POLITECNICO DI TORINO

Corso di Laurea
in Ingegneria Elettronica

Tesi di Laurea

Sistema di acquisizione di eventi AER basato su Ethernet

Relatore/i
prof. Danilo Demarchi
dott. Paolo Motto Ros

firma del relatore (dei relatori)
……

……

Candidato
Michele Biscione

firma del candidato
……

A.A. 2017/2018

Indice generale

1 Introduzione

1.1 Introduzione al progetto

Il protocollo AER che andremo ad analizzare in questa tesi è molto utile in vari ambiti, in quanto replica sostanzialmente il funzionamento di un cervello umano:infatti quando si tocca un oggetto, viene generato qualcosa di molto simile a un segnale elettrico, o come viene chiamato dagli scienziati “un evento” o una serie di “eventi”, quindi, questi tipi di segnali possono essere usati, chiaramente, con dei sistemi adatti, in quegli scenari dove ci deve essere uno scambio molto veloce delle informazioni; ecco perché, uno degli ambiti in cui questa tecnologia calza meglio, è quello dei robot.
Nel mondo odierno i robot fanno ormai parte dell’industria per svariati compiti, soprattutto per alcuni lavori pesanti e pericolosi nelle fabbriche (ad esempio spostare pezzi pesanti, lavorare con vernici o acidi, alte temperature ecc). Ultimamente, stanno iniziando a essere progettati anche per l’utilizzo a supporto dell’uomo, per compiti semplici e specifici, ma di grande utilità (portare referti ospedalieri, robot domestici pulitori, animali o semplicemente in ambito ludico e ricreativo (con abilità musicali, per l’utilizzo nel cinema, animatronics, cioè robot animali, nei parchi di divertimenti). I primi che ho citato, i cosiddetti robot umanoidi, vengono realizzati anche per analizzare e studiare il comportamento delle persone.
Di robot umanoidi ne esistono vari tipi, alcuni anche molto sofisticati, come il francese Innorobo (fig. 1), ASIMO della Honda (fig. 2) o l’italiano Icub (fig. 3).

Figura 1: Robot francese NAO della Aldebaran Robotics (I)

Figura 2:robot giapponese ASIMO della Honda (II).

Figura 3 Robot italiano ICub dell'Istituto Italiano di Tecnologia (IIT) (III)

Negli ultimi anni, i robot open source (IV), vengono sviluppati, seguendo due filoni differenti:la parte software (i sistemi operativi che li gestiscono), come Orocos, OpenRTM e ROS e le parte hardware (componenti meccaniche), per permettere i movimenti degli automi. ICub, comunque è un progetto finanziato in ambito europeo, con una piattaforma comune, per realizzare un sistema artificiale di tipo cognitivo.
L'idea dei creatori, in sostanza, è quella di creare un robot alla portata di tutti e per questo, tutti i piani e progetti, design, sono a licenza libera (GPL, FDL, LGPL).
Icub usa un sistema operativo compilabile in diversi ambienti, come Linux, Windows e MacOs, i più usati attualmente, oltre ovviamente, a tutta una serie di tools dedicati.
Purtroppo, i disegni dei componenti elettronici e meccanici (2D e 3D), non possono essere riprodotti, in quanto per la progettazione i progettisti hanno dovuto usare forzatamente tools di tipo CAD, in quanto non esiste un'alternativa gratuita.
Come detto in precedenza, e sempre nell'ottica di fornire tutta la documentazione utile per la realizzazione in proprio di questo automa e in base a quanto indicato, poco prima, hanno deciso di rilasciare tutti i file CAD realizzati, su licenza libera.
L'idea di base era quella di replicare un bambino di circa tre anni, con semplici capacità manipolatorie e di di gattonare, mentre quello attuale è alto 1 metro, pesa 24 kg e ha 53 gradi di libertà, l'età apparente, con queste misure, è di circa 8 anni.
Per dargli un' aspetto il più vicino possibile alla realtà, compresi i movimenti, il gruppo di ricerca si è servito di modelli dati biomedici (per cinematica) e tabelle antropomorfe (per dare le giuste proporzioni delle parti del corpo);sono state realizzate anche simulazioni di torsione per ottenere movimenti corretti e quasi reali.
Il movimento di cui accennavamo prima, è permesso da motori frameless Kollmorgen-DanaherMotion RBE di tipo brushless, CSD frameless ed Harmonic Drive flat speed reducer.
Come occhi, il robot utilizza due videocamere all'interno degli stessi, collegate a un meccanismo, che permette ad essi di muoversi, focalizzandosi sul soggetto da osservare, le espressioni facciali sono visualizzabili tramite appositi LED che le riproducono.
Come elettronica e sensori, utilizza una scheda PC104 (fig. 4), un computer in miniatura, basata su un processore Intel Core 2 Duo Pentium da 2, 16 Mhz, con un 1GB di RAM, con elettronica di controllo e sensori di acquisizione.

