disposizioni relative delle singole onde. Ad esempio una foto può essere definita come una visione trasposta di una scena reale che rimane identica proporzionalmente e relativamente a sé stessa ma che poi può essere spostata in relazione all'esterno. Quando questo tipo di immagine "riportata" non è perfetta e non svolge al meglio la sua funzione di trasposizione, spesso è perché sono presenti delle aberrazioni.

Un elemento influenzante lo sviluppo dell'aberrazione è lo spessore del mezzo ottico attraversato dalla luce, dalla scomposizione di quest'ultima legata al fenomeno della rifrazione ed alla suddivisione nelle diverse lunghezze d'onda dei colori percepiti nel visibile.

Le aberrazioni si dividono in 2 grandi famiglie, quelle assiali e quelle extra-assiali, a seconda che si verifichino o meno lungo l'asse ottico del sistema.

Le principali aberrazioni assiali sono:

- **Aberrazione Cromatica**
- **Aberrazione Sferica**

Le principali aberrazioni extra-assiali sono invece:

- **Coma**
- **Astigmatismo**
- **Distorsione**

Aberrazione Cromatica

Quando si parla di indice di rifrazione di un vetro, comunemente indicata con n , bisogna sempre riferirla a una precisa lunghezza d'onda (λ). Se pertanto sulla superficie di separazione di un diottro incide un raggio di luce policromatica questo subirà una dispersione a causa del diverso valore di n per le varie componenti

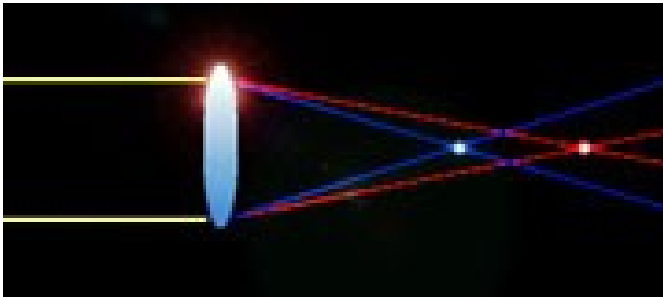


Figura 1 – Aberrazione Cromatica

luminose: maggiore per quella azzurra (raggio più rifratto), minore per quella rossa (raggio meno rifratto).

Lo stesso si verifica facendo passare un fascio parallelo di luce bianca (ad esempio la luce del Sole o di una

lampadina a incandescenza) attraverso una lente: esso focalizzerà in due posizioni diverse, dal momento che la focale della lente sarà minore per la luce blu e maggiore per quella rossa. Il risultato di questo è che se si punta una stella l'immagine presenterà una fastidiosa colorazione, cangiante a seconda della porzione dello spettro che andiamo a focalizzare. Quando tra la fine del '600 e l'inizio del '700 non erano ancora disponibili i doppietti acromatici, ottenuti accoppiando vetri di diversa dispersione, l'unico sistema per ridurre al minimo l'aberrazione cromatica era quella di aumentare enormemente la focale della lente. Se infatti quest'ultima è uguale 20 volte il quadrato del diametro ($F = 20 \times \varnothing^2$) il cromatismo non si nota. In



Figura 2

compenso si notano...le dimensioni dello strumento! Per dare un'idea, se un cannocchiale di soli 3 cm può mantenere una focale al di sotto dei 2 metri, uno da 10 cm, dovrebbe essere lungo una 20-ina di metri! Non sarebbe neppure possibile costruire tubi tanto lunghi e stretti perché tenderebbero a flettersi, vanificando così l'osservazione. Questo problema era stato risolto dai grandi pionieri del passato, fra i quali Huygens e Cassini, con l'utilizzo dei telescopi aerei, costituiti da semplici lenti fissate in cima a un palo mediante un complicato sistema di pulegge e da un oculare unito all'obbiettivo da un filo volante che permetteva all'osservatore di tragarne l'immagine. Un sistema scomodo e precario che non di meno permise al Cassini la scoperta della divisione dell'anello di Saturno che porta il suo nome. Dopo l'invenzione del doppietto acromatico, costituito dai classici vetri crown e flint a diversa dispersione, il problema della aberrazione cromatica è stato in gran parte risolto. Rimane tuttavia una considerazione da fare. Anche studiando la curvatura dei due vetri, nonché il loro corretto posizionamento reciproco di modo da portare a coincidenza le righe blu e rosse dello spettro, le rimanenti righe rimangono leggermente sfasate, in quanto la correzione cromatica del flint non è lineare per tutte le lunghezze d'onda. Ciò conduce al fatto che foccheggiando per la λ cui l'occhio è maggiormente sensibile, ossia 0.56 micron, la luce rossa e, in maggior misura, quella blu resteranno leggermente sfocate. Questo effetto, noto come spettro secondario, si manifesta con una colorazione bluastra attorno alle stelle più brillanti o al bordo di soggetti luminosi come la Luna o i pianeti. Anche in questo caso è possibile rendere innocuo il cromatismo residuo indotto dallo spettro secondario a patto che il rifrattore possieda una focale almeno pari al quadrato del diametro (ossia $F = \varnothing^2$). Ciò comporta una focale di un metro per un'apertura di 10 cm, il che rende l'ingombro del telescopio assolutamente accettabile. Per avere una correzione cromatica superiore a vantaggio di un ingombro ancora minore si può ricorrere a un obbiettivo costituito da 3 elementi, che come tale risulterà corretto per 3 lunghezze d'onda, o all'impiego di vetri particolari a bassissima dispersione (i cosiddetti ED) sino ad arrivare all'utilizzo di certi cristalli monometrici (cioè esenti da birifrangenza) come la fluorite; in questi ultimi casi si hanno gli obbiettivi apocromatici che presentano una correzione e un'incisività delle immagini superbe.

Aberrazione Sferica

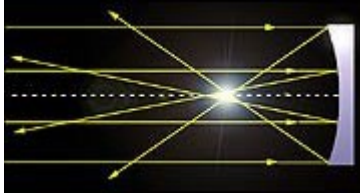


Figura 3 – Aberrazione Sferica

Per focalizzare in un punto l'immagine di una stella servendosi, stavolta, di uno specchio, è necessario che questo sia a sezione parabolica. Il motivo è che la parabola, come tutte le curve coniche, possiede 2 fuochi, solo che in questo caso uno dei due si trova all'infinito. Siccome le stelle sono situate virtualmente all'infinito è ovvio che i raggi paralleli che incidono sulla parabola vanno, per definizione, a convergere nel fuoco; se però i raggi arrivano su uno specchio sferico, anziché parabolico, quelli più vicini all'asse vanno a fuoco più lontano di quelli periferici; il fuoco in questo caso non è definito, ma delimitato da una figura simmetrica complessa chiamata caustica di riflessione. Tuttavia è più facile, nonché più economico, costruire uno specchio sferico, piuttosto che uno parabolico; oltre tutto lo specchio sferico, non avendo assi ottici preferenziali (di fatto ne possiede infiniti!), è totalmente esente da aberrazioni extra-assiali come il coma. Il diffusissimo newtoniano da 114 / 900, che ha segnato una tappa importante nella storia di molti astrofili, fa appunto uso di uno specchio sferico e le immagini offerte sono otticamente molto corrette. È però importante che sia soddisfatta la seguente relazione:

$$F = 3 \cdot \varnothing^{4/3}$$

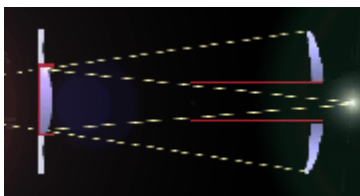


Figura 4 - Newton-Cassegrain

ossia la focale, espressa in centimetri, deve essere almeno uguale a 3 volte il diametro elevato alla potenza di 4/3, altrimenti l'aberrazione diventa molesta; normalmente questa si presenta come un marcato alone circolare attorno alle stelle più brillanti.

Per correggere l'aberrazione sferica mantenendo, al contempo, lo specchio con un alto rapporto di apertura si utilizza la configurazione Schmidt-Cassegrain o Newton-Cassegrain: la luce, prima di giungere allo specchio primario, passa attraverso un correttore opportunamente sagomato di modo da introdurre nello schema ottico una caustica di forma uguale e contraria.

Coma



Figura 5 -Coma

Viene talvolta definito anche aberrazione sferica trasversale, in quanto i raggi periferici anche in questo caso vengono focalizzati diversamente da quelli centrali. Il coma è piuttosto fastidioso, ma inevitabile in newtoniani puri a largo campo, quando cioè vengono impiegati rapporti di apertura forzati come F/4 o F/5 e si osserva lontano dall'asse ottico; l'accentuazione del coma avviene linearmente ed è quindi proporzionale sia al quadrato del rapporto d'apertura, sia all'inclinazione dei raggi. Visualmente, una stella soggetta a coma se opportunamente sfuocata si presenta come nella figura a destra. Per uso fotografico esistono ad ogni modo delle lenti correttive che inserite nel porta oculare svolgono una funzione analoga al correttore di Schmidt nei telescopi a specchio sferico.

Astigmatismo

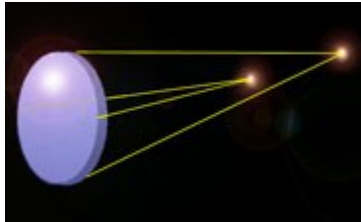


Figura 6 -Astigmatismo

È un fenomeno molto comune che può verificarsi anche lungo l'asse ottico dello strumento, ma in questo caso ci troveremmo di fronte a un vero e proprio difetto di lavorazione dell'obiettivo. L'astigmatismo, infatti, si verifica quando uno specchio sferico (o parabolico) presenta curvature diverse nelle diverse direzioni. Se osservate lo schema della figura di sinistra (potete ingrandirla) noterete che i raggi che incidono sullo specchio in un piano orizzontale vanno a fuoco più vicini di quelli che invece si trovano sul piano verticale. Se dunque si foceggia l'immagine stellare relativamente al primo piano, questa apparirà con dei vistosi baffi in senso verticale; se invece si foceggia in corrispondenza del 2° fuoco (quello più esterno) i baffi appariranno orizzontalmente. Istintivamente l'osservatore cercherà il miglior compromesso tra le 2 posizioni, col risultato che le stelle appariranno crocettate. L'astigmatismo si può individuare facilmente sfocando l'immagine e posizionando alternativamente l'oculare in posizione intra ed extra focale. Il risultato, molto schematico, è riprodotto nella figura di destra. Nelle ottiche ben lavorate quest'aberrazione si nota solitamente con aperture modeste in strumenti a grande campo; a differenza del coma infatti l'astigmatismo è proporzionale al diametro dello strumento (e non al suo quadrato), ma al quadrato dell'inclinazione dei raggi, diversamente dal coma che varia invece

linearmente con l'obliquità. Significa che l'astigmatismo diventa preponderante soltanto a grandi distanze dall'asse ottico, mentre a piccole distanze prevale il coma. Bisogna tuttavia sottolineare che, seppur fastidioso, un leggero astigmatismo a centro campo non compromette osservazioni precise come le misure astrometriche, in quanto l'immagine, a differenza del coma, mantiene sempre la sua simmetria. Ad ogni modo, una delle cause frequenti di astigmatismo sono le tensioni indotte sulla superficie dello specchio. Se dovete rimuovere il primario per pulirlo o per un intervento alla cella, assicuratevi, dopo averlo riposto, che le clip di sicurezza non

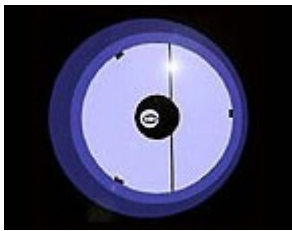


Figura 7

siano mai a contatto dello specchio; quest'ultimo deve quindi essere sempre in grado di ruotare attorno al suo alloggiamento. Una cosa importante per prevenire effetti di coma o astigmatismo è anche il corretto allineamento delle ottiche. I grandi dobsoniani con rapporti di apertura forzati e che vengono di sovente trasportati in montagna alla ricerca di siti idonei sono particolarmente sensibili al disallineamento; è quindi importante sistemare lo strumento in modo corretto per ottimizzarne le prestazioni. Indicativamente, un'ottica aperta a $F/4$ deve mostrarsi come in figura (supposto che l'oculare, guardando da fronte, si trovi sulla sinistra); lo specchio secondario, e di conseguenza le varie riflessioni, devono apparire in modo leggermente eccentrico. Ovviamente un elevato grado di precisione si ottiene mediante allineamento laser o tramite appositi oculari denominati Cheshire. Si tratta indubbiamente di finezze che tuttavia si possono facilmente bypassare operando una correzione fine delle ottiche, fine tuning in inglese, direttamente sul campo puntando la Polare, l'unica stella del cielo a restare ferma.

Distorsione

È un'aberrazione tipica dei comuni binocoli commerciali che si verifica ai margini del campo. È noto che i binocoli, essendo concepiti prevalentemente per l'uso terrestre, hanno ingrandimenti limitati e, conseguentemente, un grande campo di veduta, di



Figura 8 - Distorsione

solito non inferiore ai 4 o 5 gradi (esistono numerosi binocoli concepiti per l'osservazione astronomica che essendo di gran lunga più costosi adottano una serie di accorgimenti tecnici per limitare la distorsione ai bordi, ma è un discorso a parte). A questo di aggiunga che i binocoli più economici montano usualmente degli oculari Kellner che, pur essendo acromatici, sono costituiti da 3 sole lenti le quali non sono in grado di compensare le carenze introdotte dell'obbiettivo.

Come nel caso delle aberrazioni di cui abbiamo già trattato, anche la distorsione non va comunque vista come un difetto dello strumento, ma come una caratteristica insita nel fatto che un semplice doppietto acromatico a corta focale, tipico dei binocoli, non riesce a fornire un ingrandimento uniforme su tutto il campo. Ci può consolare il fatto che se il coma e l'astigmatismo sono proporzionali rispettivamente all'inclinazione e al quadrato dell'inclinazione dei raggi, la distorsione varia invece col cubo di tale inclinazione, il che spiega perché si verifichi di solito ai bordi del campo di un binocolo, ma assai raramente in quello di un telescopio.

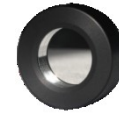
È possibile correggere queste anomalie presenti nel fronte d'onda attraverso sistemi a loop-chiuso nei quali vengano introdotti ed applicati i principi su cui si basa l'Ottica Adattiva.

1.3 Sistemi di Ottica Adattiva

I componenti usualmente utilizzati in sistemi di OA sono:

SPECCHIO DEFORMABILE (DM):

Ci sono diversi tipi di specchi deformabili (DMs), alcuni sono già presenti nel mercato, altri sono ancora nei laboratori di ricerca. I parametri che caratterizzano uno specchio deformabile sono:



1. Tecnologia attuativa: meccanica, piezoelettrica, elettrostatica (membrana), magnetica, bimorfa, MEMS.
2. Qualità della Superficie:
 - a) segmentata, fatta di piccole sezioni di specchio, ognuna mobile separatamente
 - b) continuo, costituito da una singola superficie riflettente che può essere mossa in differenti punti. In ogni caso la qualità ottica deve garantire l'effettiva correzione desiderata.
3. Numero di attuatori: il numero di attuatori definisce la qualità e il numero di diverse forme che lo specchio può riprodurre
4. Dimensione: si può andare da DMs da alcuni millimetri, a specchi deformabili secondari per telescopi, da decine di centimetri.

Ogni combinazione ha vantaggi e svantaggi riguardanti i costi, consumo di potenza, soglia di danneggiamento, isteresi e ripetibilità. Il particolare le specifiche degli specchi a membrana sono: basso costo, ampia dinamica di comportamento, acromaticità, assenza di isteresi, soglia di danneggiamento relativamente alta, buone prestazioni nella generazione di aberrazioni, basso consumo di potenza. Gli svantaggi di questi dispositivi sono la limitata quantità di massimo stroke (massima deformazione ..) e l'alta correlazione tra gli elettrodi. Per ovviare a questo problema si possono pensare altre soluzioni, come mettere gli attuatori nella parte superiore della struttura, così da essere in grado sia di tirare che spingere la membrana.

SENSORE FRONTE D'ONDA

Il sensore di fronte d'onda è usato per misurare la deformazione del fronte d'onda. Ci sono diverse tecniche che possono essere utilizzate allo scopo di ottenere questa informazione. La più comune è l'approccio Shack Hartmann: esso consiste nel campionamento spaziale del fronte d'onda, per mezzo di un lenslet (array di microlenti). Le posizioni degli spot prodotti dalle lenti sul CCD rivelano l'informazione sulla fase locale del fronte d'onda. Quest'ultimo può essere ricostruito usando questa informazione, o tramite una decomposizione modale. Altre tecniche sono il Sensore di Curvatura, l'Interferometria a Cammino Comune, sensore di fronte d'onda a piramide.

ELETTRONICA DI CONTROLLO

L'elettronica di controllo rappresenta il "core" del sistema. Ha il compito di fare da ponte fra analisi del fronte d'onda e controllo dello specchio deformabile. Preleva e analizza i dati provenienti dal sensore fronte d'onda effettuando su di essi una serie di algoritmi in grado di stimare gli errori, ossia le aberrazioni presenti, e pilotare gli attuatori dello specchio deformabile con le opportune tensioni in modo tale da chiudere il loop.

Il più utilizzato dei sistemi a loop-chiuso in OA vede l'inserimento fra sorgente ed utente finale di un blocco in retroazione come illustrato in figura:

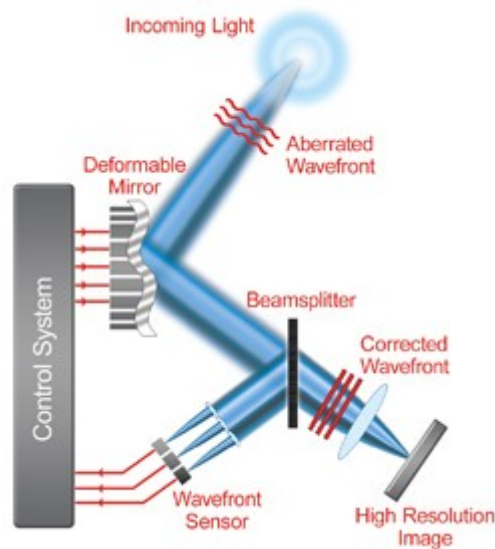


Figura 9 – Loop di Controllo

L'applicazione illustrata viene utilizzata in astronomia per migliorare la visione dei corpi celesti.

Il principio di funzionamento è molto semplice:

Il telescopio riceve il fronte d'onda aberrato, introdotto durante il suo passaggio attraverso gli strati dell'atmosfera terrestre, proveniente dalla stella. Questo viene di fatto riflesso da uno specchio deformabile controllato in retroazione da un'opportuna elettronica di controllo.

Il fascio riflesso viene "sdoppiato", da un ottica particolare chiamata "Beam Splitter", capace di creare due immagini speculari e orientate l'una dall'altra di 45°.

Il fascio orizzontale sarà indirizzato verso l'utente finale, mentre l'altro fascio sarà analizzato da un particolare **sensore di fronte d'onda** , basato sulla tecnologia Shack Hartmann, capace di analizzare il fronte d'onda e di estrapolarne, tramite considerazioni di carattere geometrico, informazioni necessarie a correggerlo.

I dati vengono quindi analizzati e convertiti, dalla successiva elettronica di controllo, in segnali ad alta tensione capaci di deformare correttamente lo specchio deformabile e quindi correggere l'aberrazione del fascio.

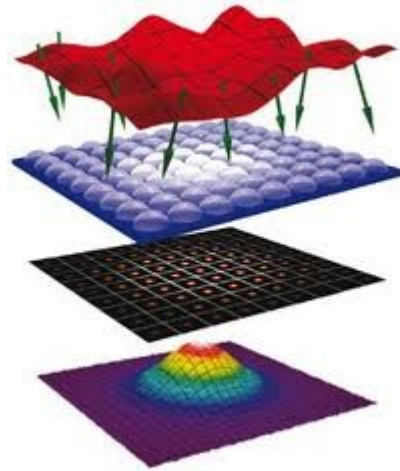


Figura 10 –Analisi Shack-Hartmann

In questo modo è quindi possibile “eliminare” le aberrazioni introdotte nel fascio di luce, riportandolo quindi, il più possibile, ad un livello simile alle condizioni iniziali. L’utente potrà quindi osservare l’oggetto in questione con una qualità e nitidezza paragonabile a quella ottenuta dall’osservazione dell’oggetto da un telescopio satellitare.

2 OBIETTIVO DELLA TESI

Durante il periodo di tirocinio presso la sede di Adaptica s.r.l. mi è stato proposto, come tema, lo sviluppo di un'interfaccia hardware capace di scavalcare l'insostenibile sviluppo, in termini di prestazioni e risoluzione, dei sensori di fronte d'onda basati sulla tecnologia Shack-Hartmann.

2.1 SENSORE SHACK HARTMANN

Il sensore fronte d'onda Shack-Hartmann è un semplice ed elegante metodo di misurazione della forma del fronte d'onda. Questa tecnica ha trovato utilizzi in un'ampia varietà di applicazioni per migliorare, ad esempio, la visione della vastità del cosmo dai telescopi terrestri.

Questa tecnica si è diffusa in tutto il mondo con centinaia di milioni di immagini astronomiche che beneficiano del processo di miglioramento della visione. È raro che una tecnologia abbia un impatto drammatico su un singolo campo, per non parlare più campi, come è successo per il sensore di Shack-Hartmann.

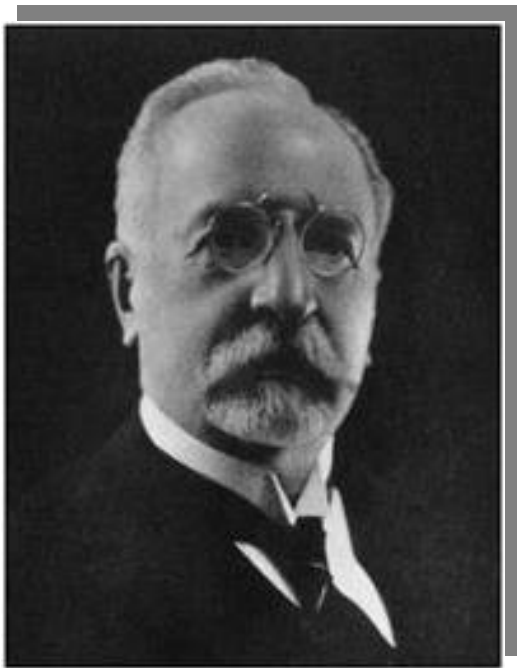


Figura 11 - Hartmann

Il sensore di Shack-Hartmann è nato durante la Guerra Fredda dagli sforzi degli Stati Uniti. L'Air Force aveva il desiderio di vedere i satelliti telescopi terrestri. Purtroppo la turbolenza atmosferica distorceva la qualità delle immagini ottenute. I militari avvicinarono, quindi, Aden Meinel e l'Optical Sciences Center (OSC) presso l'University of Arizona con lo scopo di intraprendere un progetto che avrebbe risolto il problema dell'imaging. È stato Meinel che per primo ha suggerito di attingere alla luce incidente e di utilizzare il test di Hartmann per determinare le aberrazioni atmosferica al momento della cattura dell'immagine del satellite. Conoscendo l'entità di queste aberrazioni, l'immagine risultante potrebbe essere trattata per migliorarne la qualità. A Roland Capanna, che Meinel aveva reclutato per OSC alcuni anni prima, fu dato il compito di determinare se questa tecnica era percorribile.

Molte limitazioni della tecnica di Hartmann divennero subito evidenti. I satelliti inviano poca luce ai telescopi terrestri. La maggior parte di questa luce è necessaria

per salvare l'immagine del satellite, in questo modo solo una piccola frazione di la luce disponibile può essere inviato allo schermo Hartmann. La schermata di Hartmann, inoltre, riduce drasticamente la quantità di luce, ma i fotoni passano attraverso i suoi buchi. La prima innovazione di Shack è stata quella di posizionare lenti all'interno di ciascuno dei fori nello schermo Hartmann. Con l'aggiunta di lenti, la luce che passa attraverso le aperture si sarebbe concentrata in un punto focale. Questa concentrazione consentì di aumentare e stimolare la densità di fotoni consentendo di essere registrati. La seconda intuizione di Capanna era che lo schermo stesso non era più necessario. Se il diametro delle lenti si fosse espanso fino ai bordi, allora tutti i fotoni incidenti sullo schermo avrebbero trovato la loro strada in un punto focale. Di conseguenza, questa configurazione consente l'uso più efficiente della luce incidente, una situazione ideale per un sistema che soffre di troppi pochi fotoni. Il prossimo passo nell'attuazione della tecnica era quello di ottenere una matrice adeguata di lenti. Quest'ultima problematica, nel tempo, ha trovato sempre più soluzioni tecnologiche in grado di massimizzare e garantire le condizioni di realizzazioni migliori alla realizzazione dei lenslet.

In parallelo con il Dipartimento della Difesa, l'astronomia civile lentamente ha incominciato a familiarizzare con il potenziale della tecnologia Hartmann-Shack. L'Università dell'Arizona è situata nel cuore della comunità astronomica americana. L'interesse per la tecnologia di Shack non è riuscita a catturare l'interesse di questo gruppo. Nella prima metà degli anni 1970, Shack distribuì Lenslet ai ricercatori astronomici in visita, ma il valore della sua opera non fu realizzata finché Ray Wilson del European Southern Observatory visitò l'OSC nel 1976. Wilson per primo concepì la correzione tramite l'ottica adattiva nei telescopi nel 1968. Il suo concetto era che se l'entità e le direzioni delle varie aberrazioni potevano essere misurate, allora potevano essere anche attivamente compensate. Quello che mancava a questo punto però era uno strumento di misura del fronte d'onda che potesse eseguire rapidamente l'analisi. Il classico test Hartmann era troppo lento per permettere la correzione attiva. A seguito di una seconda visita in OSC 1977, Wilson tornò con una delle matrici Lenslet originali e la utilizzò per dimostrare una correzione ottica attiva con uno specchio di 1 m. L'altra tecnologia chiave che ha reso il sistema attivo realizzabile è stato l'avvento della camera CCD. Il sensore fronte d'onda Shack-Hartmann fornisce la misura precisa delle aberrazioni e la camera CCD consente l'acquisizione rapida e analisi dei modelli di spot che ne derivano. Anche con il concetto dimostrato, la Tecnica Shack-Hartmann non è stata attuata sui telescopi fino al 1980. Il telescopio di 3,6 metri dell'ESO ha visto la prima luce nel Marzo del 1989 e ha rappresentato il primo telescopio astronomico attivamente corretto. Le immagini elaborate da questo telescopio mostravano dettagli ben al di là delle immagini prive di correzione. A seguito di questa dimostrazione reale, gli astronomi di tutto il mondo sono diventati molto interessati alla tecnica, e mentre diverse

altre tecniche di misurazione dei fronte d'onda sono stati attuati per la correzione di telescopi astronomici, la tecnica di Shack-Hartmann è di gran lunga la più utilizzata.

Il principio di funzionamento è schematizzato in figura. Su una immagine della pupilla è posta una griglia di micro lenti, che forma una corrispondente griglia di immagini della sorgente di riferimento su un rivelatore panoramico.

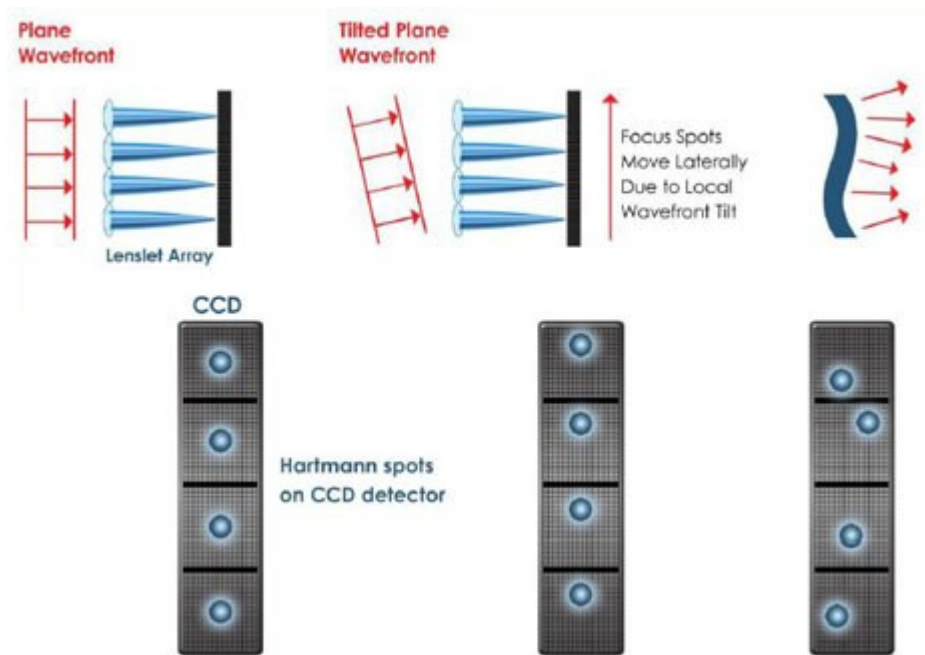


Figura 12 – Esempi tecnologia Shack-Hartmann

Lo spostamento del baricentro della immagine prodotta da ogni lentina, rispetto alla posizione in assenza di perturbazioni, fornisce una stima dell'inclinazione (tilt) media della porzione del fronte d'onda che incide sulla lentina stessa.

La proiezione dell'apertura di ogni lentina sulla pupilla determina le varie zone di sensing le cui dimensioni devono essere $\approx r_0$. Su distanze minori o uguali al parametro di Fried le aberrazioni introdotte dalla turbolenza possono essere considerate trascurabili, per cui considereremo le immagini formate dalle lentine limitate dalla diffrazione. Otteniamo che la dimensione dell'immagine prodotta sul rivelatore è $f_L \cdot \lambda / D_L$ dove f_L e D_L sono, rispettivamente, la focale e il diametro delle lentine.

Consideriamo il caso in cui l'immagine prodotta da ogni lentina cada all'interno di una matrice 2x2 di elementi fotosensibili (quad-cell). Questo è il numero minimo di canali in cui posso suddividere il segnale che permetta di misurare il baricentro dell'immagine. Supponiamo che l'immagine, in assenza di perturbazioni introdotte dalla turbolenza atmosferica, sia centrata rispetto alla quad-cell e indichiamo con S_a, S_b, S_c e S_d i segnali forniti dai quattro elementi sensibili. Se le perturbazioni introducono piccole variazioni ($\Delta x_B, \Delta y_B$) della posizione del baricentro rispetto alle dimensioni dell'immagine, valgono le relazioni:

$$\Delta x_B \approx \frac{1}{2} \left(\frac{\lambda f_L}{D_L} \right) \frac{(S_a + S_d) - (S_b + S_c)}{S_a + S_b + S_c + S_d}$$

$$\Delta y_B \approx \frac{1}{2} \left(\frac{\lambda f_L}{D_L} \right) \frac{(S_a + S_b) - (S_c + S_d)}{S_a + S_b + S_c + S_d} .$$

Le derivate $\partial W / \partial x$ e $\partial W / \partial y$ forniscono gli angoli di inclinazione del fronte d'onda rispetto al piano della lentina, per cui

$$(\Delta x_B, \Delta y_B) = \left(f_L \frac{\partial W}{\partial x}, f_L \frac{\partial W}{\partial y} \right) .$$

sostituendo la relazione precedente nelle equazione, otteniamo una stima per le derivate della funzione di aberrazione nella corrispondente zona di sensing, data da

$$\frac{\partial W}{\partial x} \approx \frac{1}{2} \left(\frac{\lambda}{D_L} \right) \frac{(S_a + S_d) - (S_b + S_c)}{S_a + S_b + S_c + S_d}$$

$$\frac{\partial W}{\partial y} \approx \frac{1}{2} \left(\frac{\lambda}{D_L} \right) \frac{(S_a + S_b) - (S_c + S_d)}{S_a + S_b + S_c + S_d}.$$

Poiché il sensore di Shack-Hartman fornisce una stima delle derivate di W , la funzione di aberrazione può essere ricostruita tramite l'integrazione di un'equazione differenziale di primo grado a meno di un valore costante, che comunque, risulta essere inessenziale ai fini della formazione dell'immagine.

Se le deviazioni della posizione del baricentro sono confrontabili con le dimensioni delle immagini prodotte dalle lentine, le equazioni ottenute non sono più valide. In particolare si perde la relazione lineare tra le derivate di W e i segnali (normalizzati alla somma dei segnali stessi) e si introduce una dipendenza esplicita dalla forma della PSF prodotta da ogni lentina.

2.1.1 ALGORITMO CALCOLO CENTROIDI

Gli algoritmi sviluppati per la detection dei centroidi sono innumerevoli ed a volte utilizzano metodologie ed algoritmi complessi dal punto di vista computazionale. L'algoritmo della **Media Statistica (MS)** o "Centri di Massa" è sicuramente uno degli algoritmi più largamente utilizzati in Ottica Adattiva. Consente una semplice e diretta individuazione della posizione degli spot simmetrici:

$$\begin{cases} \widehat{X}_{MS} = \frac{\sum x * I_{x,y}}{\sum I_{x,y}} \\ \widehat{Y}_{MS} = \frac{\sum y * I_{x,y}}{\sum I_{x,y}} \end{cases}$$

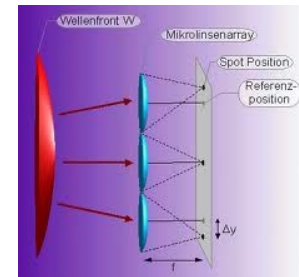


Figura 13 – Esempio Fronte d'onda su sensore SH

Al fine di migliorare le prestazioni dell'algoritmo, sono state studiate diverse varianti:

- **MS Ponderata**

È una variante molto semplice che sfrutta l'idea di attribuire ai pixel con valore associato maggiore un "peso" più alto rispetto a quelli con valore associato minore.

$$\begin{cases} \widehat{X}_{MS} = \frac{\sum x * I^N_{x,y}}{\sum I^N_{x,y}} \\ \widehat{Y}_{MS} = \frac{\sum y * I^N_{x,y}}{\sum I^N_{x,y}} \end{cases}$$

Con N si identifica quindi la potenza con cui viene modificato il "peso" del campione.

Naturalmente maggiore sarà la potenza, maggiormente sarà evidente il picco del massimo dello spot.

- **MS con Soglia Mobile**

A differenza della variante *ponderata* l'introduzione del concetto di soglia sottolinea la particolare sensibilità da parte dell'algoritmo della MS da parte di segnali aggiuntivi allo spot come il rumore di fondo, ed in complesso dell'asimmetria in generale.

La soglia ha proprio lo scopo di eliminare tutti i segnali spuri presenti nell'analisi dello spot.

Per seguire la dinamica dello spot, soprattutto in termini di ampiezza, viene introdotto il concetto di soglia *mobile*.

Non siamo dunque costretti a fissare un livello di soglia fisso, ma il livello si adatta al segnale corrente migliorando la detection.

Di spot in spot la soglia sarà scelta come:

$$I_T = \max(TI_{max}, mN_{oise})$$

Il massimo fra una percentuale T (tipicamente 25% di I_{max}) ed un multiplo intero m (tipicamente 3) del Noise Level.

2.1.2 Pregi e Limiti algoritmo

L'utilizzo del suddetto metodo per la detection dei centroidi è, per sua natura, un algoritmo molto semplice e performante se messo nelle condizioni ottimali di funzionamento.

A livello di implementazione hardware ha il pregio di essere costituito da semplici operazioni di base; somme, moltiplicazioni e divisioni.

Questo consentirà di poter approcciare ad uno sviluppo su FPGA abbastanza agevole utilizzando il minor numero possibile di blocchi aritmetici.

Inoltre il ridotto numero di elaborazioni associate ad ogni singolo pixel consente di utilizzare un'architettura non troppo spinta nelle prestazioni e quindi di poter ottenere buoni compromessi fra area(dell'FPGA) utilizzata e velocità di esecuzione.

Dall'altra parte la semplicità dell'algoritmo ha anche il limite di dover far fronte a possibili malfunzionamenti dovuti ad anomalie nella forma degli spot analizzati. Infatti l'algoritmo dei centri di massa risente molto di elementi destabilizzanti come il rumore di fondo o in generale delle asimmetrie presenti nello spot analizzato.

2.2 SVILUPPO ALGORITMO IN MATLAB

Prima di iniziare l'implementazione dell'algoritmo in VHDL, ho cercato di validare la funzionalità dell'algoritmo sviluppandolo in ambiente matlab.

Vedi APPENDICE A.

Lo script, come il progetto finale, è stato diviso in due fasi:

- INIZIALIZZAZIONE
- ANALISI A REGIME

2.2.1 Inizializzazione

Consiste nell'individuazione del corretto posizionamento della griglia di riferimento rispetto agli spot analizzati dal sensore

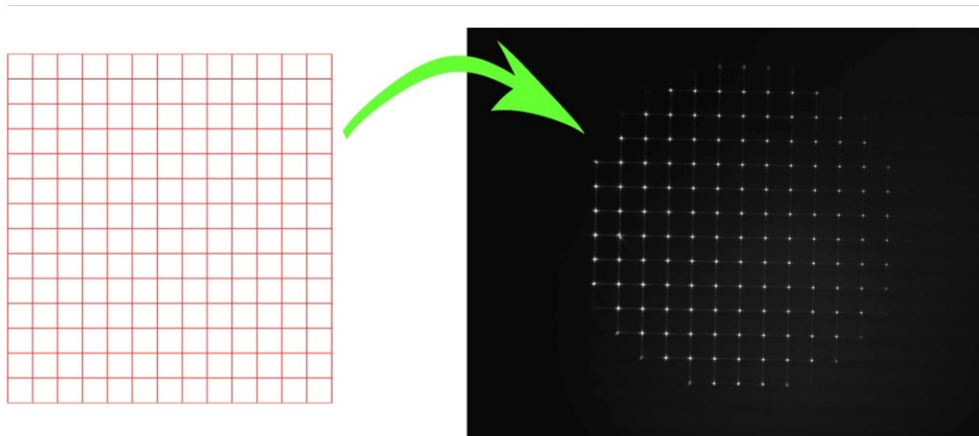


Figura 14 – Posizionamento Griglia Virtuale

2.2.2 Analisi a Regime

Consiste nella detection dei centroidi vera e propria, ha luogo solo dopo il posizionamento della griglia di riferimento e quindi dopo la fase di inizializzazione.

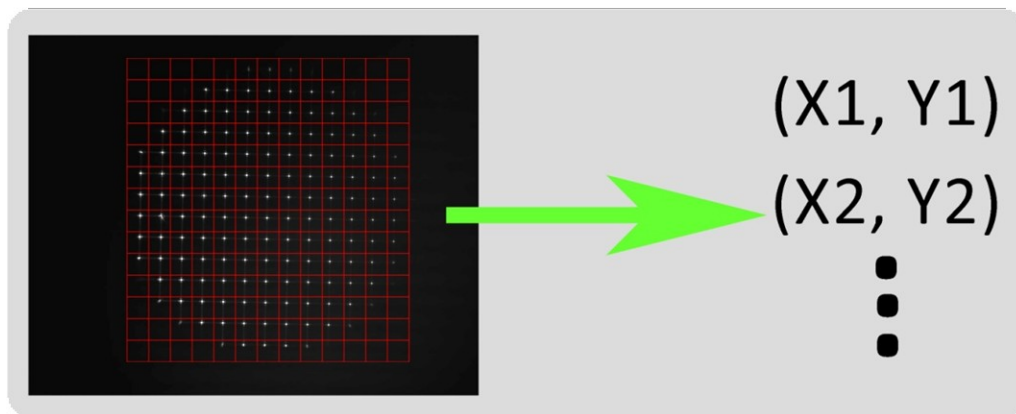


Figura 15 – Analisi a Regime

3 ANALISI PRELIMINARI

Il problema attuale nell'utilizzo di sensori fronte d'onda Shack Hartmann, è il continuo aumento tecnologico legato alle prestazioni e alla risoluzione.

Se andiamo a considerare i componenti costitutivi di un sensore Shack Hartmann questo si traduce in:

- lenslet sempre più piccoli e con un numero di aperture sempre maggiore
- camere con risoluzioni sempre maggiori dall'elevato frame rate

Questo porta a dover far fronte ad un aumento esponenziale della mole di pixel da elaborare da parte del calcolatore a cui il sensore è collegato.

Inevitabilmente questo porta ad una limitazione delle prestazioni globali del sistema. Pur avendo sensori ad alta risoluzione e velocità, ci si ritrova a non poterne sfruttare a pieno le potenzialità a causa di una limitata potenza di calcolo.

3.1 PUNTI DI CRITICITA'

Per sviluppare un'efficiente interfaccia hardware si deve innanzitutto avere un'idea di quelli che possono essere i vincoli progettuali più restrittivi.

Sulla base di questi risultati verranno fatte le considerazioni e le scelte necessarie ad un corretto sviluppo del progetto.

3.1.1 VELOCITA' DI ELABORAZIONE

L'algoritmo scelto risulta subito di facile implementazione. Il punto chiave sul suo utilizzo si pone nel come impostare la struttura di calcolo. Le operazioni più "delicate" risultano i prodotti parziali, accumulati dalle diverse sommatorie indicate nella formula:

$$\begin{cases} \widehat{X}_{MS} = \frac{\sum x * I_{x,y}}{\sum I_{x,y}} \\ \widehat{Y}_{MS} = \frac{\sum y * I_{x,y}}{\sum I_{x,y}} \end{cases}$$

Analizzare un numero elevato di frame al secondo si deve tradurre inevitabilmente, nell'analisi, pixel dopo pixel, dell'intera matrice dati provenienti dal relativo sensore Shack – Hartmann. Naturalmente all'aumentare della risoluzione del sensore e del frame rate della camera, aumenta anche il pixel rate da analizzare.

Questo porterà a dover sviluppare blocchi aritmetici molto veloci!

3.1.2 FLESSIBILITA'

Concetto molto importante e necessario a rendere il progetto utilizzabile da più sensori Shack-Hartmann.

Non prendere in considerazione un elemento così importante vorrebbe dire creare un dispositivo utilizzabile limitatamente, soluzione poco proponibile anche da un punto di vista prettamente commerciale.

4 SCELTE PROGETTUALI

Alla luce delle considerazioni fatte sull'algoritmo da implementare, su i suoi pregi e difetti sono state fatte delle scelte sulle specifiche da utilizzare come base per lo sviluppo del progetto:

4.1 Camera Link

Camera Link è un'interfaccia di comunicazione per applicazioni di visione.

L'interfaccia estende la tecnologia di base Link Channel per fornire un'applicazione più utile per la visione.

Per anni, il mercato scientifico e industriale del video digitale ha difettato di un metodo standard di comunicazione. I produttori di frame grabber e fotocamere sviluppate con connettori diversi, rendendo la produzione difficile e di conseguenza un utilizzo molto confuso per i consumatori.

Uno standard di connettività tra le fotocamere digitali e frame grabber sarà di gran lunga utile e diventerà ancor più necessario in quanto la mole di dati continua a aumentare.

Il crescere di sempre più telecamere diverse, segnali e trasmissione dei dati hanno fatto della connettività standard Camera Link una necessità.

L'interfaccia Camera Link semplificherà la trasmissione, così come il costo di tale trasmissione.

Il cavo standard sarà in grado, inoltre, di gestire la maggiore velocità dei segnali e l'assemblaggio del cavo consentirà ai clienti di ridurre i costi mediante tariffazione di volume.

4.1.1 LVDS Descrizione Tecnica

Low Voltage Differential Signaling (LVDS) è uno standard di interfaccia ad alta velocità e basso consumo.

La norma, conosciuta come ANSI/TIA/EIA-644, è stata approvata nel marzo 1996.

LVDS utilizza segnali differenziali, con uno swing differenziale nominale di 350 mV. Il basso

diminuisce swing segnale di salita e discesa volte per raggiungere un massimo teorico velocità di trasmissione di 1,923 Gbps in un mezzo loss-less. Il basso segnale di swing significa anche che la norma non dipende da un'alimentazione particolare di tensione. LVDS utilizza i driver current-mode, che limitano il consumo di energia.

I segnali differenziali sono immuni ai ± 1 volt V comune di rumore.

4.1.2 Channel Link

National Semiconductor ha sviluppato la tecnologia Link Channel come soluzione per la visualizzazione a schermo piatto, sulla base di segnali LVDS. La tecnologia è stata poi estesa in un metodo per la trasmissione generale dei dati. Link Channel è costituito da una coppia di driver e ricevitore. Il driver accetta 28 segnali single-ended di dati e un clock single-ended. I dati vengono serializzati 7:1, e i quattro flussi di dati e del clock dedicato sono

trasmessi attraverso cinque coppie LVDS. Il ricevitore accetta i quattro dati LVDS e di clock, e quindi guida il 28 bit ed il clock in uscita.

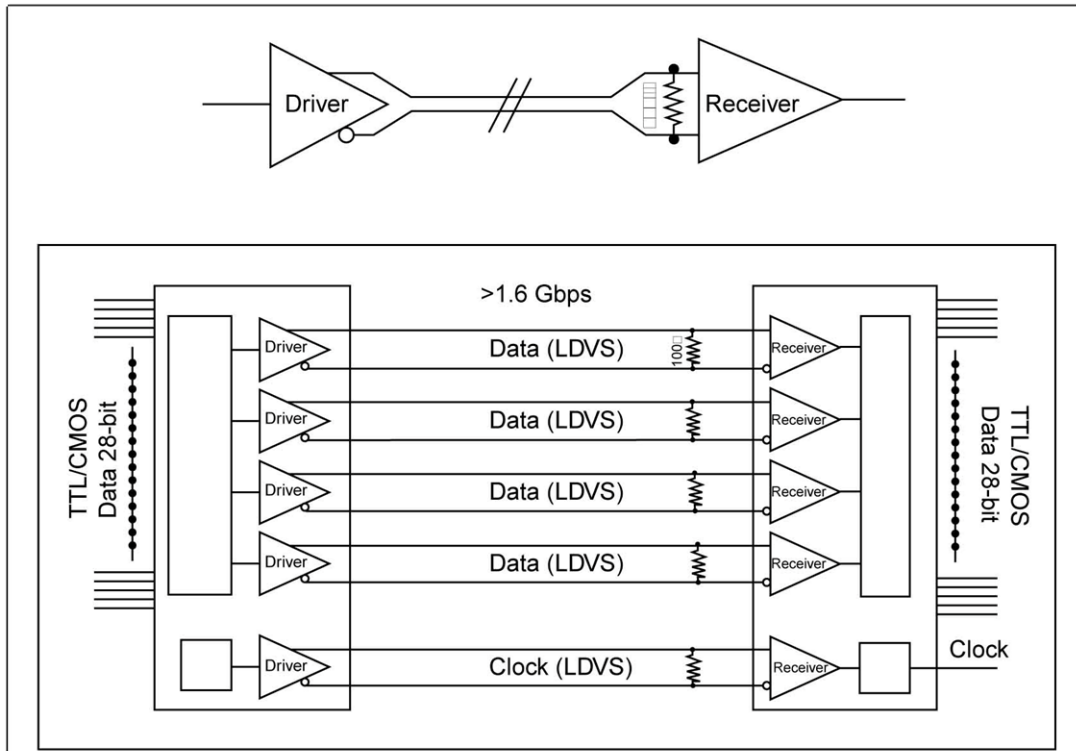


Figura 16 – Collegamento LVDS Camera Link

4.1.3 Connettore e Cavo di Dimensioni Ridotte

Il metodo di trasmissione dati Camera Link richiede pochi conduttori per il trasferimento dei dati. Cinque paia di fili possono trasmettere più di 28 bits di dato. Questi fili riducono le dimensioni del connettore, seguendo la riduzione delle dimensioni delle telecamere in commercio.

4.1.4 Alta Velocità di Trasmissione Dati

La velocità di trasmissione dati del Camera Link chip (più di 2.38 Gbits/s) supportano il crescere delle esigenze di elevate prestazioni.

4.1.5 Principio di Funzionamento

Lo standard Camera Link presuppone la possibilità, in base alle differenti camere, di settare a piacere i vari parametri di funzionamento quali, risoluzione, velocità di trasmissione, tempo di integrazione, finestra di interesse e molti altri parametri dipendenti molto dalla tipologia di camera.

In base alla scelta di questi parametri, variano le modalità e le frequenze in uscita dei seguenti segnali:

- **Strobe** : Segnale di sincronismo
- **FVAL** : Frame Valid (FVAL) è definito ALTO per linee valide
- **LVAL** : Line Valid (LVAL) è definito alto per pixels validi
- **DVAL** : Data Valid (DVAL) è definito alto per dati validi

Fondamentalmente possono essere due le tipologie di funzionamento di una Camera:

High Speed e Low Speed.

In figura18 sono illustrate queste due tipologie di funzionamento.

Nel primo caso (High Speed) ci troviamo nella condizione in cui la camera funziona al limite delle sue prestazioni possibile.

Al fine di massimizzare il flusso dati, il segnale DVAL (data valid) viene mantenuto sempre alto.

Questo significa che l'utente ad ogni fronte di salita del segnale di sincronismo (strobe) dovrà campionare i dati in ingresso. In questa situazione saranno solo i segnali di FVAL (frame valid) e LVAL (line valid) a determinare la validità dei dati.

Nel secondo caso (Low Speed) le condizioni di funzionamento del sensore sono meno vincolanti.

DVAL1 viene quindi utilizzato per "rallentare" il flusso di dati in uscita. Quando il frame e la linea sono validi sarà DVAL1 a discriminare la validità del dato, al fronte di salita del segnale di strobe.

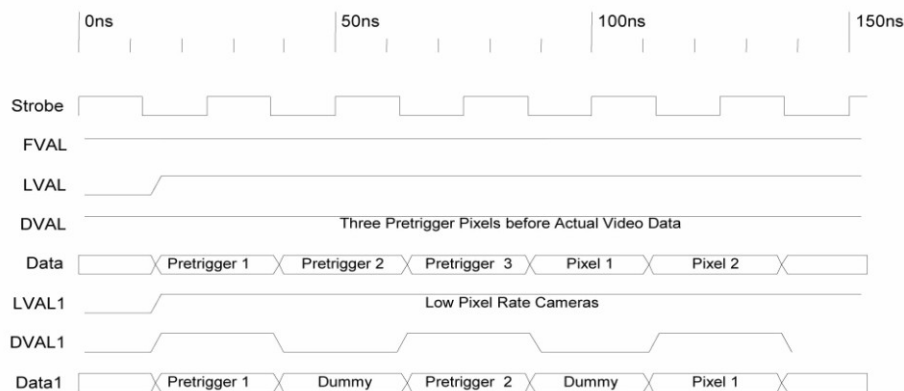


Figura 17 – Segnali Camera Link

4.2 FPGA

La FPGA sta diventando uno dei più importanti strumenti per circuiti digitali e la chiave del suo successo risiede sia nella sua programmabilità, che consente di realizzare immediatamente qualsiasi funzione, sia nei bassi costi di progettazione, in quanto non è richiesta la realizzazione di uno specifico design come le Standard-Cell. Questa maggior flessibilità in fase di progettazione si paga in termini di area non utilizzata perché le FPGA occupano un area almeno dieci volte più grande. Inoltre, questi dispositivi programmabili risultano almeno tre volte più lenti delle Standard-Cell e; la bassa velocità con cui viaggia il segnale è dovuta principalmente al fatto che i diversi blocchi logici all'interno di una FPGA sono connessi attraverso dei blocchi programmabili che presentano al loro interno elementi (componenti) attivi il cui attraversamento introduce un grosso termine di ritardo, mentre nelle Standard-Cell il segnale attraversa solo piste metalliche e via, che non presentano gli elevati valori di resistenza e capacità dei pass-transistor presenti negli switch module.