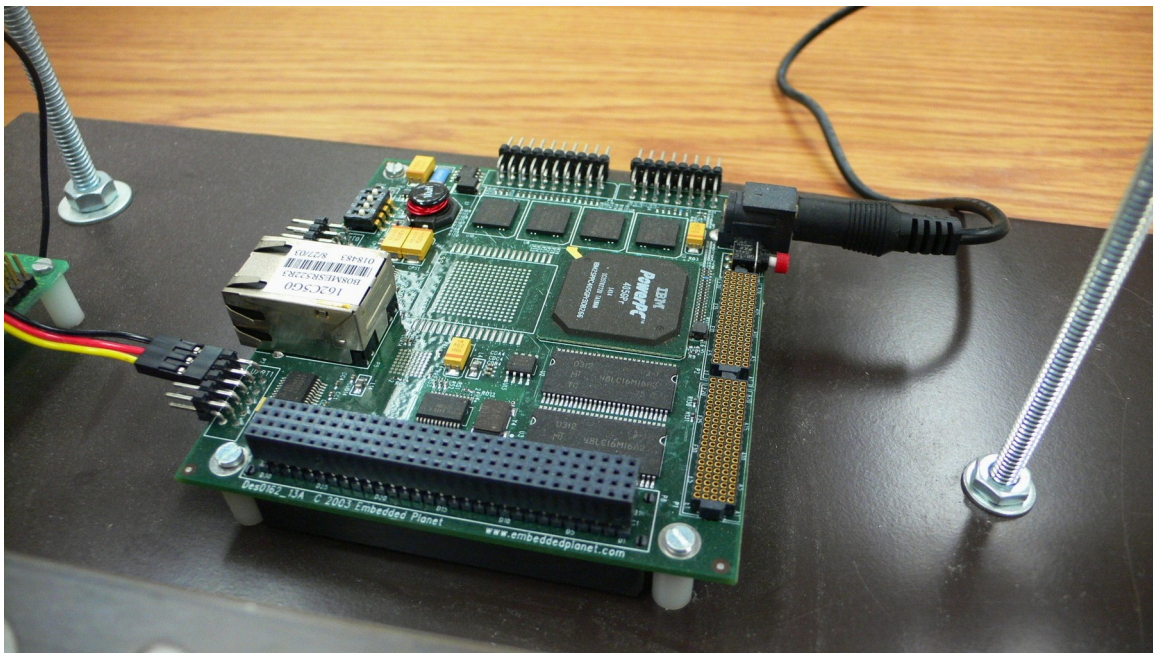

Figura 4 scheda PC-104 (V)

Questo sistema trasmette/riceve tutti i dati, verso le varie parti del robot, tramite dieci porte CAN bus.
Il robot è provvisto anche di sensori tattili, tipo pelle.
Per la comunicazione, si utilizzano come detto vari bus di tipo CAN, con velocità di trasmissione di 1 Mbit/s. Sulla scheda PC 104 converge tutta la rete con topologia a stella e all'interno dei bus, ci sono dei sottobus, in numero sufficiente da raggiungere tutte le parti funzionali del robot, anch'esse facenti capo al Pc-104.
Anche se questo ICub è ormai completo e funzionante, da ormai alcuni anni, il progetto non è ancora terminato, infatti, si è gia iniziato da tempo, lo sviluppo del suo successore, il "fratellino" ICub2, giunto quasi in via di conclusione.
Tra i miglioramenti apportati, troviamo il rilevamento della coppia articolare, in modo da interagire con gli umani senza arrecare loro danni fisici, infatti tutto ciò è reso possibile da sensori con coppie piezoresistive con semiconduttori di tipo SSG, con un sistema che controlla la torsione, in modo da applicare la giusta forza (ad esempio la semplice stretta di mano con una persona). Segue il miglioramento del feedback tattile, sempre nell'ottica dell'interazione sicura con persone, oggetti e ambiente. Infatti ciò è possibile grazie all'utilizzo di sensori di pressione, di tipo tattile, collocati nelle apposite zone, come ad esempio, nella zona delle impronte digitali, il torso, le ginocchia, le braccia, gli avambracci, eccetera, per un totale di 4200 sensori, maggior numero al mondo su un robot di questo tipo, quindi saranno ora in grande quantità, i dati da processare.
Oltre a ciò, sono stati rivisti i design di testa, per risolvere alcuni problemi, come il rapido surriscaldamento del collo e proposta una soluzione con un attuatore di tipo parallelo, e di occhi, dove vi era un problema di backslash nella trasmissione, risolto con un riduttore di velocità di tipo Harmonic Drive. Infine la ritaratura della cinghia dentata, per migliore il posizionamento delle telecamere, ancora, migliorati gli encoder ottici, usando motori brushless con standard PWM trapezoidali, la control board invece è stata rivista, ed infine, la profonda revisione del bus Ethernet, a causa di tutte le migliorie del progetto e di conseguenza, aumento del traffico dati, con diminuzione del throughput. Si utilizzerà un livello intermedio con una board di controllo denominata DEV e un buon protocollo di trasmissione di tipo CAN, connesso a un bus Ethernet tramite un cavo CAT5, in modo punto punto o daisy chain.
In questo progetto, comunque è stato utilizzato un sistema di trasmissione a eventi, come il protocollo AER, oggetto di questa tesi, un'evoluzione del classico protocollo Ethernet;quest'ultimo è largamente usato praticamente in tutti i computer connessi ad internet ed ultimamente anche nelle console di videogames, nella domotica, sia in modalità a filo che wireless, nella sua ultima specifica.
L'oggetto di questa tesi, appunto, riguarda lo sviluppo di un nuovo tipo di protocollo Ethernet basato su sistema AER per acquisizione e generazione di eventi, per essere utilizzato in moderne workstation.

2 Descrizione delle tecnologie utilizzate

2.1 Descrizione Ethernet

Lo standard Ethernet (VI), è stato creato dall'IEEE e vengono utilizzate la versione 802. 3 e revisioni successive, nei collegamenti a internet via cavo.
Queste revisioni, indicano semplicemente, la velocità di connessione, 10 Mbit/s per 802. 3 e così via, fino alle ultime, che si attestano oltre il Gigabit;sono indicate tutte come LAN, cioè Ethernet Local Area Network, altre specifiche sono la Common Media Access e il Control (MAC) Management Information Base (MIB) sono presenti.
La parte denominata PHY (Phisical Layer Entity), che si occupa di inviare/ricevere pacchetti (dopo averli codificati/decodificati), con un'opportuna frequenza, in base alla velocità selezionata, può essere scelta tramite il MIIs (Media Independent Interface), un'interfaccia opzionale, leggermente diversa;in aggiunta ci sono anche la fornitura di alimentazione sui doppini del PHY e la gestione e il controllo dei protocolli.
Per dirigere il traffico di dati si usa il metodo di accesso CSMA/CD, di tipo half duplex:in sostanza lo stesso mezzo trasmissivo è condiviso da tutte le stazioni (da una coppia a tre, quattro. . . .) e solo una e una soltanto, può trasmettere a un'altra, quindi, in caso di collisione, dopo del tempo, può ritrasmettere, attendendo ogni volta, prima di ritentare la trasmissione, permettendo così il movimento della collisione nel sistema.
E' possibile anche utilizzare il metodo full duplex, che permette trasmissioni contemporanee tra coppie di stazioni, senza collisione, attese o controlli dell'uso del mezzo;ciò è possibile se le stazioni sono configurate in questo'ultima modalità.
Nella pila Iso/Osi, Ethernet è posto ai livelli 1 (fisico) e nel sottolivello 2 (MAC), del livello 2 (collegamento).
Il pacchetto è mostrato nella figura successiva:

Figura 5:formato del pacchetto

E' composto da Preambolo, uno Start Frame Delimiter (SFD), gli indirizzi di frame sorgente e destinatario, lunghezza/tipo di protocollo del campo seguente, che contiene il MAC del dato del client, un campo con il padding, se richiesto e il Frame Check Sequence (FCS), contente un cyclic redundancy check, per verificare errori di frame MAC ricevuto, infine, un campo aggiuntivo, extension, (solo per operazioni a 1 Gb/s) se si necessita.
Tutti i campi, hanno dimensione fissa, tranne MAC Client Data, Pad e extension.
Per quanto riguarda AER invece, il nome è l'acronimo che sta ad indicare Address-Event-Representation (VII), un protocollo che si posiziona a un livello bassissimo nella pila iso-osi, praticamente al livello 1 (fisico), simile alla connessione usb (si pensi appunto della velocità di trasmissione dati di hard - disk-pendrive ecc), rapido, quindi molto adatto allo scopo di cui verrà descritto in maniera più chiara in seguito.
Questo sistema si adatta in particolare a quelle applicazioni che necessitano bassi consumi di trasmissione dati, per esempio nell'integrazione di un piccolo robot, per trasmissione segnali relativi al movimento, in campo biomedico, ecc.

2.2 Descrizione protocollo Address-Event Representation

Il protocollo AER, proposto da Mahwald e Silivotti, serve ad inviare picchi o impulsi da una locazione di array di neuroni su un primo chip a un'altro, inviandolo come unico indirizzo binario, come nella figura seguente.

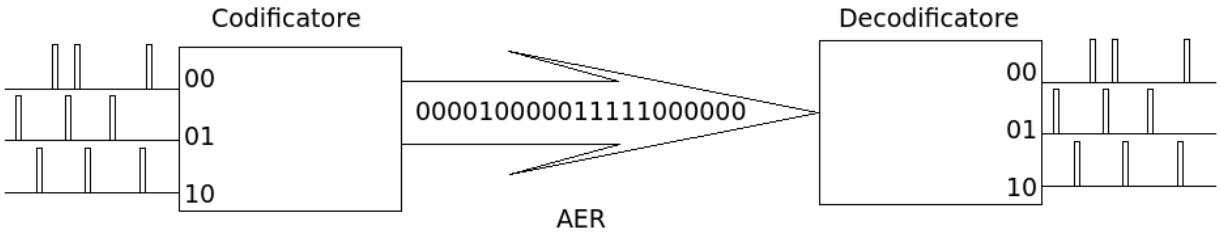

Figura 6. Trasferimento dati protocollo AER.

così, quando viene ricevuto il segnale, viene selezionato il corretto indirizzo destinatario da un address decoder.
L'idea di questo protocollo è nata alcuni anni fa, ha subito vari miglioramenti e si è distinto come il più adatto per essere utilizzato nelle trasmissioni neuromorfiche. ad esempio, i circuiti arbitrari gerarchici, sono stati sviluppati, prima come array 1-D, poi in 2-D.
Il primo supporta varie trasmissioni simultanee, una prova è stata realizzata con successo in laboratorio con 20000 neuroni e 120000 connessioni AER, come anche un sistema con miliardi di connessioni, ma quest'ultimo, solo sulla carta;in futuro, si pensa a un sistema neuromorfico con miliardi di connessioni e fili virtuali riprogrammabili.
L'accuratezza del sistema neuromorfico, ha otto impulsi di larghezza fissata, proprio come il sistema nervoso umano, in modo tale che le informazioni vengano codificate in maniera corretta, tramite dispersione temporale e latenza, misuriamo l'intervallo di precisione, l'intervallo temporale tra il segnale iniziale e lo spiking e la "Latenza neuronale", la "Dispersione neuronale temporale", invece, dipende dai singoli neuroni.
L'aggiunta di offsets sistematici e stocastici, agli spikes times a causa della latenza temporale finita e dispersione temporale, avviene quando viene inviata una comunicazione, per trovare la stazione destinataria, neuroni con spikes o identità;in questo modo, la precisione diminuisce.
L'organizzazione degli array di tipo 2D, è stata introdotta, in quanto era necessario leggere i treni di spikes;infatti si è optato una retina siliconica o cochleare, eliminare la ridondanza e codificare meglio le informazioni, si è provveduto a una modellazione tramite preprocesso, ed anche una miglior design dei canali, per il movimento delle unità neuronali, con miglioramenti, come implementazione ed efficienza, in aggiunta anche il supporto a quantizzazione di pixels paralleli.
Dato che la codifica delle informazioni è molto efficiente, questo protocollo, si adatta appunto a processori neuromorfici di alto livello.
Un esempio di utilizzo può essere per la retina:i treni di impulso che ottiene, derivano dalle immagini che capta (luce e oggetti, spazio), che vengono convertite;la quantità di dati necessari alla trasmissione è molto elevata, si parla di circa 40 GB/s, a 17 bits, 10 Hz, 2x90°x90° come campo visivo;con la codifica per spike, i vari assiomi di nervi, trasmessi sono un migliaio di volte meno, circa 40 Mb/s.
Durante lo sviluppo, queste retine hanno subito notevoli miglioramenti, come ad esempio, come il controllo automatico del guadagno nel fotoricettore, che dipende dal contrasto, non più dalla luce, miglior range di uscita, il filtraggio spaziotemporale della banda passante, eliminando ridondanza (sopprime basse frequenze) e riducendo rumori (esclude alte frequenze), filtraggio spaziale e passa