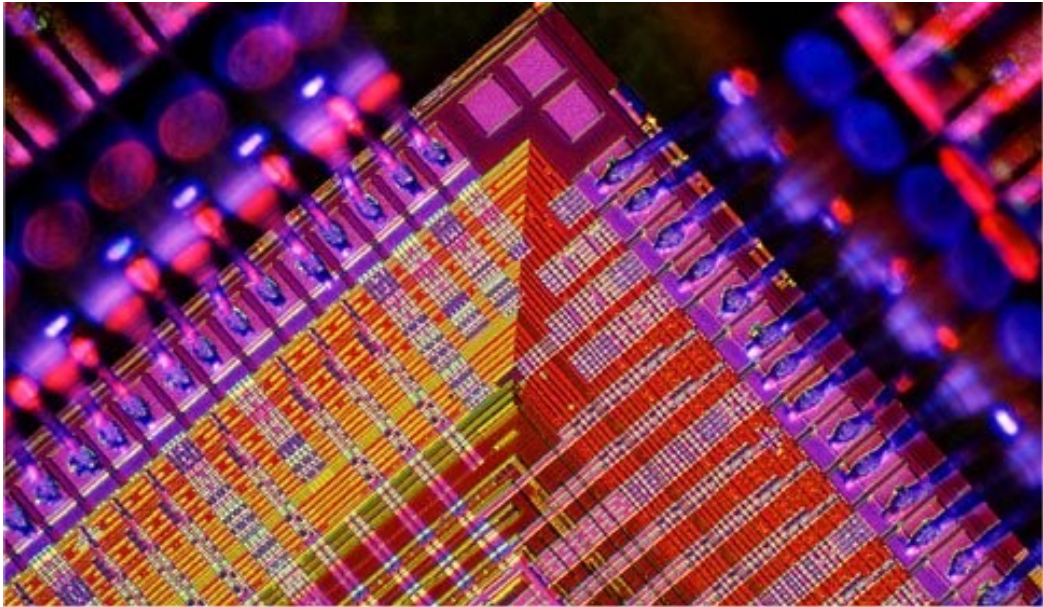


Figura 18 – Struttura FPGA

Questo capitolo ha il compito di presentare l'ambiente in cui si articola lo studio condotto per lo svolgimento di questa tesi e si prefigge l'obiettivo di analizzare i componenti circuitali utilizzati per lo sviluppo di tale lavoro.

E' necessario inquadrare il problema da affrontare partendo da una descrizione dell'architettura della FPGA, argomento cardine attorno al quale ruotano tutte le simulazioni effettuate mediante l'impiego degli appositi tool. Si esegue, inoltre, una breve trattazione dei blocchi e delle strutture che costituiscono l'architettura della FPGA al fine di delineare i possibili problemi e le peculiarità che la caratterizzano.

4.2.1 Considerazioni generali di una FPGA

La FPGA è un tipo di circuito VLSI e consiste in una matrice di blocchi logici (CLB), in cui vengono implementate delle funzioni logiche, e che sono interconnessi tra di loro mediante risorse programmabili; quest'ultimi sono responsabili di un grande consumo in termini di area a causa degli elementi attivi che li compongono.

In figura è mostrato un esempio di architettura di FPGA: i blocchi funzionali (indicati con L) sono separati tra loro mediante canali verticali e orizzontali, all'interno dei quali corrono parallele le piste di interconnessione definite segment wire; nei punti di intersezione tra un canale verticale e uno orizzontale sono posizionati gli switch module, i quali contengono della logica programmabile per poter realizzare i collegamenti necessari.

La connessione tra due diversi CLB avviene mediante l'uso di elementi attivi che possono essere programmati, ossia posti ad ON (il gate del transistor MOS è posto a Vdd) o ad OFF (il gate è posto a ground), dove tali valori vengono memorizzati nelle celle di memoria di una RAM.

Pertanto, se il pass-transistor è acceso, consente il passaggio bidirezionale del segnale, viceversa, se è spento, non permette il suo attraversamento. La presenza dei

pass-transistor introduce resistenze e capacità parassite lungo il cammino percorso dal segnale.

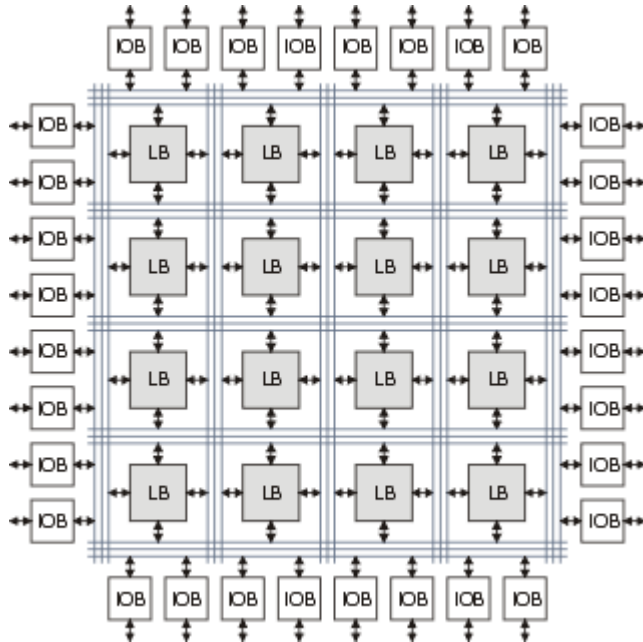


Figura 19 – Struttura FPGA

4.2.2 Blocchi e parametri caratteristici

Una FPGA è costituita da blocchi logici, blocchi Input/Output e interconnessioni programmabili. Per implementare un circuito ciascuno dei blocchi logici viene opportunamente programmato, realizzando una particolare funzionalità del circuito, e ognuno dei pad viene fissato in input o in output.

Di seguito si analizzano i singoli blocchi che realizzano una FPGA e si forniscono le definizioni principali, ciò è necessario per delineare le caratteristiche di base dell'architettura in esame e rendere maggiormente comprensibili i ragionamenti e le considerazioni fatte.

4.2.3 Cluster Logic Block

Ciascun blocco logico (CLB) consiste in un gruppo di elementi logici di base (BLE) collegati tra loro mediante interconnessioni locali (local routing). Un esempio esplicativo di quanto appena descritto è riportato in figura.

Ciascun BLE è un'unità indivisibile che contiene al suo interno elementi logici combinatori e sequenziali.

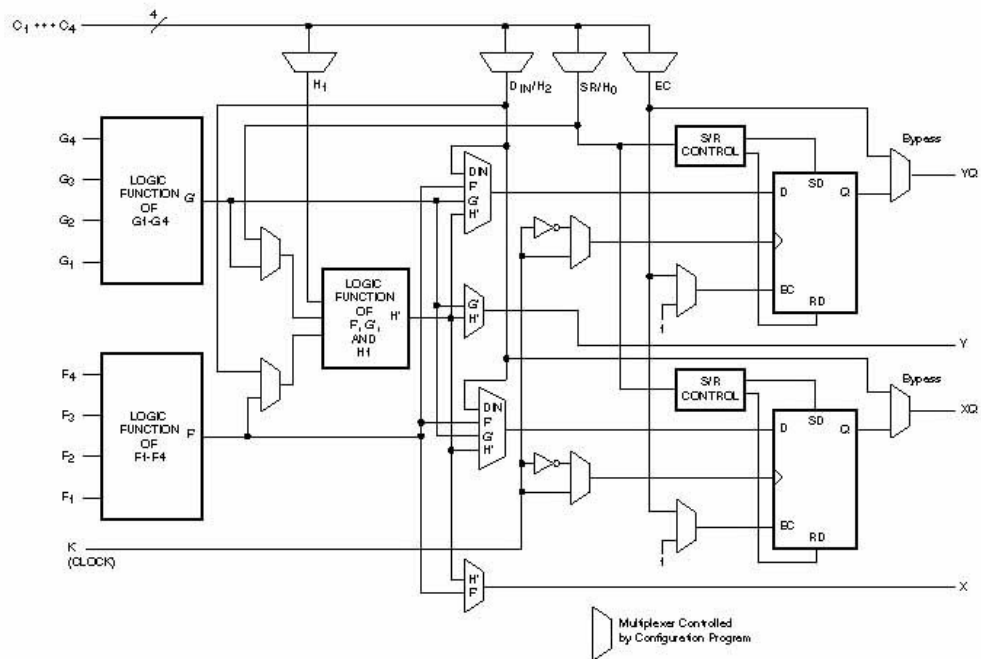


Figura 20 –Struttura CLB

Un generico CLB è descritto dai seguenti 4 parametri:

1. k numero di ingressi di una LUT;
2. N numero di BLE presenti all'interno di un CLB e numero di uscite dell'intero blocco logico;
3. I numero di ingressi al CLB;
4. Mclk numero di ingressi al CLB dedicati ai segnali di clock.

Una LUT con più ingressi può implementare più logica al suo interno, perciò il numero di CLB da utilizzare in un circuito diminuirebbe e aumenterebbe l'area da poter dedicare al routing. Tuttavia, visto che la complessità di una LUT cresce esponenzialmente all'aumentare del numero di ingressi, risulta poco gestibile l'utilizzo di una LUT come un unico blocco logico avente un elevato numero di ingressi. Così, invece di realizzare un blocco con molti pin, risulta più semplice raggruppare insieme alcune LUT ed interconnetterle con un routing locale. Questo tipo di interconnessione è più veloce rispetto a quello di tipo generale tra blocchi logici, e migliora le caratteristiche di velocità della FPGA, ossia i tempi di ritardo sono ridotti. Tuttavia, il routing locale effettuato tra le diverse LUT aumenta quadraticamente al crescere delle dimensioni delle stesse LUT, quindi non è conveniente realizzare CLB molto grandi altrimenti l'area necessaria per le interconnessioni locali supererebbe quella risparmiata grazie al fatto che non bisogna più effettuare il routing generale.

In figura si può osservare il routing locale all'interno di un CLB caratterizzato da N BLE.

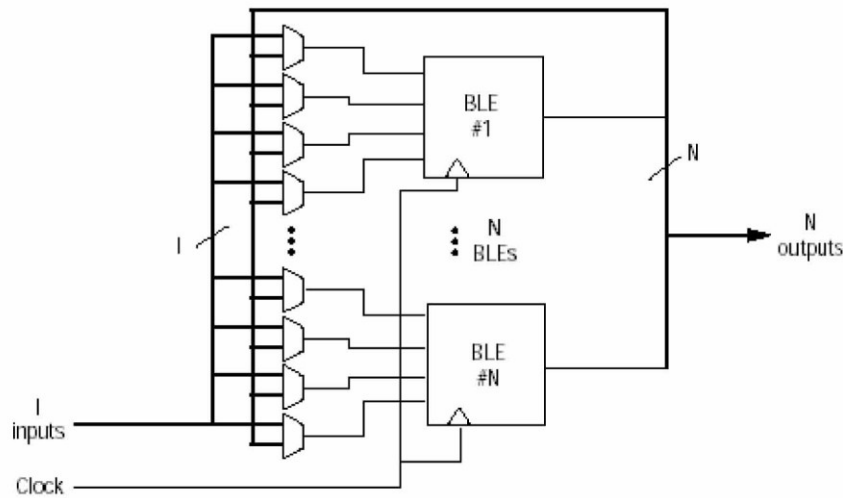


Figura 21 – Connessione tra BLE

Lungo il perimetro di ciascun blocco logico sono distribuiti i pin di ingresso e di uscita, e questi sono collegati ai segmenti wire di un canale adiacente mediante dei transistor, elementi attivi programmabili che realizzano l'interconnessione. Un esempio di quanto appena detto è mostrato in figura.

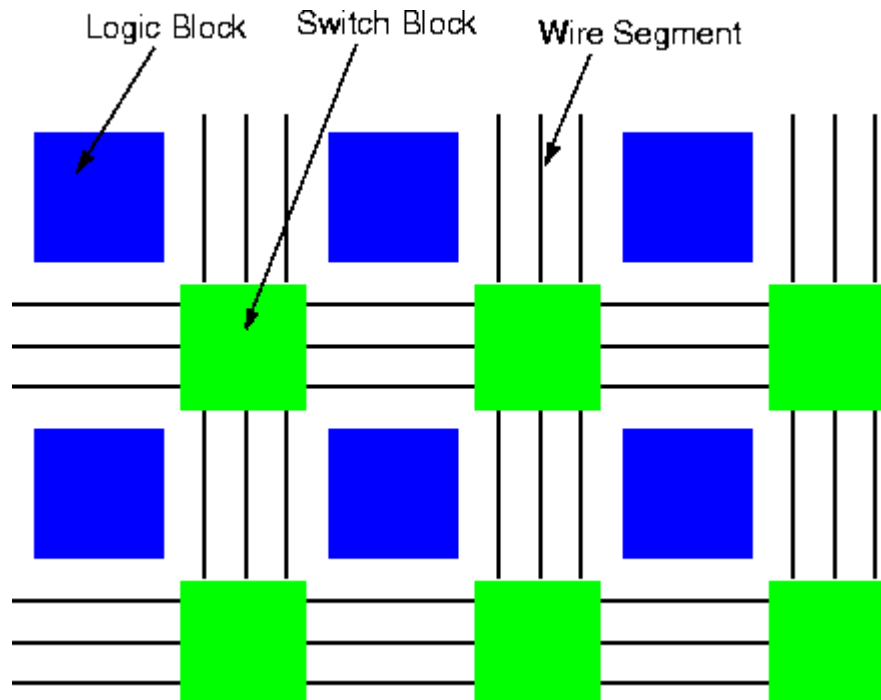


Figura 22 – Matrice interconnessioni

4.2.4 Connection module

Le connessioni tra i diversi blocchi logici si realizzano grazie a due tipi di componenti: i segment wire presenti lungo i canali orizzontali e verticali che corrono adiacenti ai blocchi logici, e gli elementi programmabili (sostanzialmente dei MOS) che realizzano i collegamenti tra i vari wire.

Il connection module, si utilizza per connettere i pin dei blocchi logici ai diversi segment wire che appartengono al canale.

Si definisce flessibilità F_c di un connection module il numero di piste, appartenenti ad un canale, ai quali può connettersi ciascun pin dei vari blocchi logici. Il caso mostrato in figura presenta un valore pari a $F_c=2$. Utilizzando un piccolo valore di F_c si riduce il numero di transistori MOS necessari per collegare i vari pin alle tracce del canale, tuttavia ciò provoca solo un'apparente riduzione di area, in quanto il processo di routing risulta più complicato e richiede una capacità di canale (W) più grande. Questo implica un aumento dell'area occupata per far spazio alle nuove piste necessarie. Pertanto si richiede un valore

appropriato per F_c , e, generalmente, lo si fissa a W/N ; questa scelta garantisce che tutte le piste all'interno di ciascun canale siano pilotate almeno da un pin di uscita di ogni blocco logico.

4.2.5 Switch module

Uno switch module è un blocco di forma quadrata avente un numero di terminali su ciascun lato pari a W . Si posiziona nel punto di intersezione tra un canale verticale e uno orizzontale e, programmando i pass-transistor al suo interno, permette il collegamento tra una pista appartenente ad un canale e quella di un altro. Per semplicità si può pensare che le interconnessioni viaggiano su due livelli di metal: sul primo livello tutte le piste appartenenti al canale orizzontale, sul secondo quelle appartenenti al canale verticale.

Si definisce flessibilità F_s di uno switch module il numero di connessioni programmabili esistenti tra ciascun terminale e gli altri, questo valore è la chiave di volta dell'intera architettura perché condiziona la fase di routing.

La figura mostra un'architettura di FPGA di tipo Island-Style dove i blocchi logici sono disposti come gli elementi di una matrice bidirezionale e sono separati gli uni dagli altri da canali verticali e orizzontali nei cui punti di intersezione si trovano gli switch module che realizzano le connessioni tra le diverse tracce poste all'interno dei vari canali adiacenti ai CLB.

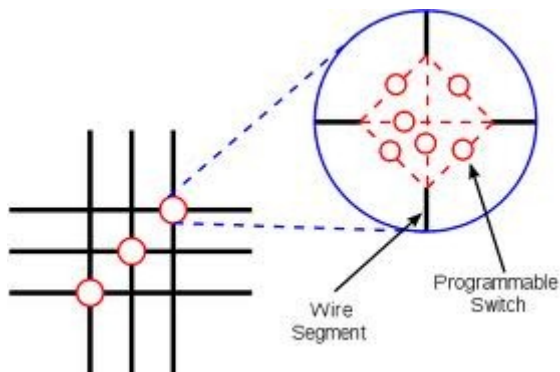


Figura 23 - Island-Style

Uno switch module è costituito da elementi attivi, il cui ingombro è abbastanza rilevante, quindi si tende a limitare il loro numero all'interno di ciascun blocco di connessione; è necessario, però, un giusto compromesso tra area occupata e capacità di effettuare un routing comodo ed efficiente mediante un numero considerevole di transistor che garantiscano una flessibilità adeguata. Inoltre, per poter programmare lo switch module è necessario che vi sia qualche elemento di memoria che tenga conto se la connessione rispetto ad una delle possibili direzioni permesse risulta ON o OFF e questo aumenta notevolmente l'occupazione di area.

Infine, va tenuto in conto che questo blocco ha un effetto preponderante sulla velocità dei collegamenti tra i vari CLB, dunque risulta l'elemento architetturale più critico all'interno della FPGA. Pertanto, in questi ultimi anni l'attenzione si è focalizzata sullo sviluppo di architetture in cui il processo di routing, che gestisce questi collegamenti programmabili, risulti il più efficiente possibile.

4.2.6 Interconnessioni

Le interconnessioni all'interno di una FPGA sono costituite da piste metalliche, che viaggiano all'interno di canali orizzontali e verticali e che possono avere una qualsiasi lunghezza, dove per lunghezza di ciascun segmento si intende il numero di blocchi logici attraversati. Il numero di segment wire all'interno di ciascun canale si indica con W .

Un'architettura si definisce "segmented" se possiede segment wire aventi diversa lunghezza, anche maggiore di 1, ossia che percorrono all'interno dei canali dei tratti più lunghi di un singolo CLB. In figura 25 è riportato un esempio in cui vi sono segmenti di lunghezza 1, 2, 3 e 4. Pertanto i wire più lunghi di un CLB passeranno attraverso alcuni switch module prima di giungere a destinazione nel blocco logico sink.

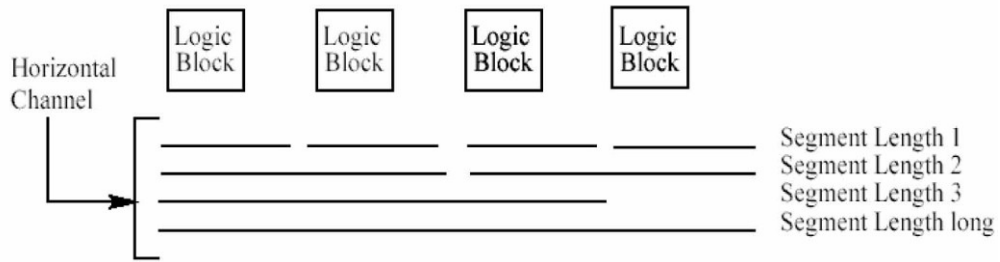


Figura 24

I tool che effettuano il routing cercano di trovare i cammini più brevi possibili e, in primo luogo, suddividono le reti di interconnessione tra i vari CLB in collegamenti punto-punto (source-sink), andando alla ricerca del percorso minimo per ciascuna connessione. E' importante sfruttare in maniera oculata le risorse a disposizione, distribuendo le interconnessioni lungo i canali in modo da uniformare l'occupazione delle tracce.

Il programma che effettua il processo di routing associa una funzione di costo a ciascuna interconnessione basando la sua scelta sulla decisione di ottimizzare o l'area occupata dai collegamenti o la velocità dei segnali che viaggiano attraverso questi. In figura è riportato un esempio di realizzazione di una struttura tra blocchi logici.

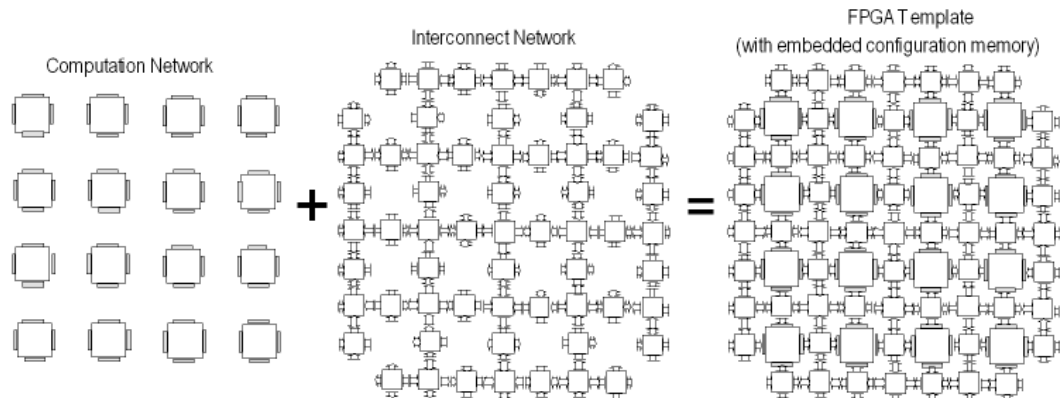


Figura 25 – Struttura a blocchi

Esistono molte tecniche di stima delle possibili interconnessioni da realizzare, sostanzialmente esse valutano il numero ottimale di tracce da utilizzare per effettuare i collegamenti più brevi possibili; tuttavia, si richiede un giusto compromesso tra l'area necessaria alla creazione dei canali e la lunghezza dei collegamenti tra il blocchi logici, in quanto canali più larghi consentono cammini più brevi e, viceversa, canali più stretti implicano interconnessioni più lunghe.

Interconnessioni lunghe presentano ritardi elevati in quanto devono essere attraversati molti switch module, ossia il segnale deve passare attraverso dei pass-

transistor che presentano una resistenza e due capacità parassite. Il ritardo di propagazione viene analizzato mediante tecniche che modellizzano il percorso effettuato dal segnale mediante reti RC.

Il nostro studio si concentra sull'analisi dei cammini critici, in cui il fattore dominante è il numero di switch module attraversati piuttosto che il fan-out associato alla rete.

5 SVILUPPO DEL PROGETTO

Introduzione

Il lavoro svolto in fase di analisi ha messo alla luce la principale difficoltà nella realizzazione di questo progetto, ossia la necessità di creare un “core” dalle alte prestazioni, capace di eseguire ed accumulare ciclicamente, i dati parziali elaborati provenienti dal sensore. Una volta salvati questi dati parziali sarà compito degli altri blocchi dedicati, realizzare le operazioni mancanti. A livello Hardware questo si tradurrà in una struttura a “pipeline”, flessibile e performante, capace di alleggerire i vincoli imposti dalla velocità dell’algoritmo di detection e quindi l’occupazione dell’area utilizzata all’interno dell’FPGA.

IL tutto sarà sviluppato, analizzato e simulato in ambiente ISE 12.3 di Xilinx.

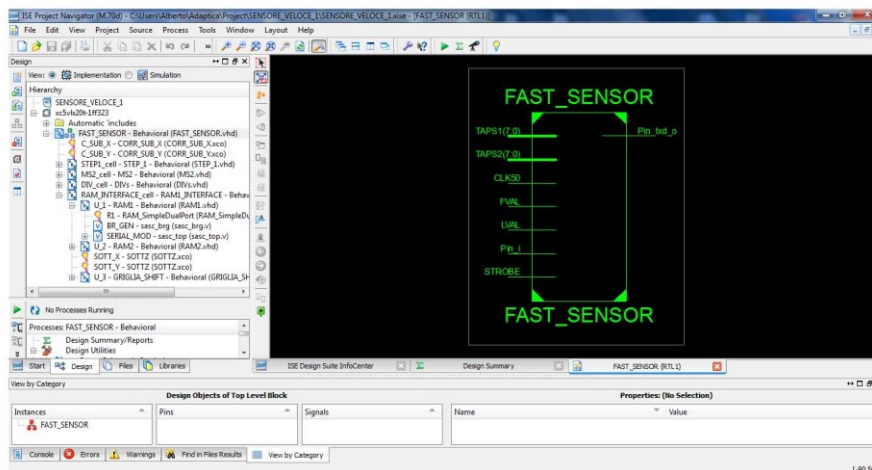


Figura 26 – Top Progetto

Prima di sviluppare in dettaglio la descrizione di ogni blocco, è doveroso soffermarsi sul funzionamento del sistema distinguendo il suo funzionamento in due fasi distinte.

Fase 1: INIZIALIZZAZIONE

Il suo ruolo è l’individuazione e l’assestamento dell’esatto posizionamento della griglia virtuale di riferimento necessaria per il corretto svolgimento dell’algoritmo della media statistica.

Esegue una funzione molto semplice ma allo stesso tempo fondamentale per il corretto funzionamento dell’intero sistema.

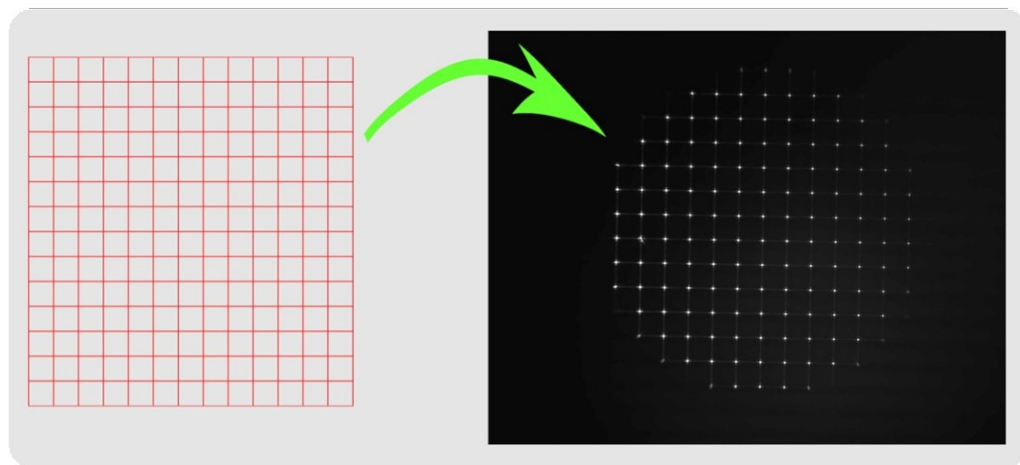


Figura 27 - Inizializzazione

Si conclude dopo l'analisi di pochi frame e se non ci sono interventi esterni che ne obbligano un nuovo ciclo, non verrà più invocato.

Fase 2: SISTEMA A REGIME

Posizionata la griglia virtuale, gli spazi di analisi da essa individuati saranno fissati e sarà quindi possibile eseguire ciclicamente un'analisi real time su ogni spot individuato.