alto, semi rettificazione dell'onda, mantenimento della risposta fasica transiente, architettura foveata.
Nella realizzazione di un canale trasmissivo, il giusto bilancio, si ottiene prestando cura ad alcuni parametri molto importanti:la latenza, la mediana della distribuzione tra spikes trasmessi/ricevuti, l'integrità, cioè le parti di spikes correttamente inviate, dispersione standard, deviazione di distribuzione della latenza ed infine, la capacità, spikes trasmessi intesi come tasso massimo.
Il throughput è cio che si ottiene da questi parametri;in base alla strategia utilizzata, le capacità del canale possono variare;alcune alternative sono:integrità arbitraria, dispersione con queuing, latenza guidata da eventi, capacità pipeline, codifica in base al tempo.
Di relativa importanza, rivestono le connessione punto-punto tra spikes di neuroni su diversi chips. Dopo i primi problemi di crosstalk e legati alle temporizzazioni e grazie ai progressi ottenuti nei sistemi digitali asincroni VLSI, i ricercatori del gruppo di Martin alla Caltech hanno realizzato un microprocessore funzionante senza clock.
I difetti iniziali, sono stati eliminati con l'utilizzo di canali arbitrari e di un sistema MAD (Mixed analog digital). Con questo tipo di sintesi è ora possibile fare il design di un circuito VLSI con specifiche di alto livello, scritte in CHP (commmunication hardware process), seguendo alcune regole produttive o production rule set (PRS), per utilizzare in tempo reale le espressioni booleane.
Il processo di sintesi, prevede decomposizione del programma (dove possiamo modificare il design in base alla necessità hardware e espansione handshaking) le porte possono essere impostate in modo da utilizzarne alcune in modalità attiva, altre passiva, connesse in modo da creare un canale, così possiamo attivare quelle che ci servono.
Vi è un'operazione primitiva chiamata S, che serve a sapere se il canale è occupato e se è impostato in forma passiva, si avrà perciò TRUE o FALSE, per questo parametro.
Tramite l'HSE o hand-shaking expansion, potremo scegliere la porta da attivare o meno o rinegoziare la comunicazione, spingendo, volta per volta i dati, in modo tale da aumentare la velocità e ridurre il consumo di memoria.
Vediamo come funziona la trasmissione e la forma del pacchetto in dettaglio, rappresentati nella figura seguente:

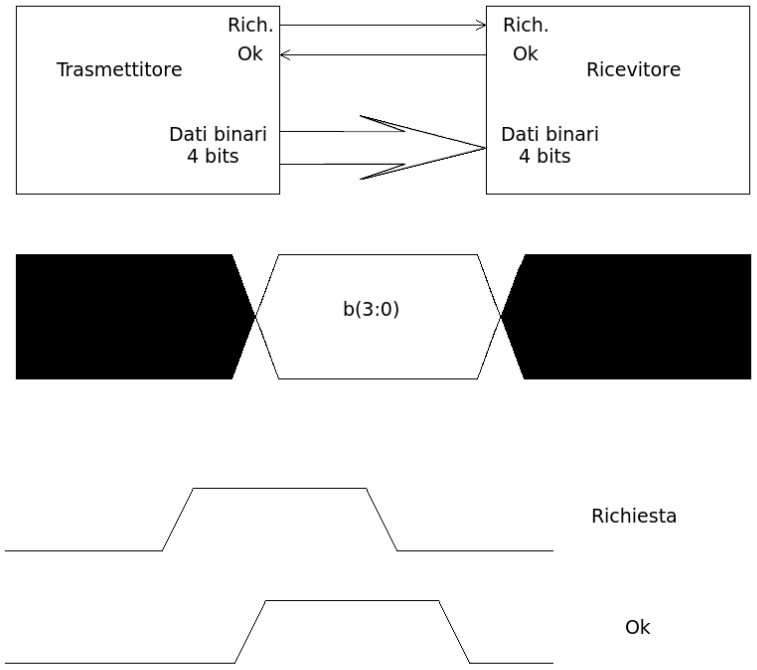

Figura 7: trasmissione tra due stazioni AER e rappresentazione pacchetto

Quando una stazione deve trasmettere, ad esempio la stazione a sinistra, nell'immagine precedente, invia una richiesta tramite un' hand-shake qui rappresentata con il nome "richiesta", la porta è impostata in modalità passiva, controllando quest'ultimo segnale. Una volta giunto destinazione, vi sarà un'attesa, pari al tempo di ritorno della richiesta, una sorta di "ok" in pratica, fin'ora, siamo nella parte nera, nel diagramma di trasmissione.
Dopo questa fase, il pacchetto può essere inviato senza ulteriori controlli, come indicato in figura, nella parte bianca;Conclusa la trasmissione, il diagramma torna nero, indicando l'assenza di trasmissioni, compresi invii di segnali di controllo, che in questo protocollo non esistono.

2.3 Confronto Ethernet- AER

Paragonandoli, la prima cosa da specificare, è il fatto Ethernet è un insieme di tecnologie per reti locali, per connettere computer ad internet, compreso il protocollo di trasmissione, invece AER è un protocollo di trasmissione a se. Concettualmente, come protocollo trasmissivo, appunto, sono molto simili, infatti entrambi trasmettono pacchetti, possono essere selezionate velocità diverse, ci sono dei segnali per verificare alcuni paramatri, ma la differenza sta proprio nella semplicità del protocollo, infatti, mentre Ethernet invia segnali durante le fasi di idle, consumando risorse ed energia, AER invece, rimane in uno stato sospeso, senza inviare nulla. .
Quando è necessaria una trasmissione, quest'ultimo, invia solo e soltanto il dato, in cui la prima parte, indica l'inizio della trasmissione, quindi, evita di appesantire il sistema e lo semplifica, infatti non servono tutti i vari segnali di cui dispone Ethernet.
Infatti, anche per il fatto che quest'ultimo, viene usato anche per instradare pacchetti esternamente alla semplice connessione punto-punto, di conseguenza ha bisogno anche di altri dettagli, per permettere la trasmissione, come la definizione di primitive, per specificare il tipo di comunicazione, come request, per richiedere di inizializzare il servizio, poi indication, per mandare il pacchetto a livello N-1, poi, indicare il physical layer.
Il pacchetto trasmesso con AER invece è molto più semplice e lo si evince, osservando le due figure;infatti è presente solamente il dato da inviare.

Preambolo	SFD	Destination MAC Address	Source Address	Lunghezza o tipo	MAC client data	PAD	FCS	Extension

Pacchetto Ethernet

Pacchetto AER

Figure 5 e 7

La scelta di utilizzare la tecnologia Ethernet, per questo tipo di progetto, deriva dal semplice fatto che non esiste un'interfaccia AER fisica, una scheda di rete apposita ed anche per il fatto che l'altro sistema è presente praticamente in tutte le schede di rete dei pc esistenti.
Quindi useremo una normale scheda di rete, conforme a questo standard, per veicolare i nostri segnali, dopo averli opportunamente impostati, anche in base all'interfaccia in uscita dall'FPGA da quest'ultimo, in base al numero di cavi/connessioni presenti sul dispositivo.

2.4 Descrizione del progetto

Come indicato nell'introduzione, per il progetto, era necessario creare un sistema di generazione eventi, per ricostruire il segnale inviato, con un sistema custom usb-ethernet. Per fare ciò, ci si è serviti di un kit di sviluppo della Digilent, denominato Arty (VII), che si può osservare in figura 8. E' un dispositivo che può essere utilizzato per svariate applicazioni di sviluppo, essendo basato su un chip FPGA (Field Program Gate Array). Tra le sue caratteristiche peculiari, la semplicità di interfacciamento al pc tramite cavo USB, dove vengono inviati e ricevuti i dati senza bisogno di alimentazione esterna (fornita dal cavo stesso) oppure con un connettore di alimentazione. Alternativamente i dati sono trasferibili tramite un cavo LAN, ma di tipo CROSS, perchè la connessione è di tipo punto punto.