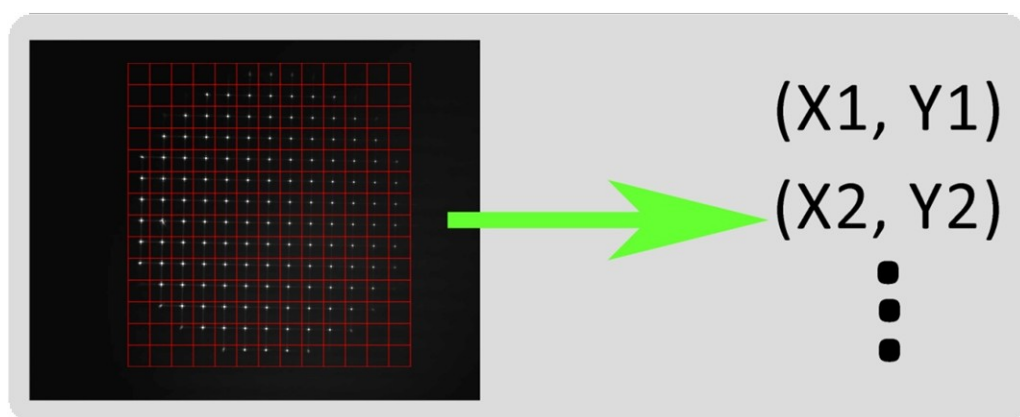


Figura 28 – Funzionamento a Regime

Ognuna di queste fasi ha blocchi logici (*entity*) dedicati ed in alcuni casi condivisi.

Nei paragrafi seguenti si andrà ad illustrare tutti i blocchi facenti parte del progetto, illustrando e spiegando nel dettaglio, le loro particolarità.

5.1 FASE 2

L'architettura dedicata all'analisi degli spot ed alla loro individuazione si può dividere essenzialmente in tre blocchi come illustrato in evidenza nella raffigurazione dell'intero sistema.

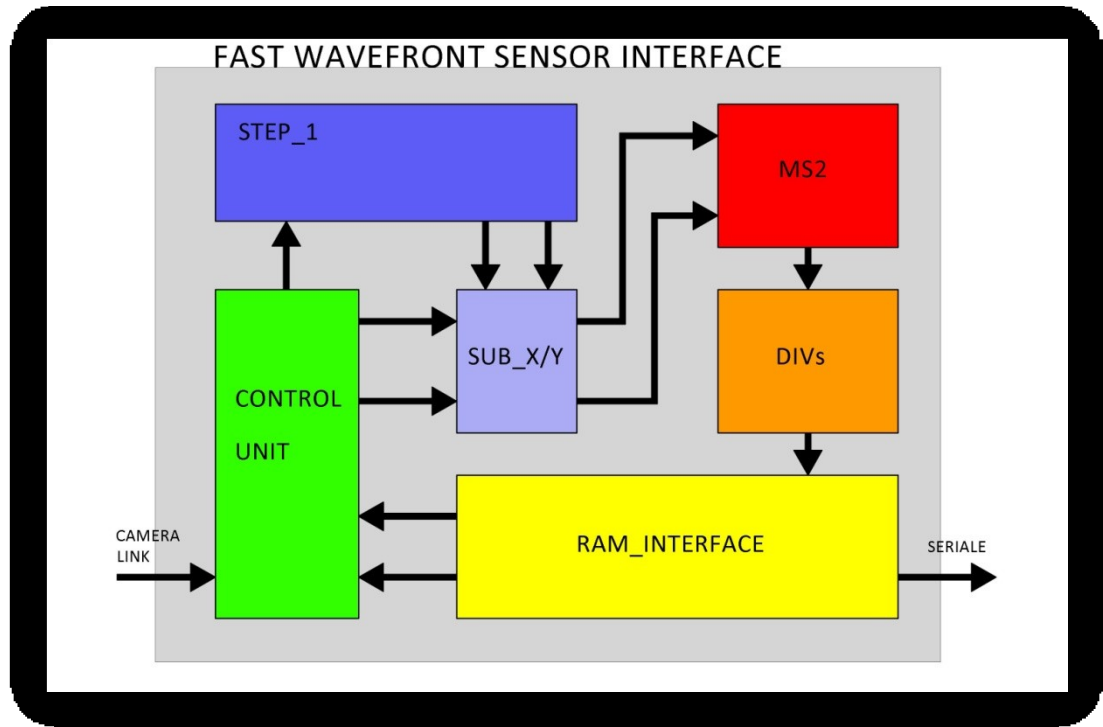


Figura 29 – Schema interno del Top

Il loro funzionamento spiegato in dettaglio nei paragrafi seguenti, effettua mediante una pipeline i passi necessari ad estrarre i centroidi attraverso l'implementazione dell'algoritmo della media statistica.

$$\begin{cases} Spot_x = \frac{\sum_{i=1}^M \sum_{j=1}^N X_i I_{Threshold}(X_i, Y_j)}{\sum_{i=1}^M \sum_{j=1}^N I_{Threshold}(X_i, Y_j)} \\ Spot_y = \frac{\sum_{i=1}^M \sum_{j=1}^N Y_i I_{Threshold}(X_i, Y_j)}{\sum_{i=1}^M \sum_{j=1}^N I_{Threshold}(X_i, Y_j)} \end{cases}$$

con

$$Threshold = Max[(I(X_i, Y_j) * 25\%), 3NL]$$

e quindi una

$$I_{Threshold}(X_i, Y_j) = (I(X_i, Y_j) > I_{Threshold}(X_i, Y_j)) ? I(X_i, Y_j) : 0$$

Algoritmo che a livello di implementazione su FPGA deve necessariamente essere diviso in più parti.

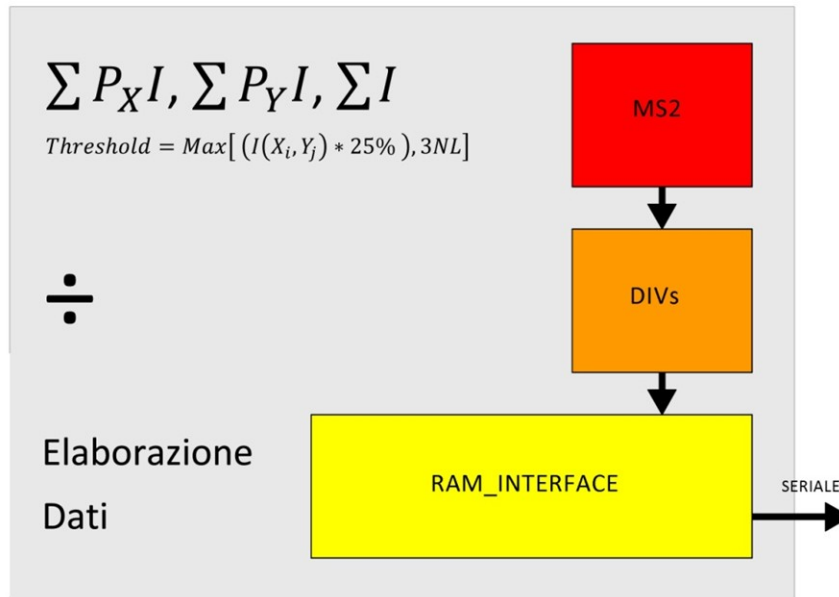


Figura 30 – Suddivisione implementazione Algoritmo

Le tempistiche e quindi i vincoli legati alla velocità impongono che le diverse fasi abbiano architetture appositamente disegnate, andiamo a descriverle in dettaglio.

5.1.1 MS2

Costituisce sicuramente il blocco più performante dell'intero progetto. Al suo interno è sviluppata un'architettura di logica puramente combinatoria, ossia formata da elementi aritmetici con tempi di latenza "nulli". Questa caratteristica garantisce un'elaborazione dei pixel in ingresso priva di qualunque dipendenza dal clock di sistema e quindi la flessibilità di adattarsi al segnale di sincronismo di qualsiasi camera utilizzata.

In questo modo sarà lo stesso segnale di sincronismo descritto dallo standard camera link (segnale di strobe) a fungere da sincronismo per il sistema di acquisizione ed elaborazione dati.

All'interno dell'entity MS2 è inoltre presente il blocco RAM_Max il quale ha il compito di memorizzare il massimo di ogni cella e di restituire, al frame successivo, una percentuale di quel massimo. Grazie a questa entity è possibile realizzare ed implementare sistematicamente il meccanismo della "soglia mobile".

5.1.1.1 ENTITY

Andiamo ora a vedere quelli che sono i segnali di ingresso e uscita:

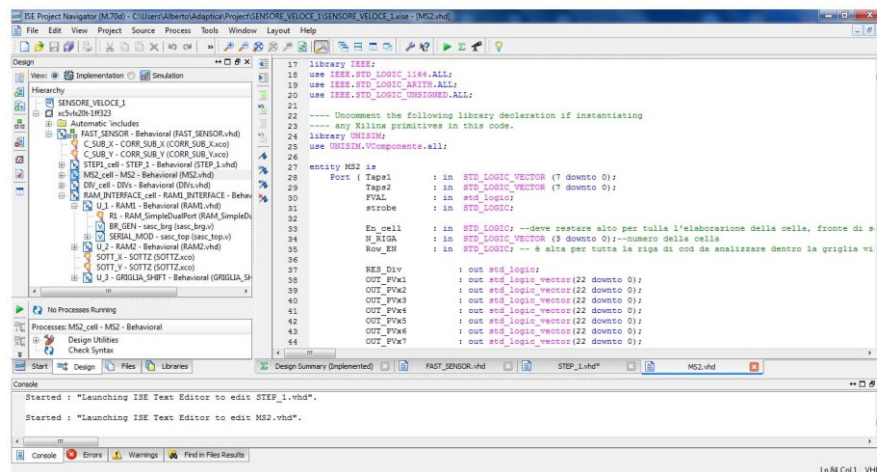


Figura 31 – Immagine Xilinx Entity MS2

- **INGRESSI**
- Taps1 : Dato ad 8 bit proveniente dalla camera del sensore (vedi Camera Link)
- Taps2 : Dato ad 8 bit proveniente dalla camera del sensore (vedi Camera Link)
- FVAL: Frame valid (vedi Camera Link)

- Strobe : segnale di sincronismo (vedi Camera Link)
 - En_cell : segnale che resta alto fino alla fine dell'analisi di ogni riga di celle
 - N_RIGA : indice numero riga di celle
 - Row_EN : segnale che resta alto per tutta la durata dei dati utili da analizzare
- **USCITE**
 - RES_Div : segnale di reset per il blocco DIVs
 - OUT_PVxn (n= 1:13): risultati ottenuti a fine elaborazione celle coordinata X
 - OUT_PVyn (n= 1:13): risultati ottenuti a fine elaborazione celle coordinata Y
 - OUT_Vn (n= 1:13): risultati ottenuti a fine elaborazione
 - Done: segnale di fine elaborazione avvenuta

Come anticipato in fase introduttiva, i vari blocchi logici lavorano in una pipeline, permettendo al sistema di lavorare a velocità diverse.

MS2 esegue una prima importante fase di pre-analisi che consiste nella realizzazione delle sommatorie dell' algoritmo implementato.

$$\sum P_X I, \sum P_Y I, \sum I$$


$$Threshold = \text{Max}[(I(X_i, Y_j) * 25\%), 3NL]$$


Figura 32 – Descrizione funzionalità Blocco MS2

5.1.1.2 Struttura interna

Andiamo a descrivere in dettaglio quelli che sono i blocchi fondamentali presenti in MS2.

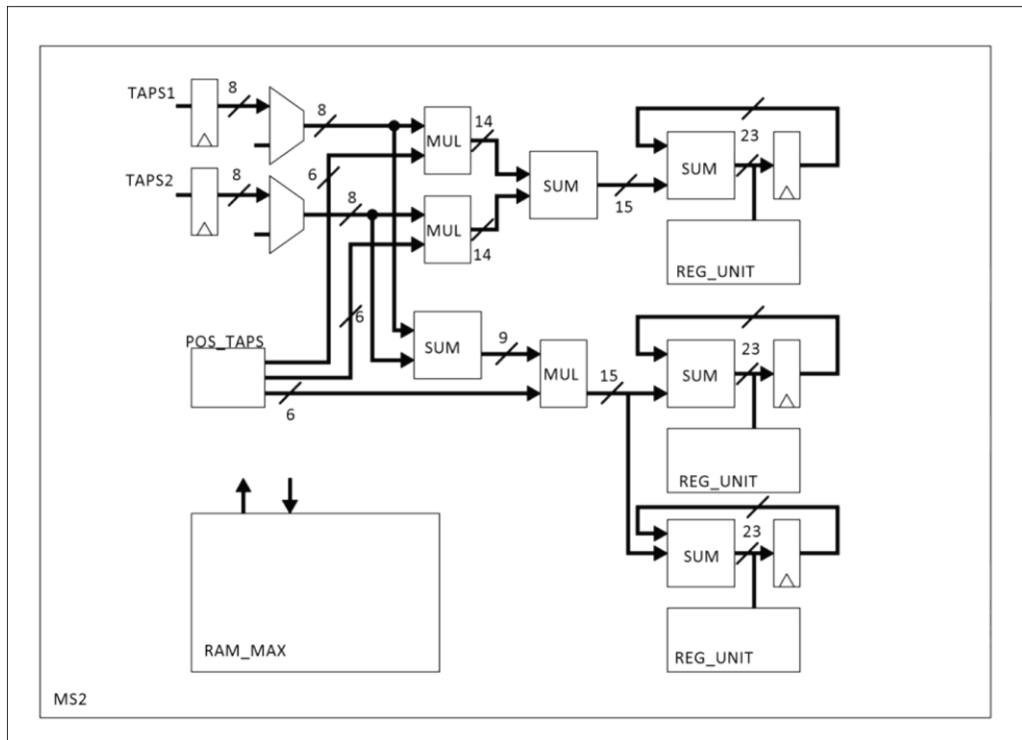


Figura 33 – Bozza circuito interno MS2

Nota: I blocchi aritmetici illustrati in figura sono stati istanziati utilizzando i moduli IP forniti gratuitamente da Xilinx.

5.1.1.2.1 Elaborazione ed Accumulazione

Analizziamo, innanzitutto, il funzionamento dei primi blocchi di accumulazione ed elaborazione dati. Come è visibile dalle figure seguenti, il processo si può suddividere in due fasi distinte ma sequenziali.

Fronte di Salita (*STROBE*)

Fronte di Discesa (*STROBE*)

- Fronte di Salita

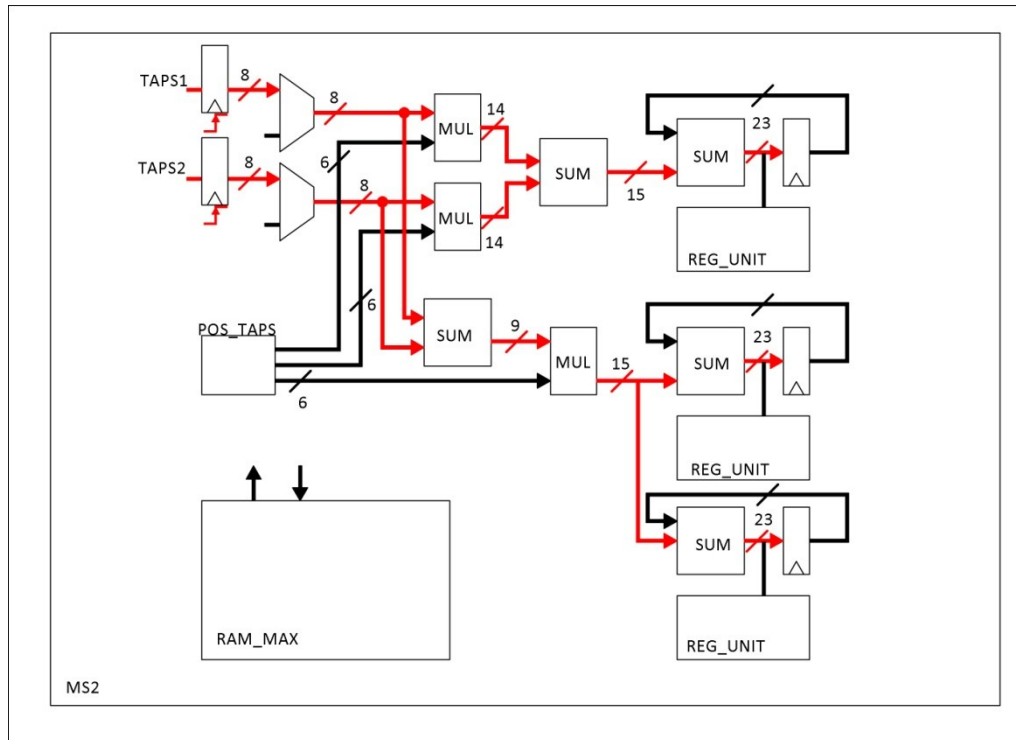


Figura 34 – Percorso dati al fronte di salita

Data la modalità di funzionamento dei segnali dallo standard camera link sappiamo che, dato l'enable dai segnali FVAL ed LVAL, ad ogni fronte di salita del segnale di strobe corrisponde l'invio di due dati validi da parte della camera CCD. Memorizzati i dati dai registri posti in ingresso, questi vengono immediatamente elaborati dai blocchi aritmetici successivi, aggiornando di fatto la sommatoria dell'algoritmo della *Media Statistica*.

- Fronte di Discesa

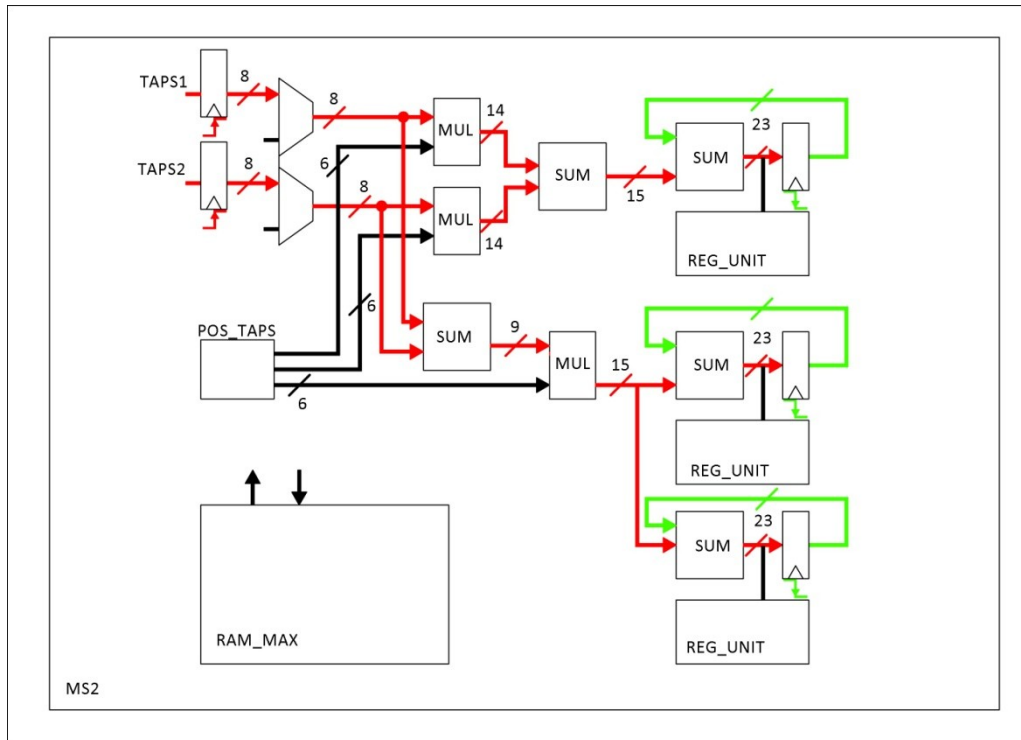


Figura 35 - Percorso dati al fronte di discesa

Al fronte di discesa vengono salvati i dati parziali ottenuti durante la fase precedente.

Da sottolineare come la logica, fin qui presentata, sia funzionale ad un'elaborazione sequenziale e continua dei dati in ingresso, a "qualsiasi" frequenza.

Lo standard Camera Link impone una lettura dei pixel riga per riga, in sequenza e con modalità che cambiano in base alla camera utilizzata. La "suddivisione" del CCD tramite la griglia virtuale comporta quindi, il problema di salvare temporaneamente i risultati parziali, provenienti dall'analisi di ogni riga di cella della griglia virtuale.

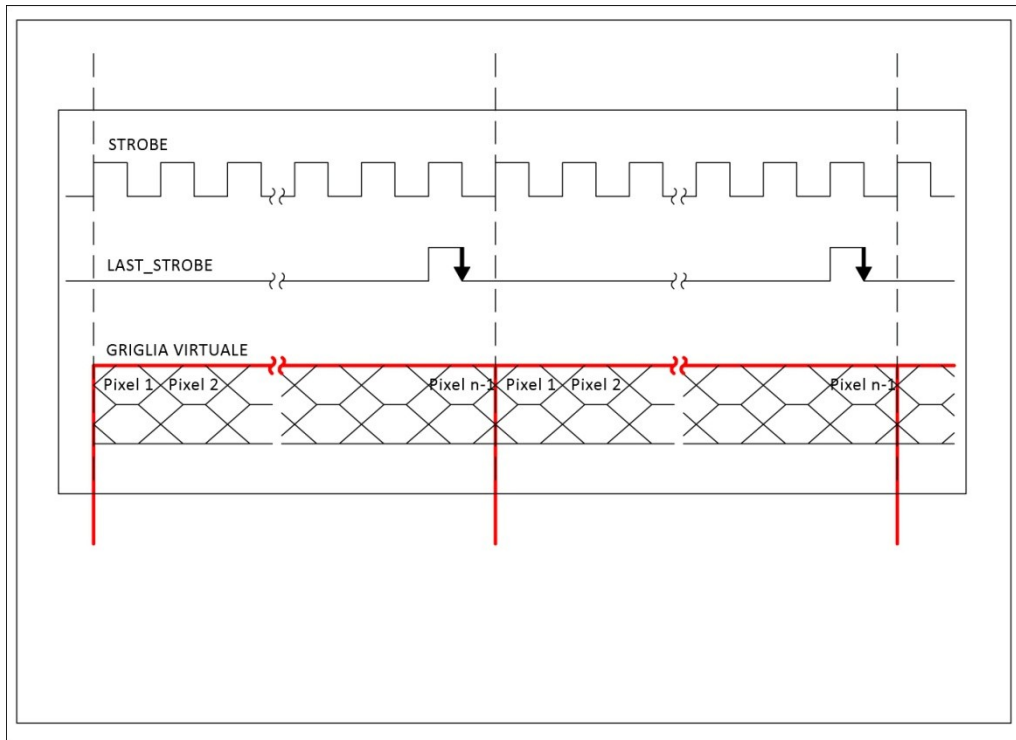


Figura 36 – Suddivisione virtuale matrice CCD

A tale scopo è stato quindi aggiunto al blocco, poco prima descritto, una serie di registri capaci di memorizzare i dati accumulati a fine elaborazione riga, sfruttando l'ultimo strobe appartenente alla cella analizzata.

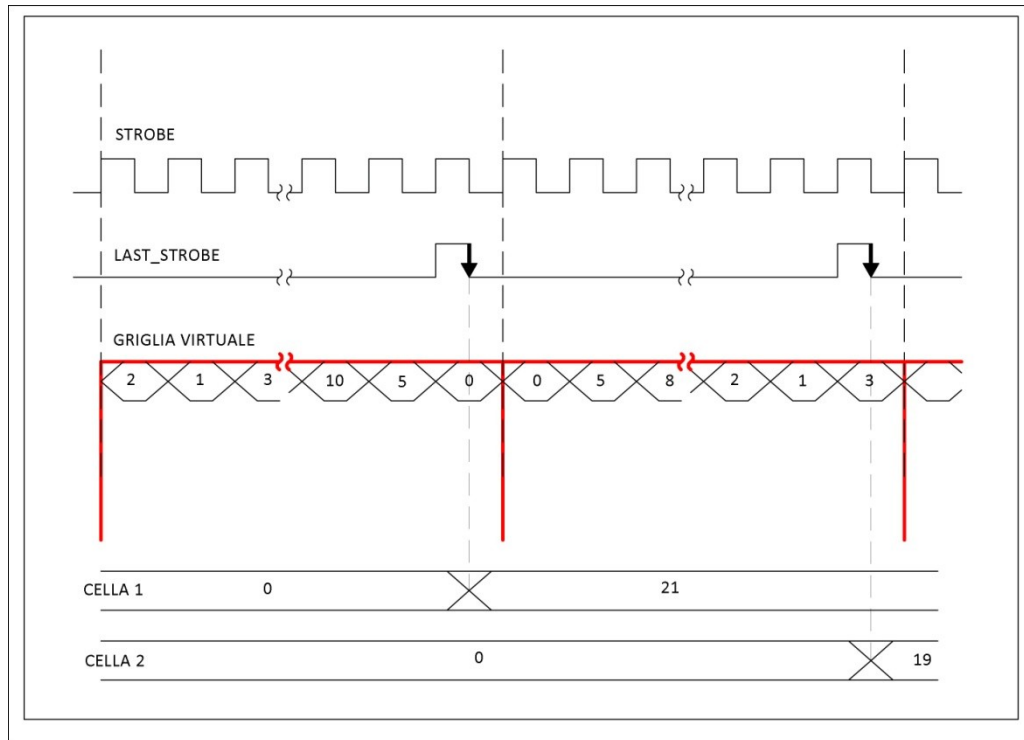


Figura 37 – Esempio di funzionamento

Allo stesso tempo vengono letti durante le elaborazioni delle righe seguenti in modo tale da consentire un'accumulazione dei pixel elaborati delle intere celle.

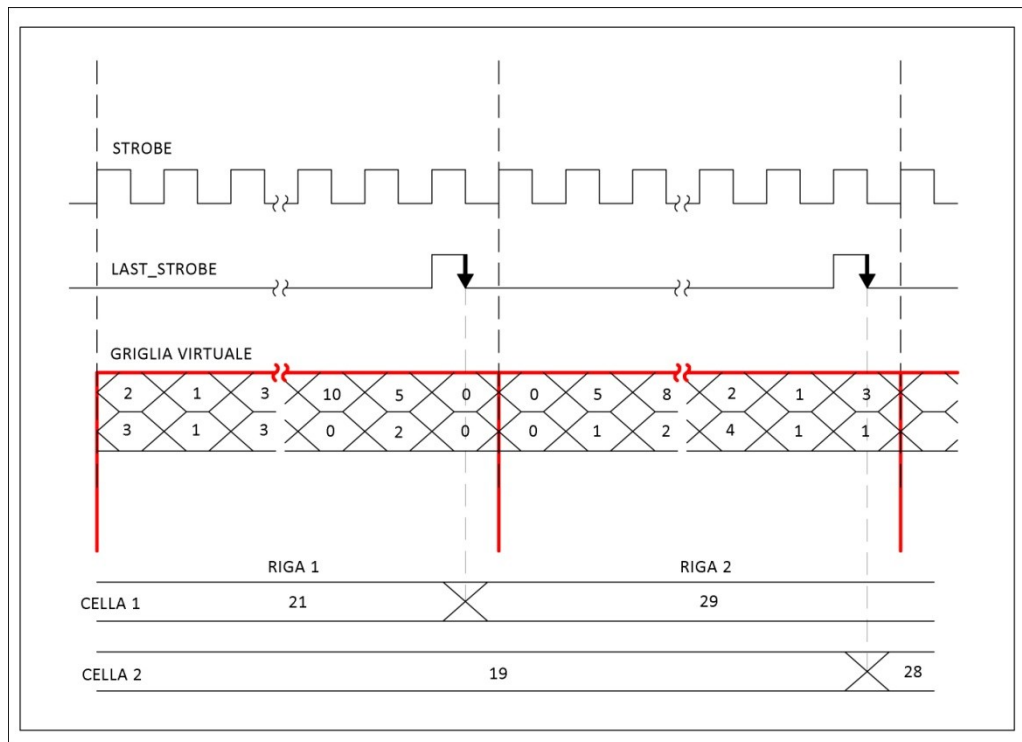


Figura 38 – Esempio di funzionamento

Questo permetterà quindi di arrivare alle sommatorie sia nell'asse x che y che serviranno per completare l'algoritmo di detection.

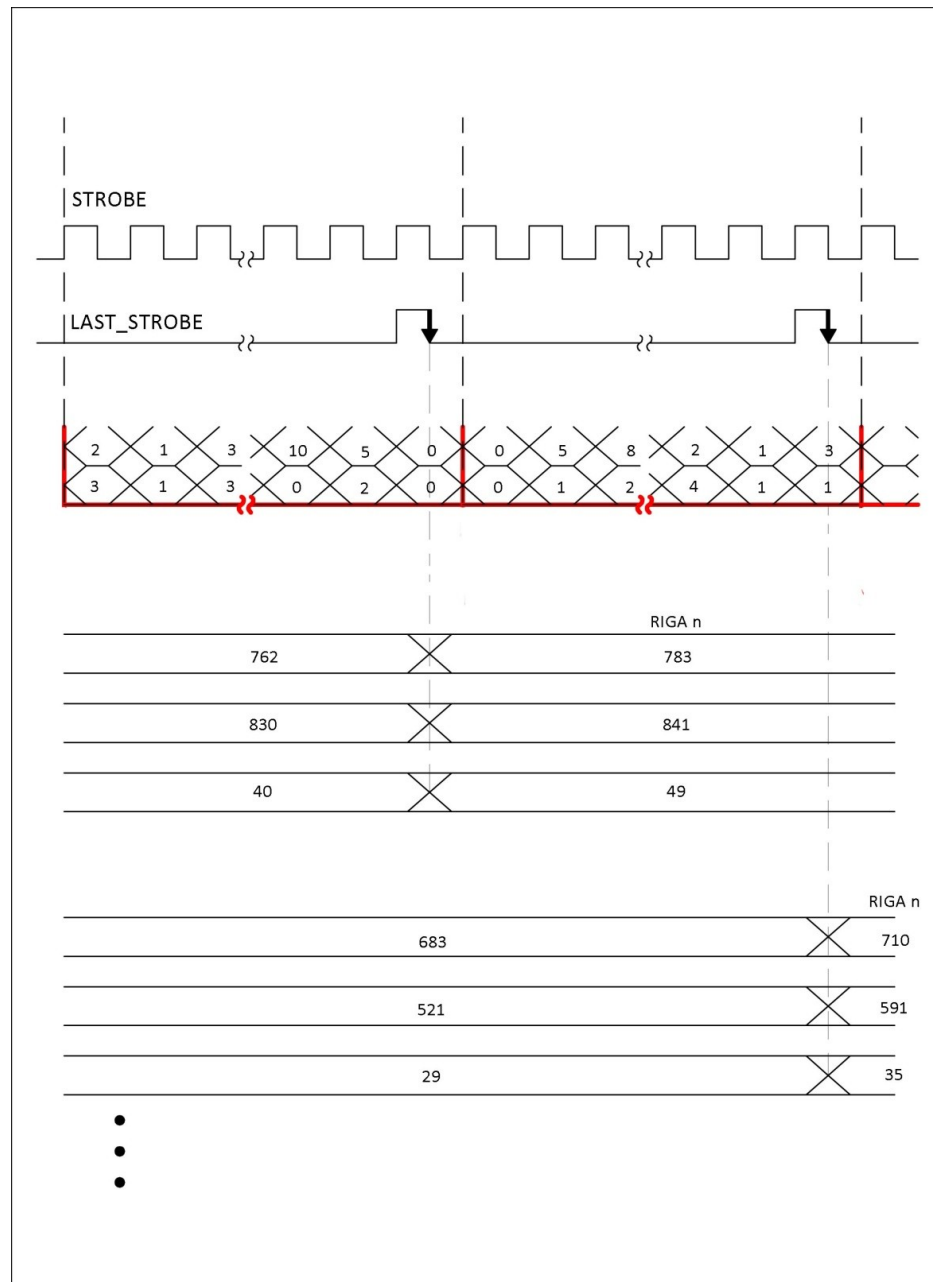


Figura 39 – Esempio di funzionamento

Il meccanismo di lettura e salvataggio dati parziali continua fino a fine cella. Essendo i pixel letti riga per riga dal CCD, ci troveremo alla fine ad avere $3 \cdot N$ **Dati** (Sommatorie: $\sum P_X I$, $\sum P_Y I$, $\sum I$), con N numero colonne del lenslet utilizzato.

Finita l'ultima riga verranno quindi salvati i risultati ottenuti nei registri d'uscita dell'entity e resettati i registri del blocco aritmetico da opportuni segnali generati.

5.1.1.2.2 RAM_MAX (Soglia Mobile)

Il meccanismo che consente all' algoritmo di sfruttare l' utilizzo della soglia mobile è invece attuato dal blocco RAM_MAX.

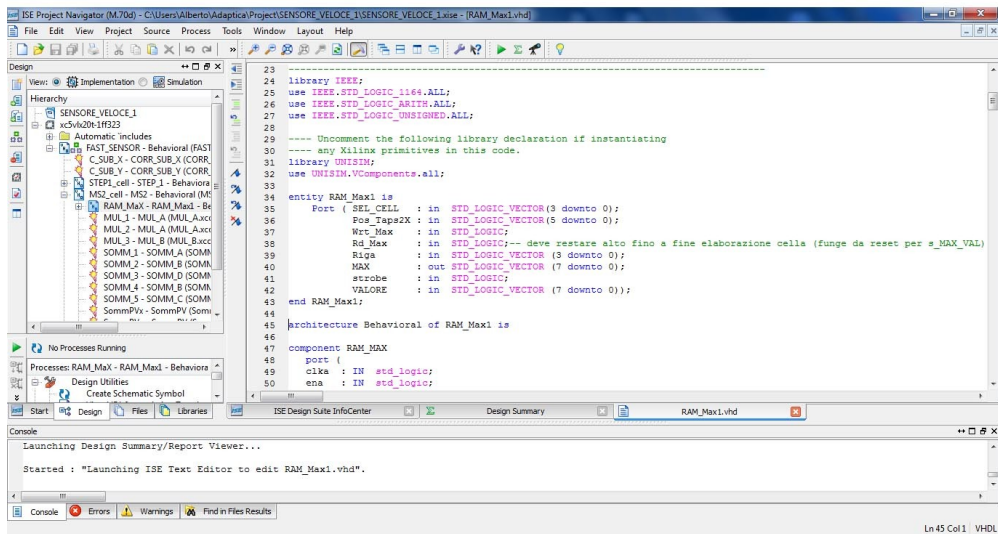


Figura 40 - Immagine Xilinx Entity RAM_MAX

Il principio di funzionamento è simile a quello attuato per la memorizzazione dei dati parziali.

Durante la lettura di ogni cella viene trovato e memorizzato il pixel con il valore più elevato ed allo stesso tempo viene moltiplicato per una costante in modo tale da ricavarne una percentuale (soglia al X%).

Ad ogni fine elaborazione celle, viene salvata in una memoria ram la parola contenente i valori calcolati.

Al frame successivo i valori calcolati verranno letti e resi disponibile al blocco di analisi garantendo un valore aggiornato della soglia da applicare all' analisi degli spot.

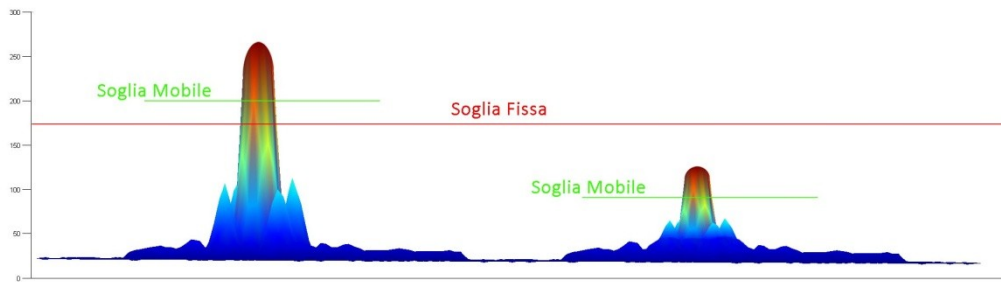


Figura 41 – Differenza di funzionamento tra soglia fissa e mobile

Nota: In RAM_MAX viene anche inserita la possibilità di fissare una soglia **minima**, sotto la quale non andare. In caso di basse intensità, da parte degli spot analizzati, la soglia applicata non sarà una percentuale X di un livello vicino allo zero, ma una soglia minima. Questo permetterà di discriminare le condizioni di “spot assente o debole”.

$$Threshold = Max[(I(X_i, Y_j) * 25\%), 3NL]$$

Come è possibile osservare dalla figura, imporre una soglia fissa può portare alla perdita di spot poco intensi.

5.1.2 DIVs

Il blocco MS2 esegue una pre-analisi dei dati, provenienti dal sensore, e ad ogni riga di celle analizzate fornisce in uscita i suoi risultati.

La divisione di tali risultati spetta al blocco successivo DIVs.

La divisione a livello hardware sottintende un’architettura ciclica con tempi di latenza diversi da zero. Il fatto di avere $2*N$ divisioni da effettuare non risulta essere un problema in quanto la struttura pipeline garantisce tempo sufficiente per completare tutti i calcoli anche con clock rate applicati al divisore molto bassi.

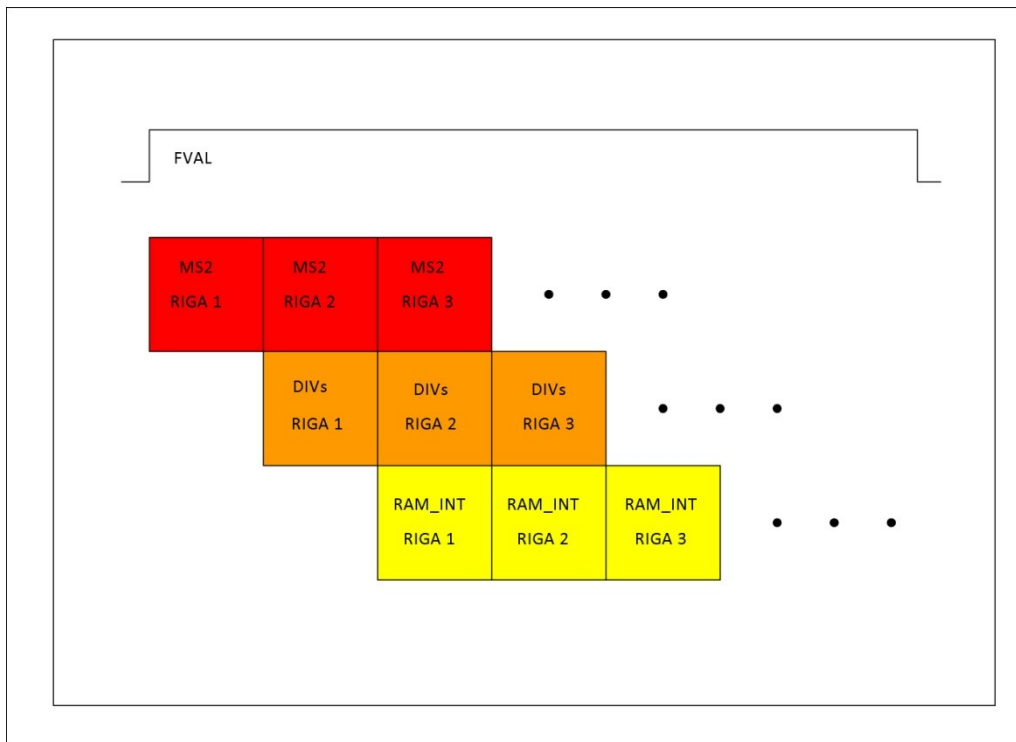


Figura 42 – Struttura funzionamento Pipeline

Come si può vedere dalla figura il tempo, massimo, necessario a completare le operazioni di divisione dei dati si può stimare nel tempo necessario al blocco precedente (MS2), ad eseguire la scansione dei dati di un'intera riga di celle. Tempo che risulta essere più che sufficiente per il completamento di tutte le divisioni.

Come si può vedere nella figura sottostante, i dati in ingresso vengono elaborati sequenzialmente a frequenze ben al di sotto della frequenza di strobe.

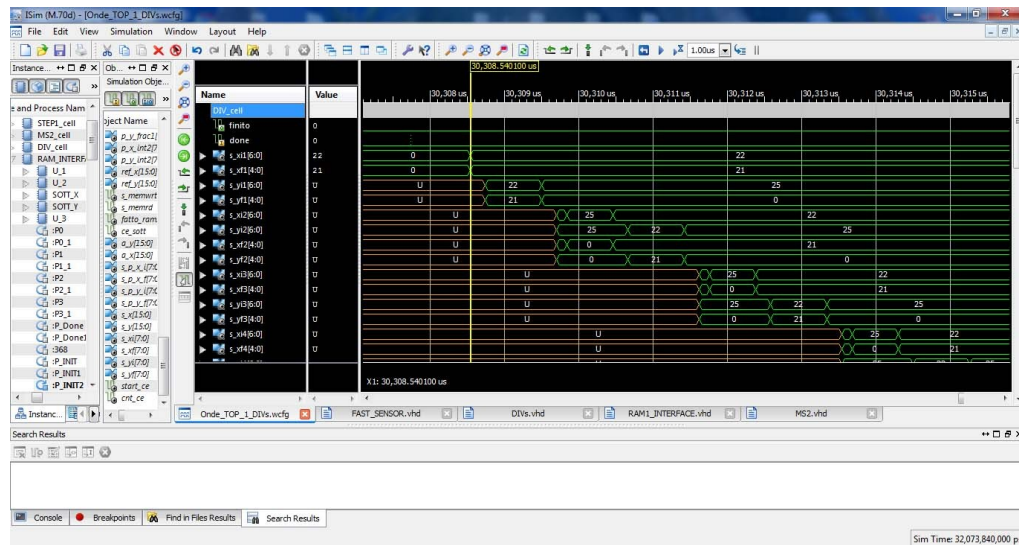


Figura 43 – Andamento sequenziale divisioni Blocco Divs

La stima delle tempistiche e quindi della scelta del clock da applicare al divisore, dipendono fortemente da molteplici fattori come: cicli di clk necessari alla divisione ($N_{CLK/DIV}$), frame rate sensore, numero di colonne lenslet ($N^{\circ}DIV$), dimensione in pixel celle. In base alla frequenza di *strobe* della modalità e della camera scelta sarà inoltre possibile stimare il tempo necessario ad elaborare un'intera riga di celle (T_{MS2}).

$$CLK_{DIV} \geq \left(T_{MS2} / (N_{CLK/DIV} * N^{\circ}DIV) \right)^{-1}$$

ES: Per lo sviluppo di questa tesi si è ipotizzato l'utilizzo di:

- 1 camera veloce (100 fps) con risoluzione 1280x1024 e con frequenza di strobe massima di 80MHZ
- 1 Lenslet 13*14 con dimensione del pitch di 60x60 pixel

Questo si può tradurre in:

$$\frac{N_{CLK}}{DIV} = 24$$

$$N^{\circ}DIV = 13$$

$$T_{MS2} = \frac{1}{80MHz} * \frac{1400}{2} * 60 = 525\mu s$$

Che porta ad avere un CLK minimo, per il funzionamento del blocco divisore, di:

$$CLK_{DIV} \geq \left(T_{MS2} / (N_{CLK/DIV} * N^{\circ}DIV) \right)^{-1} \cong 0.5MHz$$

Valore più che accessibile per un dispositivo FPGA.

DIVs è costituito da una modulo IP del divisore (tempo latenza = 24 Clock) e da una logica di controllo costituita fondamentalmente da contatori.

5.1.2.1 ENTITY

Andiamo a vedere quelli che sono ingressi e uscite della entity in questione:

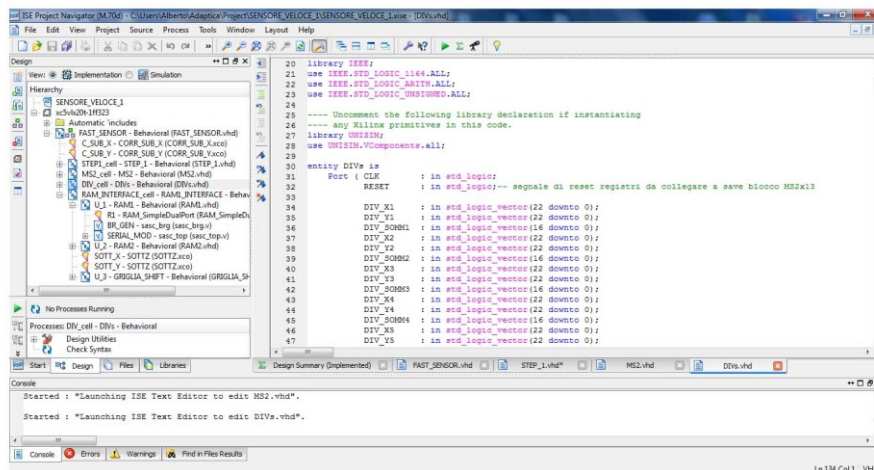


Figura 44 - Immagine Xilinx Entity DIVs

- **INGRESSI**
- CLK : Segnale di Clock funzionamento ip_core del divisore
- RESET : Reset Asincrono (attivo alto) dei Registri interni
- DIV_Xn (n= 1:13): Dividendo coordinata X proveniente da MS2
- DIV_Yn (n= 1:13): Dividendo coordinata Y proveniente da MS2
- DIV_Somm (n= 1:13): Divisore proveniente da MS2
- Done : Segnale che (al fronte di salita 0->1) avvia ciclo divisione dati

- **USCITE**

- XIn (n= 1:13): Risultato parte *INTERA* divisione coordinata X
- YIn (n= 1:13): Risultato parte *INTERA* divisione coordinata Y
- XFn (n= 1:13): Risultato parte *FRAZIONARIA* divisione coordinata X
- YFn (n= 1:13): Risultato parte *FRAZIONARIA* divisione coordinata Y
- FINITO : Segnale di concluso ciclo divisioni

I segnali d'ingresso del blocco DIVs sono tutti provenienti da MS2 (tranne il CLK) e sono costituiti principalmente da risultati parziali oltre che ad il segnale "DONE" che funge da start per l'elaborazione dei dati in ingresso.

In uscita saranno disponibili le parti intere e frazionarie dei dati presenti in ingresso ed un segnale di fine divisione "FINITO".

5.1.2.2 Struttura interna

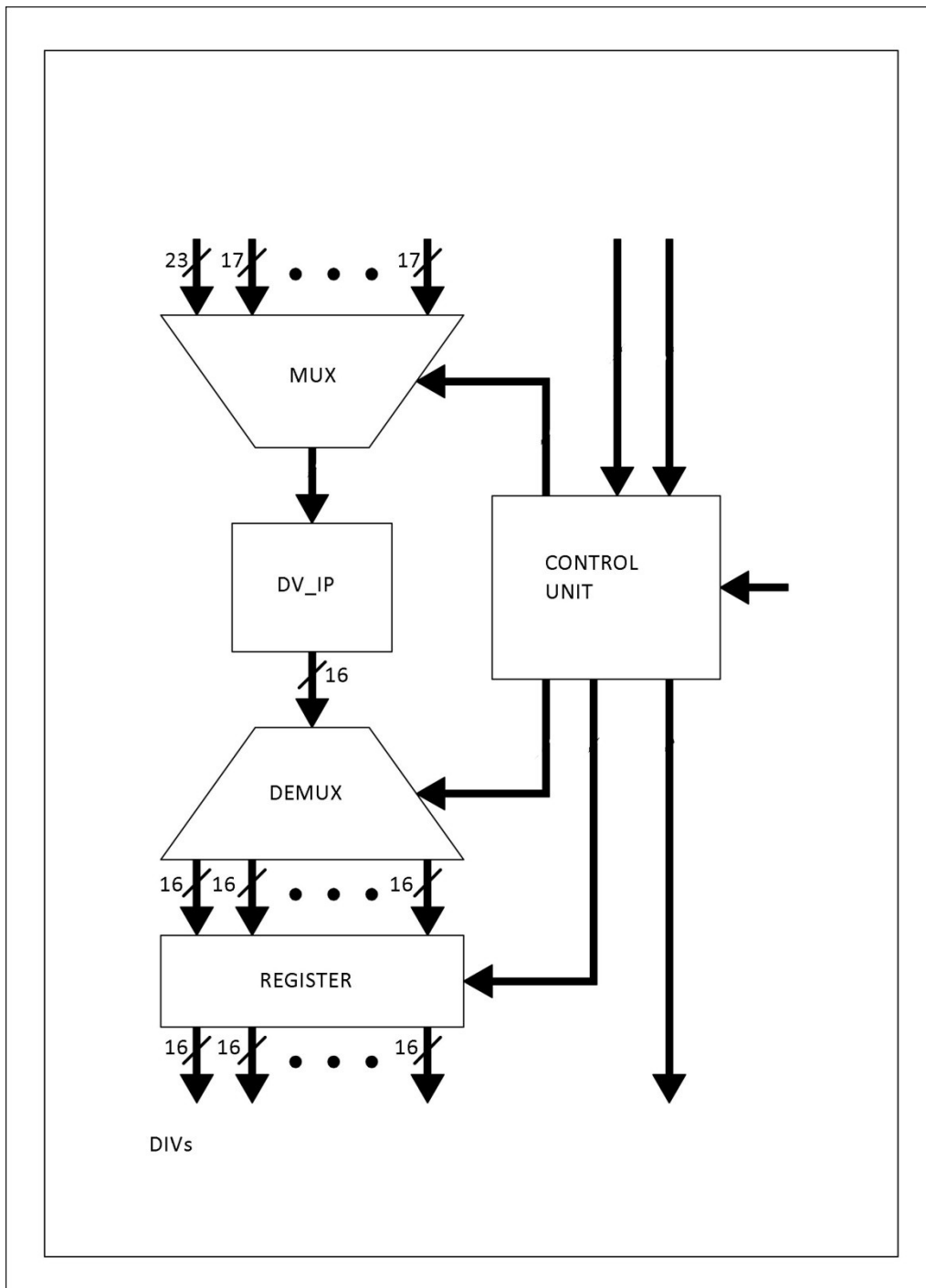


Figura 45 – Struttura interna blocco DIVs

L'architettura del blocco DIVs ruota interamente attorno al core (DV_IP della Xilinx), inserito fra una serie di tre multiplexer (MUX), che hanno il compito di smistare i dati parziali provenienti da MS2, e il duale demultiplexer (DEMUX).

In uscita al DEMUX vengono posizionati dei registri che avranno il compito di

“salvare” i risultati ottenuti.

Tutti questi componenti sono controllati da una macchina a stati finiti (*CONTROL*) la quale, sollecitata dai segnali d’ingresso, ha il compito di generare le forme d’onda necessarie a selezionare sequenzialmente i dati d’ingresso, eseguire l’operazione di divisione ed indirizzare e salvare i risultati nei rispettivi registri.

5.1.3 RAM1_INTERFACE

Ultimo blocco della pipeline, ma per questo non il meno importante, esegue molteplici funzioni; lettura/scrittura dati, elaborazione, smistamento e verifica delle condizioni limite.

Comunica in ingresso con il blocco DIVs ed in uscita con STEP_1 e tutta la fase di *inizializzazione* in generale, inoltre predispone e si occupa di gestire la lettura dei risultati (le coordinate dei centroidi) tramite porta seriale.

5.1.3.1 ENTITY

Andiamo a vedere quelli che sono ingressi e uscite della entity in questione:

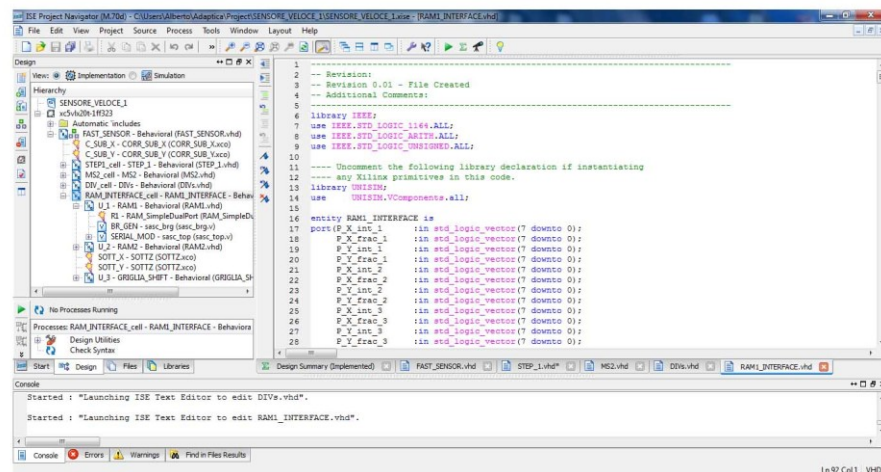


Figura 46 - Immagine Xilinx Entity RAM1_INTERFACE

- **INGRESSI**
 - CLK : Segnale di Clock
 - CLK50 : Segnale di Clock per gestione Tx Seriale
 - FVAL : Segnale usato come Reset Asincrono (attivo basso) dei Registri interni
 - P_X_int_n (n= 1:13) : Risultato Intero coordinata X proveniente da DIVs

- P_X_frac_n (n = 1:13) : Risultato Frazionario coordinata X proveniente da DIVs
 - P_Y_int_n (n= 1:13) : Risultato Intero coordinata Y proveniente da DIVs
 - P_Y_frac_n (n = 1:13): Risultato Frazionario coordinata Y proveniente da DIVs
 - INIT : Se '1' indica la fase di *INIZIALIZZAZIONE* se '0' *FUNZIONAMENTO A REGIME*
 - FINITO : Segnale che (al fronte di salita 0->1) avvia ciclo di scrittura in ram
- **USCITE**
 - FLAG1 : Bit di *flag* per segnalazione di Riga/Colonna vuota (fase *INIZIALIZZAZIONE*)
 - FLAG2 : Bit di *flag* per segnalazione di Riga/Colonna vuota (fase *INIZIALIZZAZIONE*)
 - FLAG3 : Bit di *flag* per segnalazione di Riga/Colonna vuota (fase *INIZIALIZZAZIONE*)
 - FLAG4 : Bit di *flag* per segnalazione di Riga/Colonna vuota (fase *INIZIALIZZAZIONE*)
 - SHIFT_X : Coordinata X del vettore spostamento (fase *INIZIALIZZAZIONE*)
 - SHIFT_Y : Coordinata Y del vettore spostamento (fase *INIZIALIZZAZIONE*)
 - DONE_Shift : Segnale di dati (SHIFT_X/Y) disponibili
 - Pin_txd_o : Pin trasmissione seriale
 - Pin_i : Pin ricezione seriale ("open")

I segnali provenienti in ingresso sono quasi esclusivamente provenienti dal blocco DIVs.

Oltre ai normali segnali di clock e di reset (rappresentato dal segnale FVAL), l'entity ricevi in ingresso le coordinate dei centroidi calcolati al blocco precedente, comprensivi di parte intera e frazionaria ed un segnale di *start* che avvia il processo di scrittura ed elaborazione dei segnali nella memoria ram. Il segnale "INIT" modifica la funzionalità del blocco in base alla modalità di funzionamento ('1' = inizializzazione , '0' = a regime).

I segnali in uscita si possono dividere in due gruppi:

Il primo gruppo si collega al blocco di inizializzazione fornendo i segnali necessari a completare le fasi garantendo il corretto posizionamento della griglia di riferimento.

Il secondo, pensato come primo debug del sistema, gestisce la trasmissione dei risultati parziali mediante l'uso di una porta seriale.

5.1.3.2 *Struttura interna*

RAM1_INTERFACE ha un'architettura molto semplice e che ben si presta ad essere adattata ad eventuali cambiamenti come ad esempio l'aumento del numero degli spot da analizzare.

Al suo interno sono istanziati due sottrattori e due memorie ram (*RAM1* e *RAM2*) oltre che al blocco *GRIGLIA_SHIFT*. Il tutto è controllato da un'opportuna logica di controllo e di controllo dei dati.

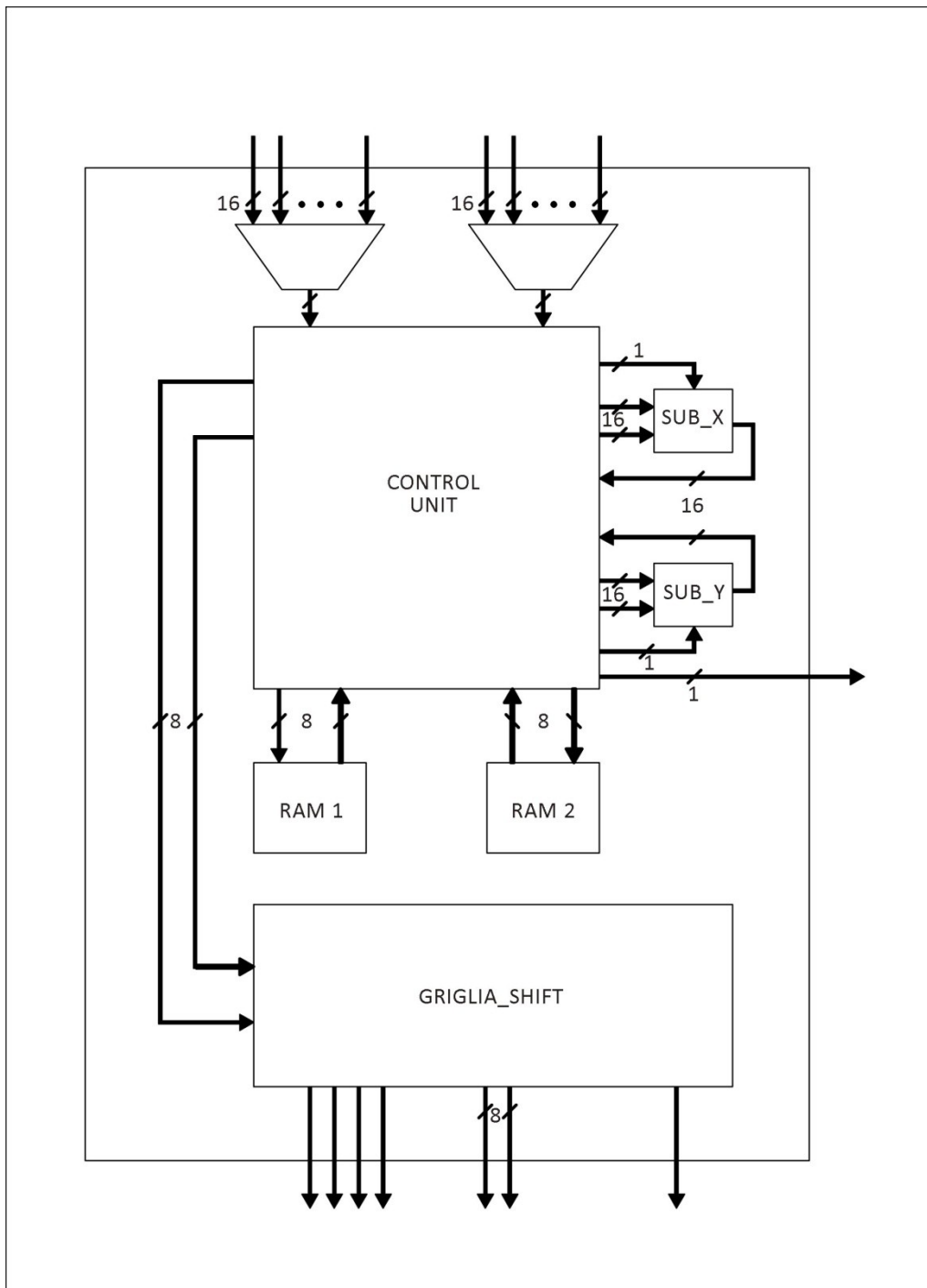


Figura 47 – Struttura interna blocco RAM1_INTERFACE

La modalità di funzionamento del sistema si riflette fortemente sul funzionamento di questo blocco.

In fase di inizializzazione il flusso di dati provenienti da DIVs viene indirizzato solamente alla memoria RAM1.

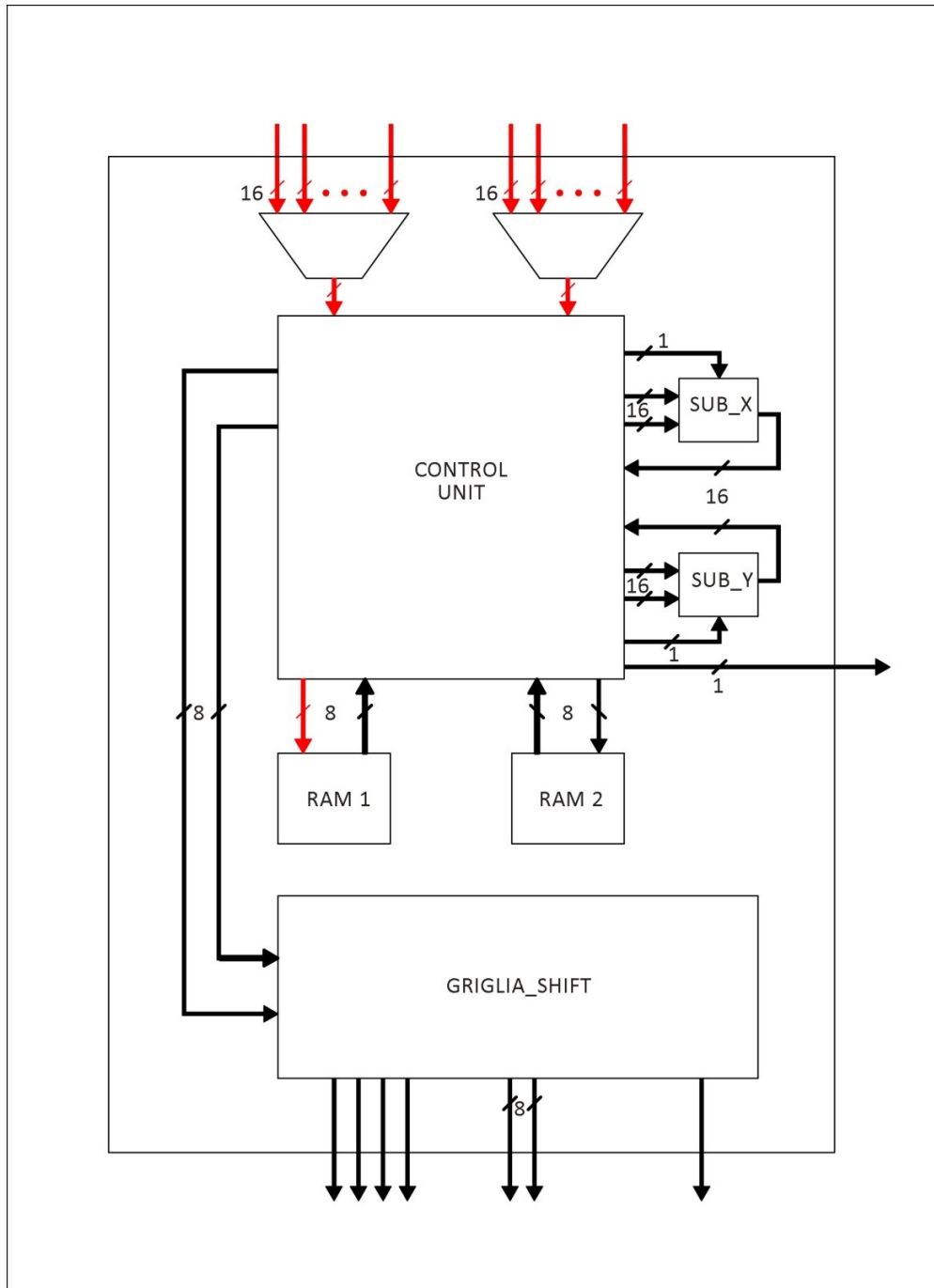


Figura 48 – Flusso dati in fase di inizzalzzazione

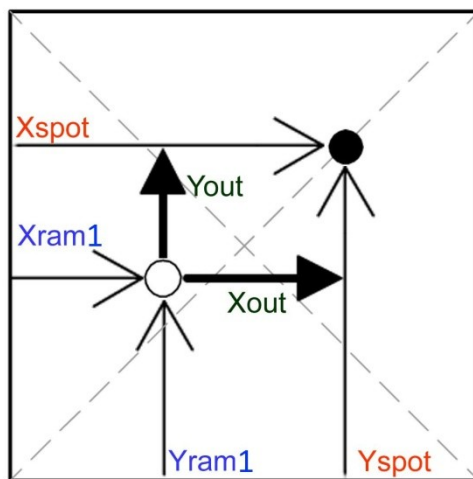
Il resto dell'architettura resterà inerte in attesa della modalità di funzionamento a regime.

Questo tipo di scelta è stata fatta nell'ottica dell'acquisizione da parte del sistema, durante la sua inizializzazione, di una **griglia di riferimento** sulla quale poi confrontare tutte le altre misure ottenute.

Le coordinate memorizzate in RAM1 saranno quindi da considerarsi come i *centri relativi*, ossia come origine per le misurazioni successive.

$$X_{out} = X_{spot} - X_{ram1}$$

$$Y_{out} = Y_{spot} - Y_{ram1}$$



LEGENDA

- SPOT CORRENTE
- SPOT DI RIFERIMENTO

Figura 49 – Funzionamento spot di riferimento

A regime il funzionamento sarà esattamente l'opposto. Ai dati provenienti dagli ingressi saranno quindi sottratti i riferimenti memorizzati in fase di inizializzazione nella memoria RAM1. I risultati verranno poi memorizzati nella memoria dual port RAM2 la quale a fine frame verrà completamente letta dalla porta seriale.

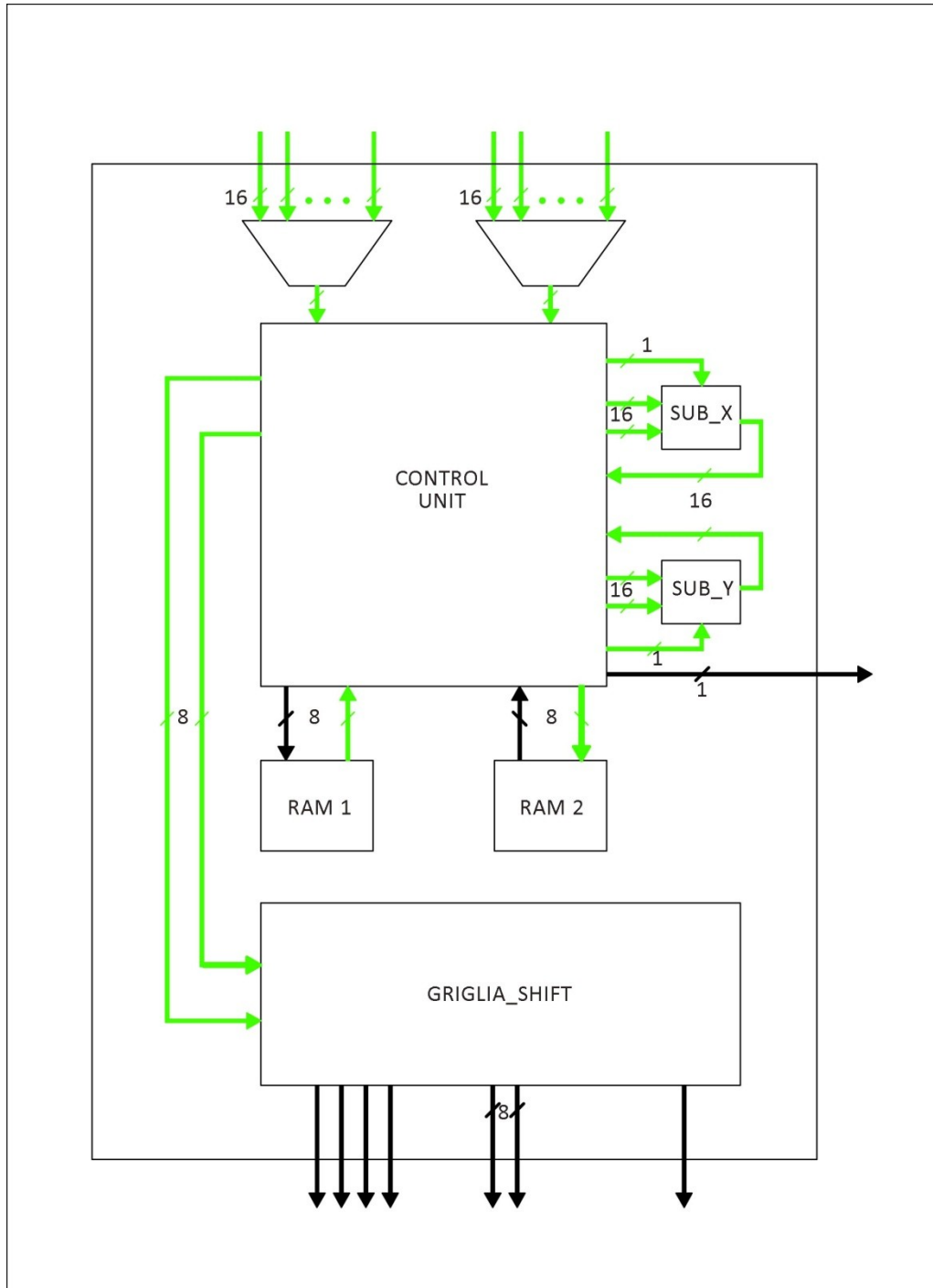


Figura 50 – Flusso dati fase a regime

Il blocco **GRIGLIA_SHIFT** ha un compito molto importante all'interno della fase di inizializzazione.

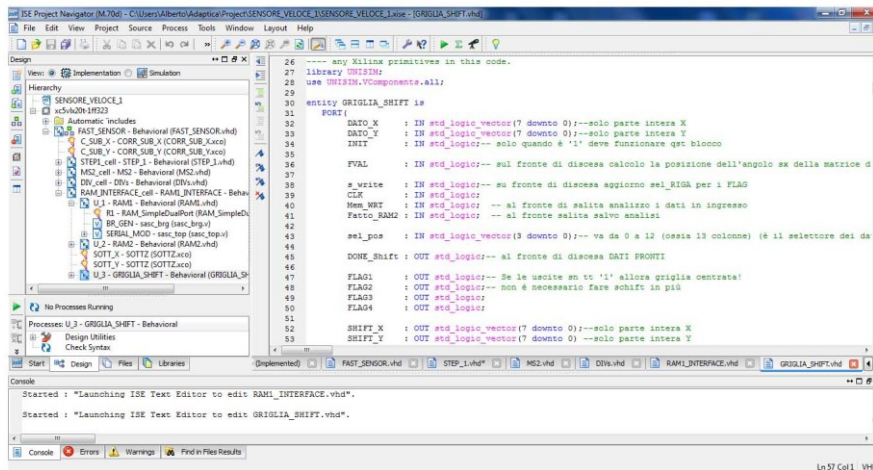


Figura 51 - Immagine Xilinx Entity GRIGLIA_SHIFT

In base alla fase dell'inizializzazione questo blocco esegue una serie di analisi sui dati elaborati ai blocchi precedenti i quali forniscono le informazioni necessarie a determinare nel modo più corretto possibile, il corretto posizionamento della griglia virtuale di riferimento.

VETTORE SPOSTAMENTO

Nella prima fase di posizionamento della griglia virtuale di riferimento ha una grande importanza, analizzare ed individuare, dopo un primo "grossolano" posizionamento, l'entità generale dello spostamento della griglia.

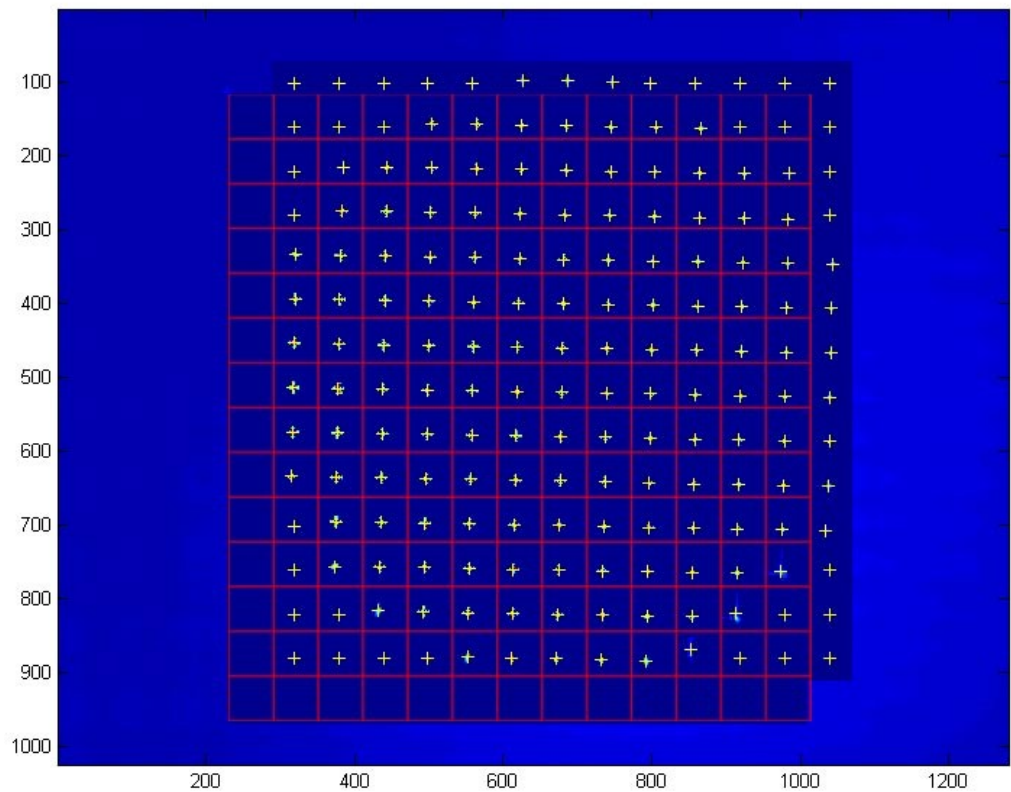


Figura 52

Naturalmente posizionare una griglia in base al posizionamento di un numero considerevole di punti posizionati non esattamente ai centri delle rispettive aperture, come avviene naturalmente nei casi reali di utilizzo dei sensori Shack-Hartmann, non è una procedura rapida ed immediata. Il metodo più semplice e preciso è quello di andare a considerare le distanze fra ogni spot e il centro ideale della sua rispettiva cella.

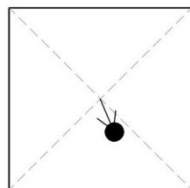


Figura 53 – Simulazione distanza spot da centro cella

Considerando tale distanza come “l’errore” del posizionamento della griglia virtuale rispetto al primo posizionamento, è facile capire come facendone una media sia possibile ricavare la più corretta misura della CORREZIONE.

$$\mathbf{Vettore\ Correzione}_{X/Y} = \frac{\sum \mathbf{Vettori\ Spostamento}_{X/Y}}{\sum \mathbf{Numero\ Spot}_{X/Y}}$$



Figura 54 – Utilizzo vettore correzione

ANALISI CONDIZIONE LIMITE

Oltre al calcolo del *vettore correzione* è necessario andare a verificare una particolare condizione che in alcuni casi può determinare un funzionamento errato del sistema di posizionamento della griglia virtuale. Come spiegato nei capitoli precedenti, un errore introdotto nella prima fase di posizionamento della griglia troppo elevato potrebbe non essere più corretto.

$$\frac{-\Delta_{Pitch}}{2} < \text{Errore}_{X/Y} < \frac{\Delta_{Pitch}}{2}$$

Allo scopo di correggere anche questi eventuali errori è stato inserito nel blocco *GRIGLIA_SHIFT* un sistema di controllo in grado di controllare la prima e l'ultima riga e la prima e l'ultima colonna della griglia segnalando con dei *FLAGS* la presenza o meno del problema in questione.

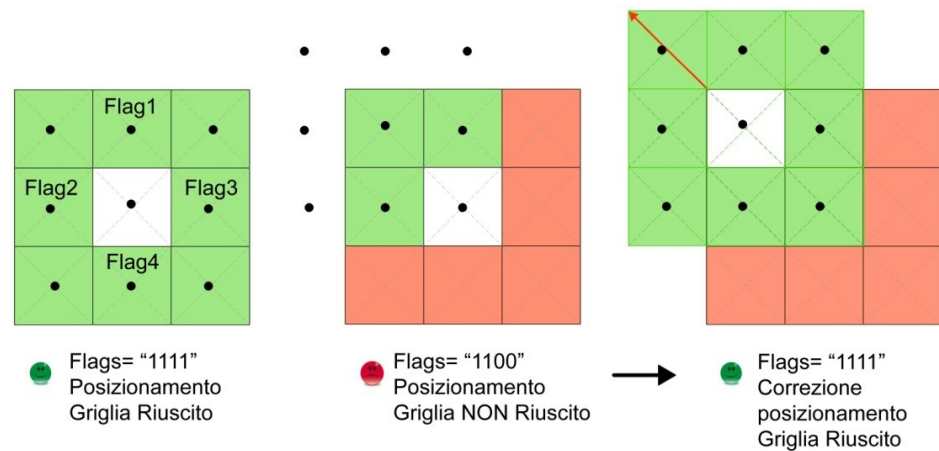


Figura 55 – Andamento analisi errore

In base alla segnalazione dei flags il sistema di posizionamento sarà in grado di effettuare la correzione.

5.2 FASE 1

Come spiegato in precedenza la *fase 1* si occupa del posizionamento della griglia virtuale di riferimento che per molti versi è risultata, nonostante la sua semplice funzione, uno dei meccanismi più complessi da implementare all'interno del progetto.

Posizionare correttamente tale griglia a partire da un'immagine in cui sono presenti molteplici spot di dimensione, posizionamento, numero variabile non è semplice da implementare in un FPGA, soprattutto se si pensa al linguaggio a basso livello con cui questi dispositivi vengono programmati.

Se si guarda la figura è possibile vedere come ad esempio, nei sensori Shack Hartmann il reale funzionamento sia condizionato da molteplici fattori sia di utilizzo ma anche di fabbricazione.

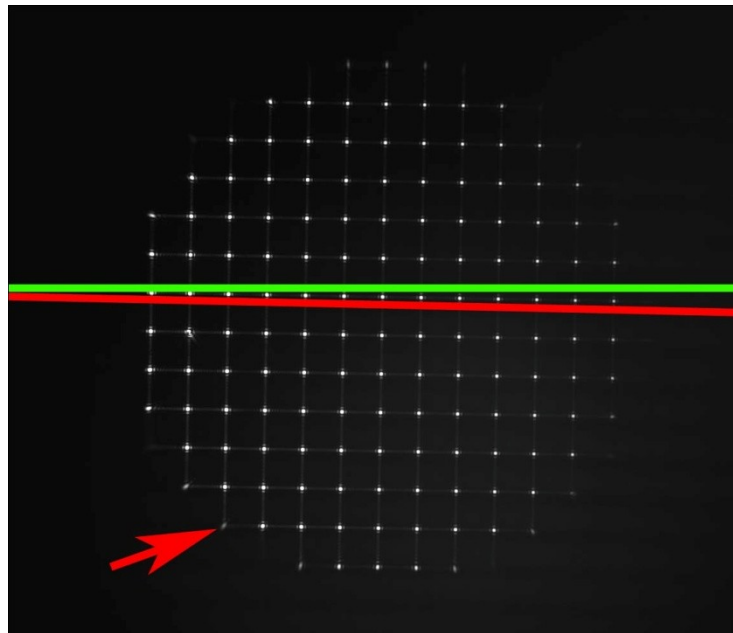


Figura 56 – Figura reale Shack-Hartmann

Sempre osservando la figura è facilmente intuibile come, ad esempio, un non preciso posizionamento del lenslet rispetto al ccd della camera vada a provocare una rotazione indesiderata (linea verde posizione corretta, linea rossa posizione reale) degli spot rispetto agli assi XY dell'immagine.

Inoltre il fatto di posizionare un apertura circolare prima del lenslet, spesso rischia di compromettere l'integrità dei segnali degli spot posizionati nel bordo più esterno (freccia rossa).

Tutte queste problematiche rendono difficile individuare un algoritmo capace di essere flessibile e di adattarsi a queste variabili di funzionamento.

L'idea più semplice ed efficace è stata quella di implementare lo stesso algoritmo della media statistica, applicato a regime sui singoli spot, sull'intera matrice CCD.

In questo modo è stato possibile individuare correttamente la posizione del baricentro degli spot e tramite la conoscenza (in pixel) del pitch delle aperture, avere un primo posizionamento della griglia.

Nonostante l'idea sia teoricamente percorribile in pratica questo porta diverse problematiche in termini, soprattutto, di area occupata nell'FPGA.

Se pensiamo che il solo blocco MS2, che si occupa di accumulare i prodotti parziali, per celle di al massimo 60x60 pixel occupa nell'FPGA (Virtex 5) il 9% delle risorse disponibili è facile intuire quale sarebbe la percentuale per creare un blocco simile capace di analizzare un intero CCD!!

Era evidente come tutto ciò imponesse la necessità di trovare una nuova soluzione.

Soluzione Posizionamento Griglia Virtuale

L'idea si è sviluppata puntando ad ottenere un sistema di posizionamento della griglia virtuale "leggero"(in termini di area occupata nell'FPGA) ma allo stesso tempo preciso e flessibile.

L'intuizione fondamentale è stata quella di utilizzare parte della stessa architettura utilizzata per l'implementazione dell'algoritmo di detection (raffinata e precisa) anche nella fase 1.

In questo modo sarebbe stato necessario disegnare solo un blocco di appoggio a quello già esistente, diminuendo drasticamente l'area occupata ed allo stesso tempo mantenendo un'elevata precisione.

L'algoritmo in questione viene eseguito in più steps analizzando una sequenza limitata di frame.

Ad ogni passo la precisione del posizionamento della griglia viene migliorato fino ad ottenere un risultato ottimale.

I passi fondamentali sono 3 e possono essere sintetizzati come segue:

- 1- Media Statistica "Semplificata"
- 2- Correzione tramite il "Vettore Correzione"
- 3- Correzione Perdita Spot

Media Statistica "Semplificata"

Come detto in precedenza pensare di applicare l'algoritmo della media statistica su di un'intera matrice CCD richiederebbe l'utilizzo di aritmetiche dedicate capaci di moltiplicare, sommare e dividere "parole" con un numero considerevole di bit. Tutto ciò si tradurrebbe inevitabilmente nell'utilizzo di architetture molto complesse e considerevoli dal punto di vista delle risorse utilizzate.

Dato l'approccio inefficiente la soluzione utilizzata è stata quella di modificare tale algoritmo rendendolo meno impegnativo. In particolare si è andati a rappresentare il valore dei singoli pixel ad 1bit e non più ad 8, come normalmente fatto.

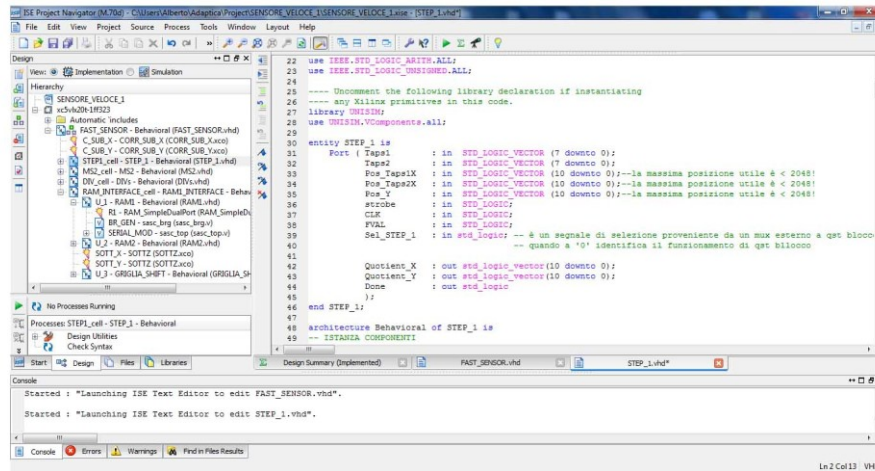


Figura 57 - Immagine Xilinx Entity STEP_1

$$\begin{array}{ccc}
 \sum \text{Valore}_i(8 \text{ bit}) * \text{Posizione}_i & \xrightarrow{\text{SEMPLIFICO}} & \sum \text{Valore}_i(1 \text{ bit}) * \text{Posizione}_i \\
 \sum \text{Valore}_i(8 \text{ bit}) & \begin{array}{c} \text{SETTANDO A '1' TUTTI} \\ \text{I VALORI AL DI SOPRA} \\ \text{DI UNA SOGLIA FISSATA} \end{array} & \sum \text{Valore}_i(1 \text{ bit})
 \end{array}$$

Questo si traduce immediatamente in una riduzione della lunghezza della moltiplicazione Posizione*Valore e quindi anche delle sommatorie dell'algoritmo.

La riduzione di 7 bit viene eseguita andando a binarizzare il frame utilizzando una soglia prestabilita e fissa.

In questo modo gli spot dell'immagine acquisita dal CCD della camera passano da una forma tipicamente proporzionale al sinc² a dei cilindri come illustrato nelle seguenti figure 3D di una stessa immagine reale prima e dopo la sogliatura.

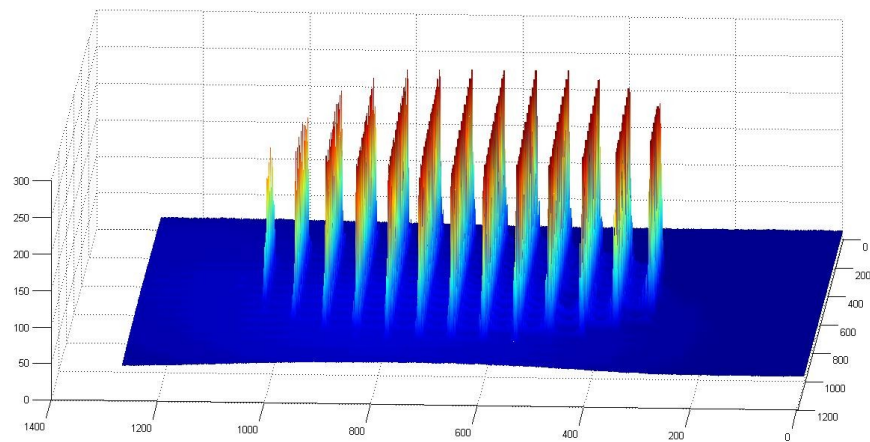


Figura 58 – Figura reale matrice di spot

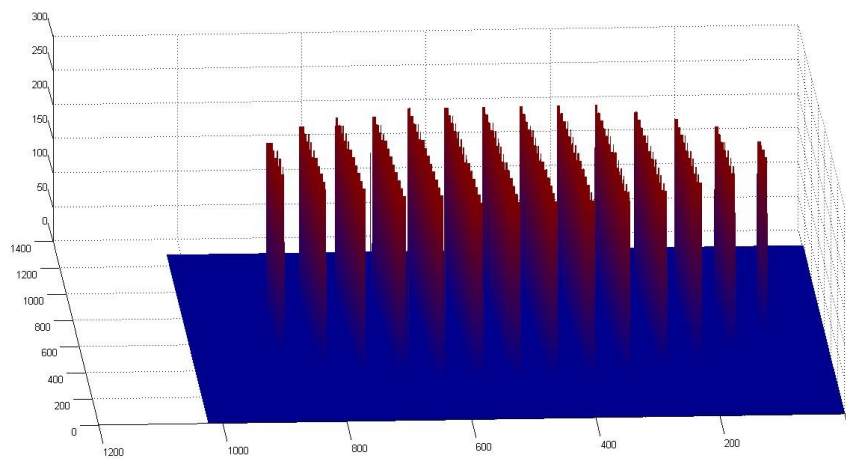


Figura 59 – Figura binarizzata matrice di spot

Osservando più in dettaglio un singolo spot è possibile avere una migliore idea dell'effetto della binarizzazione degli spot.

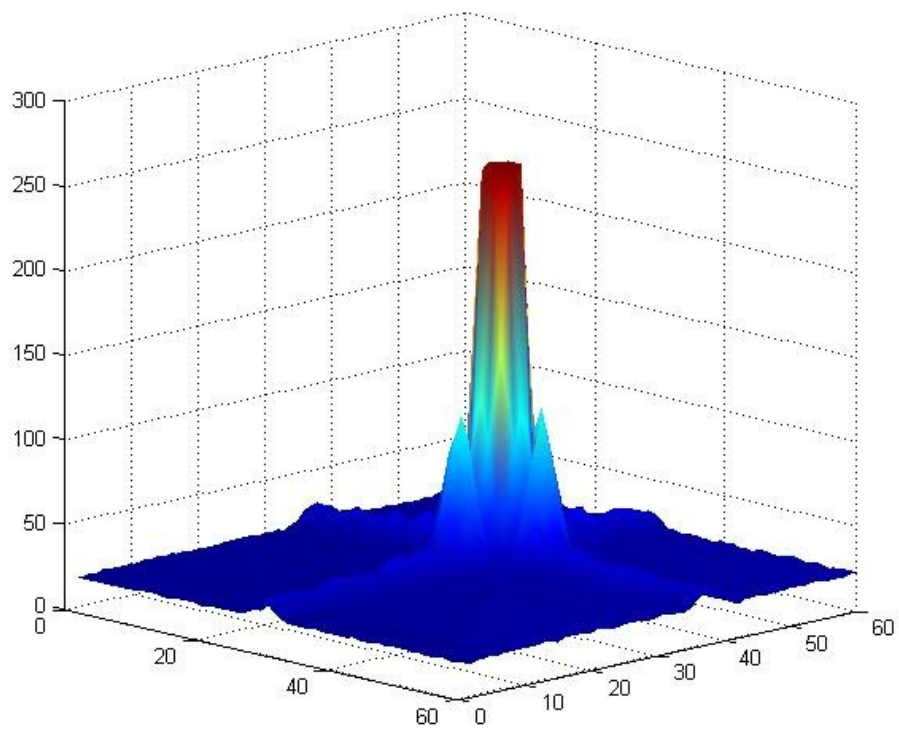


Figura 60 – Singolo spot reale

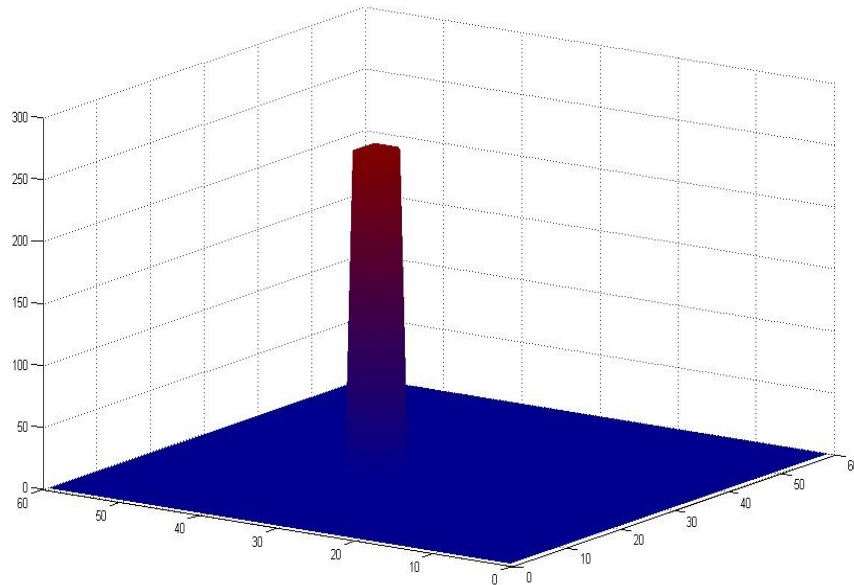


Figura 61 – Singolo spot binarizzato

L'inconveniente principale che si ottiene da una procedura simile è data dalla larghezza dei cilindri (e quindi degli spot) imposta all'altezza del livello di soglia. Infatti avere sezioni di larghezza diverse si va a tradurre inevitabilmente in un errore indotto nel calcolo del baricentro. Errore che si deve accettare come contropartita per il miglioramento dell'occupazione d'area e che verrà compensato negli step successivi.

Finito l'analisi di questo frame ci troviamo quindi al punto di avere un primo posizionamento, *non ottimale*, della griglia virtuale.

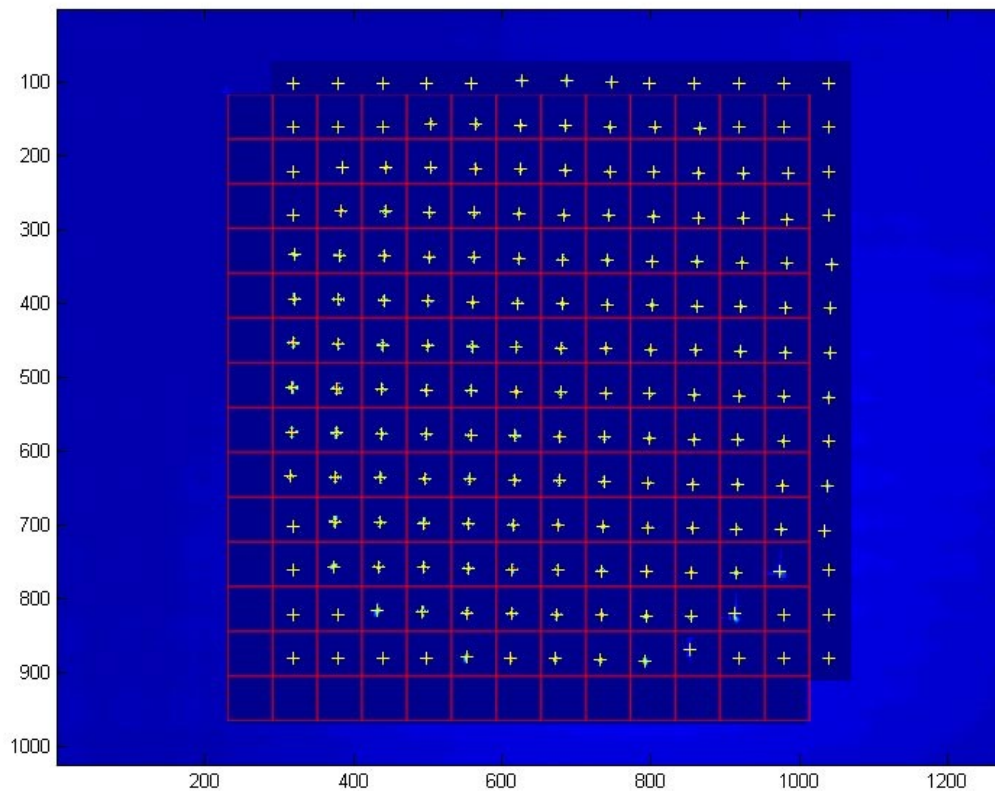


Figura 62 – Posizionamento dopo primo step

È infatti possibile notare come l'errore introdotto, in questa prima fase, vada a far perdere una quantità considerevole di spot e quindi di informazioni.

Correzione tramite il "Vettore Correzione"

Come detto, il passo precedente restituisce un posizionamento grossolano della griglia virtuale.

Il punto fondamentale, arrivati a questo punto, è quello di trovare il metodo per sistemare il posizionamento di tale griglia.

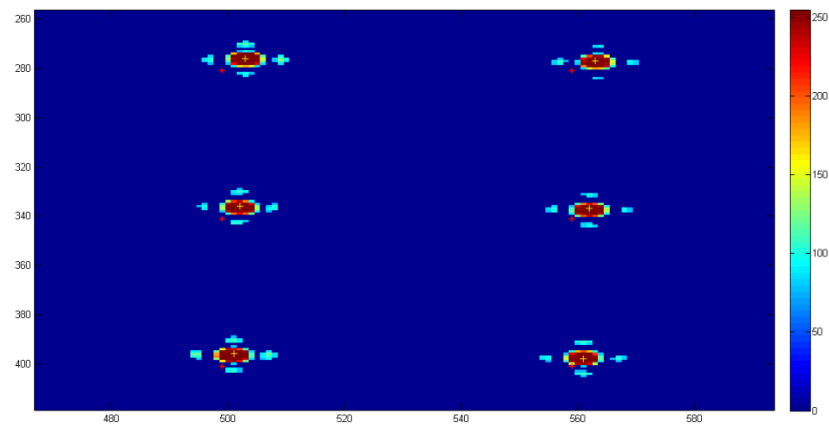


Figura 63 – Analisi spot tramite algoritmo Media Statistica

Come si può vedere dalla simulazione di figura la griglia risulta posizionata non correttamente, infatti i centri degli spot (croci gialle) sono lontani dalla loro posizione ideale rappresentata dai centri delle rispettive aperture (croci rosse).

Supponendo che l'errore iniziale non superi $\pm \text{pitch}/2$ (argomento che verrà trattato successivamente) la correzione si traduce nel poter calcolare la distanza media degli spot dal corrispondente centro ideale e spostare la griglia di conseguenza.

Per calcolare tale "vettore correzione" è stato inserito il blocco GRIGLIA_SHIFT (precedentemente descritto) in RAM1_INTERFACE il quale, proprio durante la fase 1, esegue una media dei centroidi calcolati rispetto al centro ideale.



Figura 64 – Correzione tramite vettore correzione

Le coordinate saranno poi riportate in uscita ed utilizzate, sottraendole alle coordinate del vertice della griglia di riferimento, per correggerne e migliorarne il posizionamento.

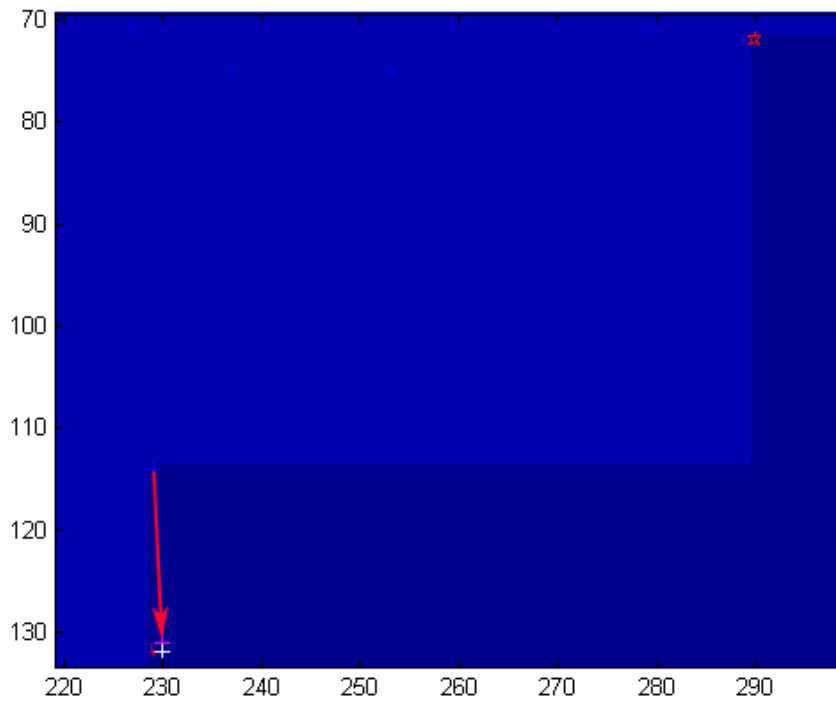


Figura 65- Le croci blu, rosa e bianca rappresentano le 3 iterazioni con il vettore correzione

NOTA: Per ottenere un risultato stabile bisogna replicare la correzione più volte. Da simulazioni Matlab® risultano sufficienti 3 iterazioni.

Completata la fase di correzione tramite il vettore spostamento avremo una griglia posizionata il più possibile correttamente rispetto alla media delle posizioni dei singoli centroidi. In questo modo è possibile correggere anche eventuali rotazioni degli spot dovuti ad errori di fabbricazione nel posizionamento del lenslet rispetto al CCD.

Correzione Perdita Spot

Conclusa anche la fase di correzione tramite il vettore spostamento esiste un ultimo controllo da perpetrare prima di dichiarare conclusa la fase1 di inizializzazione.

La fase di correzione tramite vettore spostamento ha il compito di avvicinare il più possibile, in media, la posizione degli spot reali con i rispettivi centri ideali delle celle di appartenenza.

Purtroppo esiste la possibilità che in alcuni casi il primo posizionamento della griglia sia accompagnato da un errore troppo elevato per essere corretto successivamente, in particolare abbiamo che per:

$$\frac{-\Delta_{Pitch}}{2} < Errore_{x/y} < \frac{\Delta_{Pitch}}{2}$$

l'errore che si propaga è esattamente pari alla lunghezza del pitch(in pixel) in una o in entrambe le posizioni.

Questo si traduce nella perdita di dati che non verranno analizzati dall'algoritmo, non appartenendo ai pixel sottesi dalla griglia di riferimento.

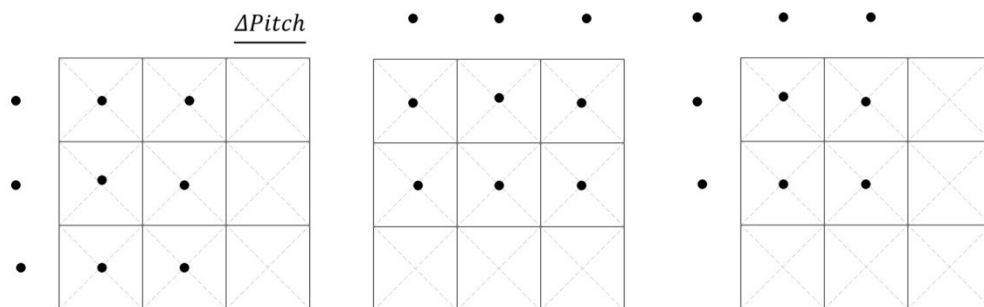


Figura 66 – Esempi perdita dati

Per cercare di correggere tale errore è stata introdotta una fase di controllo che ha lo scopo di cercare la presenza o meno di spot all'interno della prima ed ultima riga e della prima ed ultima colonna.

Il blocco GRIGLIA_SHIFT tra i vari compiti ha anche quello di verificare questa situazione settando o meno dei flags.

Se tutti i flags sono settati a '1' esiste almeno spot su ogni riga e colonna esterna della griglia.

Se almeno uno dei flags è settato a '0' esiste almeno una riga o una colonna priva di spot!

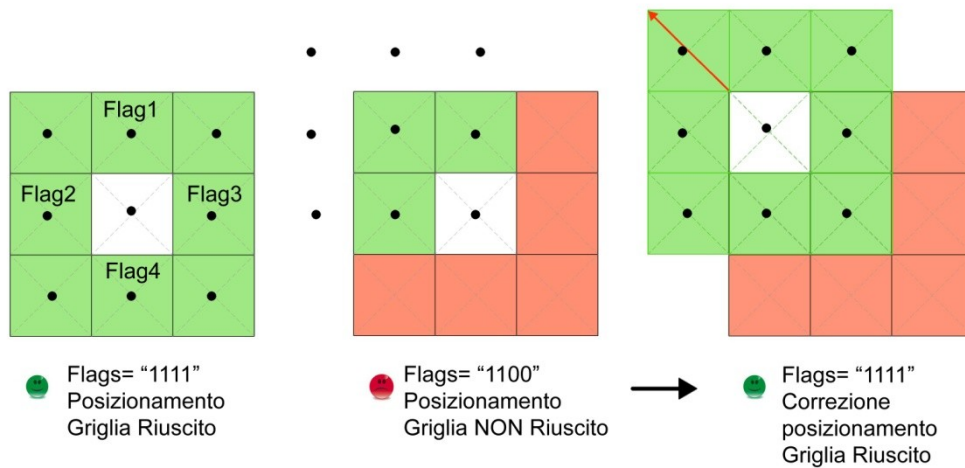


Figura 67 – Andamento correzione perdita spot

In questo caso la griglia deve essere risistemata e spostata di quantità pari alla dimensione dei pitch.

In particolare se osserviamo l'esempio illustrato in figura 68 possiamo vedere come la mancanza di spot nell'ultima colonna e riga impongono uno spostamento sia in X che Y dell'intera griglia.

I blocchi aggiunti alla struttura base di detection dei centroidi sono, come detto precedentemente, pochi e dall'architettura semplice:

5.2.1 STEP_1

È l'unico vero blocco logico aggiunto alla struttura base. Esegue un'unica scansione dell'intero CCD, applicando l'algoritmo della Media Statistica "Semplificata", ed estrapolando quindi, le coordinate del baricentro della figura analizzata.

Anche in questo blocco, come in MS2, viene applicato nell'algoritmo un meccanismo di soglia (questa volta "fisso") per cercare di limitare l'influsso del rumore dalla misurazione.

5.2.1.1 ENTITY

Andiamo ora a vedere quelli che sono i segnali di ingresso e uscita:

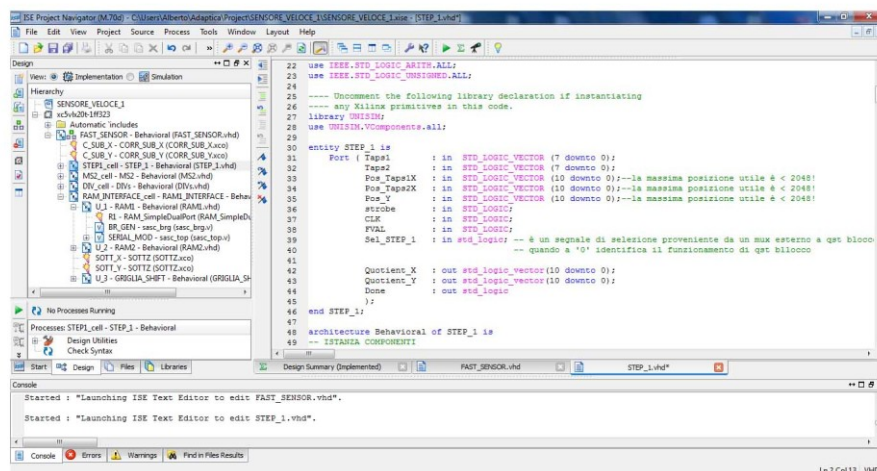


Figura 68 - Immagine Xilinx Entity STEP_1

- **INGRESSI**
- Taps1 : Dato ad 8 bit proveniente dalla camera del sensore (vedi Camera Link)
- Taps2 : Dato ad 8 bit proveniente dalla camera del sensore (vedi Camera Link)
- FVAL : Frame valid (vedi Camera Link)
- Strobe : segnale di sincronismo (vedi Camera Link)

- Pos_Taps1X : Dato ad 11 bit (dimensione massima analizzabile CCD < 2048)
- Pos_Taps2X : Dato ad 11 bit (dimensione massima analizzabile CCD < 2048)
- Pos_Y : Dato ad 11 bit (dimensione massima analizzabile CCD < 2048)
- Sel_STEP_1 : segnale dal blocco di controllo
- CLK : clock 50MHz

- **USCITE**

- Done : segnale di fine elaborazione avvenuta
- Quotient_X : risultati ottenuti a fine elaborazione celle coordinata X
- Quotient_Y : risultati ottenuti a fine elaborazione celle coordinata Y

Gli ingressi sono principalmente formati dai segnali provenienti dall'interfaccia Camera Link del sensore Shack-Hartmann i quali vengono analizzati dal blocco solo in presenza dell'abilitazione del segnale di "Sel_STEP_1". Se questo segnale è attivo basso, è permesso il ciclo di analisi mentre se al contrario è nello stato logico '1' vengono disabilitati tutti i blocchi matematici presenti al suo interno e le uscite vengono mantenute stabili da registri formati da FF.

5.2.1.2 Struttura Interna

L'architettura interna è del tutto duale a quella presente nel blocco MS2. Le differenze si possono notare nell'assenza della struttura di analisi e memorizzazione della soglia da applicare agli spot e soprattutto nella diminuzione dei bit dei dati da elaborare.

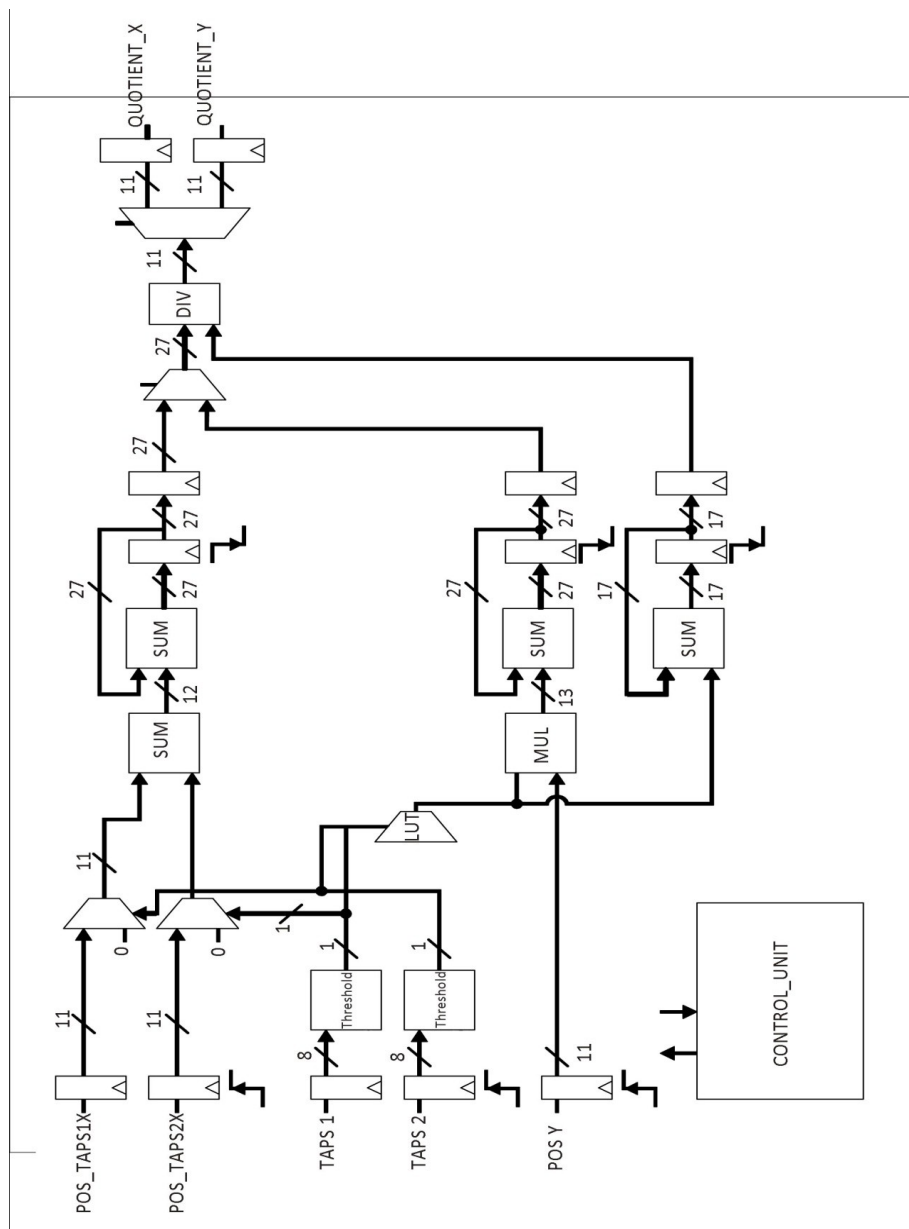


Figura 69 – Struttura interna STEP_1

5.2.2 CORR_SUB_X/Y

I blocchi CORR_SUB_X e Y fanno parte della struttura complementare per la realizzazione dell'algoritmo di posizionamento della griglia virtuale di riferimento. Non sono vere e proprie entity ma istanze di sottrattori che verranno utilizzati nel ciclo della Fase1 per raffinare, nei vari step, le coordinate esatte del vertice estremo sinistro della griglia virtuale.

I divisori sono creati attraverso dei tool forniti dalla Xilinx (CoreGenerator) grazie ai quali è possibile modellare, in base alle proprie esigenze, tutte le più comuni tipologie di interfacce e blocchi aritmetici.

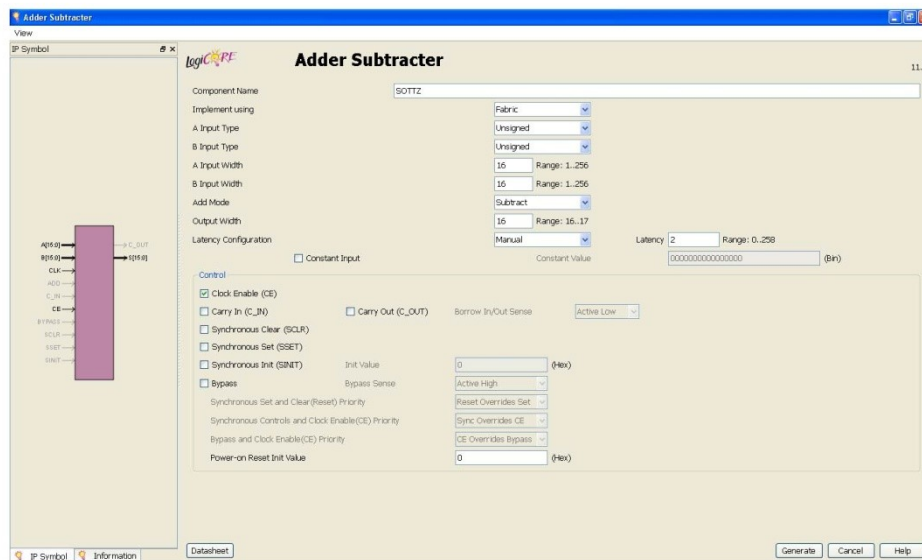


Figura 70 – Esempio utilizzo CoreGenerator

5.2.3 CONTROL

È costituito da una FSM (Macchine a stati finiti) che ha il compito di sincronizzare il funzionamento di tutti i blocchi presenti nel progetto e di guidare i dati provenienti dal sensore Shack-Hartmann attraverso le strutture delle due differenti fasi di funzionamento.

Dopo il segnale di RESET la FSM esegue un conteggio dei vari frame in arrivo, ognuno dei quali identifica una particolare azione della fase1 (*Inizializzazione*) fino alla sua conclusione e quindi al funzionamento a regime (fase2).

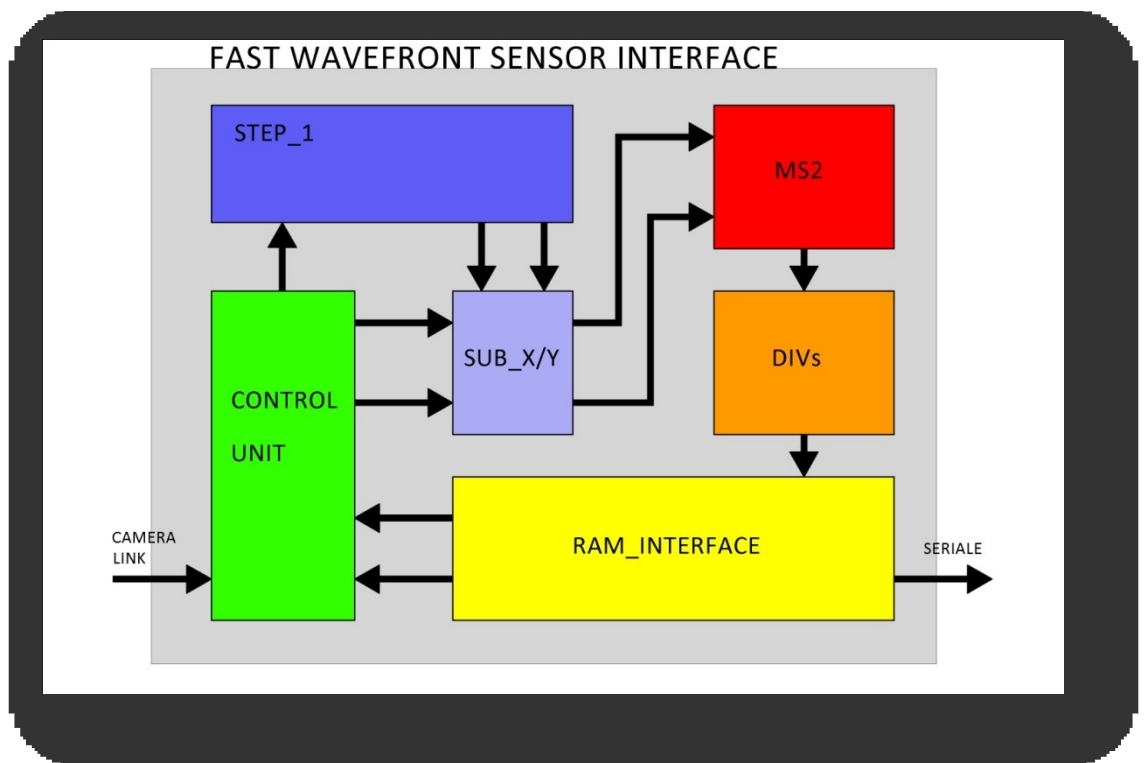


Figura 71 – Schema a blocchi TOP

6 Conclusioni

A livello prototipale la scelta dell'implementazione dell'algoritmo, per il calcolo dei centroidi, attraverso la realizzazione di una struttura a pipeline in un dispositivo FPGA è stata una scelta funzionale e performante.

L'idea di sfruttare una struttura dei dati in ingresso regolamentata da uno standard di trasmissione come lo standard Camera Link ha reso lo sviluppo del progetto molto flessibile ed adattabile alle più svariate modalità di funzionamento delle moderne camere, utilizzate per la realizzazione dei sensori Shack-Hartmann.

Per migliorare le prestazioni, in termini di area occupata dell'FPGA, si potrebbero apportare dei benefici sostituendo le numerose strutture a multiplexer/demultiplexer con ram dual port più compatte.

Le simulazioni effettuate con il tool ISim simulator di Xilinx hanno dimostrato il teorico funzionamento di ogni singolo step dell'implementazione dell'algoritmo. A livello di implementazioni post-sintesi l'effettiva realizzazione di un primo prototipo richiederà un debug più approfondito ed accurato con particolare attenzione nell'utilizzo dei casi limite.

Alla luce di quanto ottenuto il progetto è da apprezzare per la sua struttura particolarmente adattabile e performante. Potrà essere utilizzato come interfaccia di numerosi sensori differenti, non costituendo quindi una customizzazione in particolare.

Potrà analizzare in real-time immagini a frame rate elevati, sfruttando a pieno le potenzialità della Virtex5 utilizzata in questo progetto.

7 Bibliografia

[1] Andrew Rushton 1998, "*VHDL for Logic Synthesis*", Second Edition, John Wiley & Sons Ltd,

[2] Mark Zwolinski 2007, "*DIGITAL SYSTEM DESIGN WITH VHDL*", Pearson Education Limited,

[3] Robert K. Tyson 1948, "*Introduction to Adaptive Optics*", SPIE – The International Society for Optical Engineering,

[4] "*History and Principles of Shack – Hartmann Wavefront Sensing*", Ben C. Platt, Ph.D; Roland Shack, Ph.D,

[5] "*Historical Development of the Shack – Hartmann Wavefront Sensor*" Jim Schwiegerling, Ph.D. Department of Ophthalmology University of Arizona, Tucson Daniel R.Neal, Ph.D WaveFront Sciences, Inc., Albuquerque

[6] "*Analysis of optimal centroid estimation applied to Shack–Hartmann sensing*", Roy Irwan and Richard. G. Lane, 1999

[7] "*A center detection algorithm for Shack–Hartmann wavefront sensor*", Sung-Hoon Baik, Seung-Kyu Park, Cheol-Jung Kim, Byungheon Cha, 2005

[8] "*Measurement of moments for centroid estimation in Shack–Hartmann wavefront sensor—a wavelet-based approach and comparison with other methods*", P. Arulmozhivarman, L. Praveen Kumar, A.R. Ganesan, 2004

APPENDICE A

```

function Y=Fast_Sensor (Percent)
% MST_var_evolution (30)
N_lenti_X= 13;
N_lenti_Y= 14;

a= imread ('due_new.bmp');
% Coordinate inizio analisi
SOGLIA =50; %
SpV_X=0;
SpV_Y=0;
SV =0;
Valore=0;
Valore1=1;
X_pos=0;
Y_pos=0;
for x=128:1:1150
    X_pos= X_pos +1;
    for y=1 :1: 1023
        Y_pos=Y_pos +1 ;
        Valore=int32(a(y,x));
        if Valore > SOGLIA
            SpV_X =SpV_X + Valore1* X_pos;
            SpV_Y = SpV_Y + Valore1* Y_pos;
            SV = SV + Valore1;
        end
    end
    Y_pos =0;
end
Start_x= ceil (SpV_X/SV);
Start_y= ceil (SpV_Y/SV);

% STEP_1
Start_x1=128+Start_x-390; % coordinate prima scansione
Start_y1=Start_y-420; % coordinate prima scansione

% Prima Scansione di Assestamento Griglia di Riferimento
% Algoritmo Media Statistica con Soglia e controllo rumore
% Coordinate inizio analisi
Noise=30;
X_pos=0;
Y_pos=0;
Stot=0;
Valore=0;
S_X=0;
S_Y=0;
% Calcolo posizioni X_calc Y_calc
Ipos_x=1;
Ipos_y=0;
Icalc=1;
Spostamento_X=0;
I_Spostamento_X=0;
Spostamento_Y=0;
I_Spostamento_Y=0;

for M=0:1:13
    for N=0:1:12
        Max=0;
        for y=(M*60+Start_y1):1:(M*60+Start_y1+59)
            for x=(N*60+Start_x1):1:(N*60+Start_x1 +59)
                Valore=int32(a(y,x));
                if Valore > Max
                    Max= Valore;
                end
            end
        end
        Threshold= ceil (Max*Percent/100);

        for y=(M*60+Start_y1):1:(M*60+Start_y1+59)
            Ipos_y=Ipos_y +1;
            for x=(N*60+Start_x1):1:(N*60+Start_x1 +59)
                Valore=int32(a(y,x));

                if Threshold < Noise
                    Threshold = Noise;
                end
                if Valore < Threshold
                    Valore=0;
                    a(y,x)=0;
                end

                Stot=Stot+ Valore;

                S_X=S_X+ Valore*Ipos_x;
                Ipos_x=Ipos_x +1;

                S_Y=S_Y+ Valore*Ipos_y;
            end
        end
    end
end

```

```

        end
        Ipos_x=1;
    end

    X_calc_T(Icalc,1)=S_X/Stot;
    Y_calc_T(Icalc,1)=S_Y/Stot;

    if S_X > 0
        Spostamento_X = Spostamento_X+ (S_X/Stot);
        I_Spostamento_X = I_Spostamento_X +1;
    end

    if S_Y > 0
        Spostamento_Y = Spostamento_Y+ (S_Y/Stot);
        I_Spostamento_Y=I_Spostamento_Y+1;
    end

    % Resetto valori per iterazione successiva
    Icalc=Icalc+1;
    Stot=0;
    S_X=0;
    Ipos_x=1;
    S_Y=0;
    Ipos_y=1;
end
end
Spostamento_X= Spostamento_X/I_Spostamento_X;
Spostamento_Y= Spostamento_Y/I_Spostamento_Y;

DeltaSpostamento_X=0;
DeltaSpostamento_Y=0;

DeltaSpostamento_X= Spostamento_X -30;
DeltaSpostamento_Y= Spostamento_Y -30;

%Vettori correzione 1
Vett_CorrX1 = DeltaSpostamento_X;
Vett_CorrY1 = DeltaSpostamento_Y;

Start_x2 = Start_x1 +DeltaSpostamento_X;
Start_y2 = Start_y1 +DeltaSpostamento_Y;

% Seconda Scansione Assestamento Griglia di Rierimento
X_pos=0;
Y_pos=0;
Stot=0;
Valore=0;
S_X=0;
S_Y=0;
% Calcolo posizioni X_calc Y_calc
Ipos_x=1;
Ipos_y=0;
Icalc=1;
Spostamento_X=0;
I_Spostamento_X=0;
Spostamento_Y=0;
I_Spostamento_Y=0;

INDICE_1=0;
INDICE_2=0;
INDICE_3=0;
INDICE_4=0;

for M=0:1:13
    for N=0:1:12
        Max=0;
        for y=(M*60+Start_y2):1:(M*60+Start_y2+59)
            for x=(N*60+Start_x2):1:(N*60+Start_x2 +59)
                Valore=int32(a(y,x));
                if Valore > Max
                    Max= Valore;
                end
            end
        end
        Threshold= ceil(Max*Percent/100);
        for y=(M*60+Start_y2):1:(M*60+Start_y2+59)
            Ipos_y=Ipos_y +1;
            for x=(N*60+Start_x2):1:(N*60+Start_x2 +59)
                Valore=int32(a(y,x));

                if Threshold < Noise
                    Threshold = Noise;
                end
                if Valore < Threshold
                    Valore=0;
                    a(y,x)=0;
                end

                Stot=Stot+ Valore;

                S_X=S_X+ Valore*Ipos_x;
                Ipos_x=Ipos_x +1;
            end
        end
    end
end

```

```

        S_Y=S_Y+ Valore*Ipos_y;

    end
    Ipos_x=1;
end

X_calc_T(Icalc,1)=S_X/Stot;
Y_calc_T(Icalc,1)=S_Y/Stot;

if S_X > 0
    Spostamento_X = Spostamento_X+ (S_X/Stot);
    I_Spostamento_X = I_Spostamento_X +1;
end
if M == 0
    if S_X > 0
        INDICE_1 =INDICE_1+1;
    end
end
if M == 13
    if S_X > 0
        INDICE_2 =INDICE_2+1;
    end
end
if N == 0
    if S_X > 0
        INDICE_3 =INDICE_3+1;
    end
end
if N==12
    if S_X > 0
        INDICE_4 =INDICE_4+1;
    end
end

if S_Y > 0
    Spostamento_Y = Spostamento_Y+ (S_Y/Stot);
    I_Spostamento_Y=I_Spostamento_Y+1;
end

% Resetto valori per iterazione successiva
Icalc=Icalc+1;
Stot=0;
S_X=0;
Ipos_x=1;
S_Y=0;
Ipos_y=1;
end
end
Spostamento_X= Spostamento_X/I_Spostamento_X;
Spostamento_Y= Spostamento_Y/I_Spostamento_Y;

DeltaSpostamento_X=0;
DeltaSpostamento_Y=0;

DeltaSpostamento_X= Spostamento_X -30;
DeltaSpostamento_Y= Spostamento_Y -30;

%Vettori correzione 2
Vett_CorrX2 = DeltaSpostamento_X;
Vett_CorrY2 = DeltaSpostamento_Y;

Start_x3 = Start_x2 +DeltaSpostamento_X;
Start_y3 = Start_y2 +DeltaSpostamento_Y;

if INDICE_1 == 0
    Start_y4= Start_y3 +60;
end
if INDICE_2 == 0
    Start_y4= Start_y3 -60;
end
if INDICE_3 == 0
    Start_x4= Start_x3 +60;
end
if INDICE_4 == 0
    Start_x4= Start_x3 -60;
end

%
for i=0:1:(N_lenti_Y-1)
    for ii= 1:1:N_lenti_X
        p= ii+i*N_lenti_X;
        X_ref4(p,1)= Start_x4 -1 + 30 +60*(ii-1);
    end
end
for i=0:1:(N_lenti_Y-1)
    for ii= 1:1:N_lenti_X
        p= ii+i*N_lenti_X;
        Y_ref4(p,1)= Start_y4 -1 + 30 +60*(i);
    end
end
end

```



```

% Algoritmo Media Statistica con Soglia e controllo rumore
% Coordinate inizio analisi
X_pos=0;
Y_pos=0;
Stot=0;
Valore=0;
S_X=0;
S_Y=0;
% Calcolo posizioni X_calc
Ipos_x=1;
Icalc=1;
for M=0:1:13
    for N=0:1:12
        Max=0;
        for y=(M*60+Start_y4):1:(M*60+Start_y4+59)
            for x=(N*60+Start_x4):1:(N*60+Start_x4 +59)
                Valore=int32(a(y,x));
                if Valore > Max
                    Max= Valore;
                end
            end
        end
        Threshold= ceil(Max*Percent/100);
        for y=(M*60+Start_y4):1:(M*60+Start_y4+59)
            for x=(N*60+Start_x4):1:(N*60+Start_x4 +59)
                Valore=int32(a(y,x));

                if Threshold < Noise
                    Threshold = Noise;
                end
                if Valore < Threshold
                    Valore=0;
                    a(y,x)=0;
                end

                Stot=Stot+ Valore;
                S_X=S_X+ Valore*Ipos_x;
                Ipos_x=Ipos_x +1;
            end
            Ipos_x=1;
        end

        if S_X > 0
            X_calc_T1(Icalc,1)=S_X/Stot;
        else
            X_calc_T1(Icalc,1)=30;
        end
        % Resetto valori per iterazione successiva
        Icalc=Icalc+1;
        Stot=0;
        S_X=0;
        Ipos_x=1;
    end
end

% Calcolo posizioni Y_calc
X_pos=0;
Y_pos=0;
Stot=0;
Valore=0;
S_X=0;
S_Y=0;
Ipos_y=1;
Icalc=1;
for M=0:1:13
    for N=0:1:12
        Max=0;
        for x=(N*60+Start_x4):1:(N*60+Start_x4 +59)
            for y=(M*60+Start_y4):1:(M*60+Start_y4+59)
                Valore=int32(a(y,x));
                if Valore > Max
                    Max= Valore;
                end
            end
        end
        Threshold= ceil(Max*Percent/100);
        for x=(N*60+Start_x4):1:(N*60+Start_x4 +59)
            for y=(M*60+Start_y4):1:(M*60+Start_y4+59)
                Valore=int32(a(y,x));

                if Threshold < Noise
                    Threshold = Noise;
                end
                if Valore < Threshold
                    Valore=0;
                    a(y,x)=0;
                end
            end
        end
    end
end

```

```

        Stot=Stot+ Valore;
        S_Y=S_Y+ Valore*Ipos_y;
        Ipos_y=Ipos_y +1;
    end
    Ipos_y=1;
end

if S_Y > 0
    Y_calc_T1(Icalc,1)=S_Y/Stot;
else
    Y_calc_T1(Icalc,1)=30;
end

% Resetto valori per iterazione successiva
Icalc=Icalc+1;
Stot=0;
S_Y=0;
Ipos_y=1;

end

end

% Calcolo Stringa di valori assoluti per graficare i punti
for i=0:1:13
    for ii= 1:1:13
        p= ii+i*13;
        X_calc_T1(p,1)= Start_x4 -1 + X_calc_T1(p,1) +60*(ii-1);

    end

end

for i=0:1:13
    for ii= 1:1:13
        p= ii+i*13;
        Y_calc_T1(p,1)= Start_y4 -1 + Y_calc_T1(p,1) +60*(i);

    end

end

end

% Plot immagine
imagesc (a);
hold on;
plot(Start_x1, Start_y1, 'b+');
hold on;
plot(Start_x2, Start_y2, 'm+');
hold on;
plot(Start_x3, Start_y3, 'w+');
hold on;
plot(Start_x4, Start_y4, 'rp');

hold on;
scatter (X_ref4,Y_ref4,'x*');
hold on;
scatter (X_calc_T1,Y_calc_T1, 'y+');

h = legend('STEP_1','STEP_2','STEP_3','STEP_correction','Centri_Griglia di Riferimento','Centroidi',3);
set(h,'Interpreter','none','Location','SouthWest','Color',[0 1 0])

Y=[Start_x1,Start_y1;
    Start_x2,Start_y2;
    Start_x3, Start_y3];

end

```

Ringraziamenti

Si ringraziano per il supporto ottenuto, durante lo svolgimento di questa tesi; la Prof.ssa Pelizzo, l'Ing. Bonora e tutti i componenti del team di Adaptica.

Un ringraziamento speciale alla mia famiglia, senza la quale non sarei arrivato a raggiungere questo importante traguardo.

Ed infine un doveroso ringraziamento ad Alberto, il quale durante questi faticosi anni di pendolarismo, ha accumulato circa 66000Km (stima sottodimensionata) tra Actv e Trenitalia.

Grazie a tutti!!