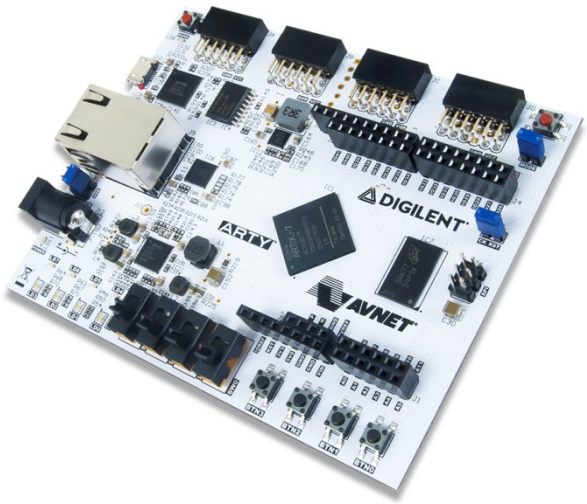

Figura 8: Digilent Arty. (IX)

Il chip FPGA può essere riprogrammato per consentirgli l'esecuzione di design ed operazioni diverse, in base anche alle necessità dei progetti da sviluppare.
Come linguaggio di sviluppo è stato usato il Verilog, il software usato per sintesi e simulazione è Vivado 2017. 1. Ci siamo anche serviti di Modelsim 6, 4ª per l'esecuzione dei test-bench.
Il lavoro si è svolto seguendo questo flusso:

1)Test del softcore Ethernet, scaricato dal sito opencores, con licenza libera.
2)Realizzazione del data generator.
3)Test ricezione lato pc tramite software Wireshark, un software sniffer per connessioni di rete.

La prima parte del lavoro, consisteva nel testare il core, in particolare un sistema di connessione dati, scaricato da Opencore (X), sito in cui si possono trovare vari IP, utilizzabili con licenza libera.
Si tratta di un’architettura Ethernet tri-mode, cioè funzionante alle velocità 10, 100, 1000 Mb/s, ;ono state rispettate le specifiche 802. 3 per il controller MAC al suo interno, ed ha 2000 Lcs/Les per tutte le funzioni per cui è stato realizzato.
Supporta anche la trasmissione half-duplex per 10 e 100 Mb/s, dispone di interfacce FIFO, ha registri configurabili, può ricevere pacchetti broadcast e dispone anche di un contatore RMON MIB.

3 Esecuzione sintesi ed elaborazione dati

3.1 Test preliminare del core fornito e creazione del progetto iniziale.

Dato che il core, così come è stato progettato, conteneva un'architettura molto estesa, comprese diverse funzionalità per molteplici applicazioni, sono state estratte le parti che occorrevano ai fini del progetto. Sono state mantenute la parte relativa alla connessione Ethernet e i file di test (questi ultimi sono stati poi adattati per conformarsi alle nostre necessità).
Di conseguenza, a partire da questa frazione del progetto originale è stata creata una nuova entità principale, AER_to_eth, visibile in figura 9, con all'interno: il modulo MAC opportunamente modificato per trasmettere dati collezionati con il protocollo AER; un modulo di configurazione, Host-sim, basato su una macchina a stati finiti che è stato inserito per programmare ad-hoc alcuni parametri memorizzati nei registri interni, il valore di questi nella versione originale era impostato tramite un sistema automatico prima di sintetizzare il circuito.
Segue il generatore di dati, Data Generator, sempre in figura 9, unità fondamentale del test, in quanto questa è l'entità che genera gli eventi che poi verranno visualizzati con l'utilizzo dello sniffer. I pacchetti, mimando un ricevitore AER sono generati con un'ulteriore macchina a stati finiti, appositamente realizzata in modo da rispettare le tempistiche di comunicazione tra le diverse parti del circuito.
Le differenti frequenze di clock necessarie per far operare tutte le parti del circuito sono ottenute dal clock principale dell' FPGA utilizzando dei moduli PLL generati tramite il wizard, presente nel software Vivado. In dettaglio, ne sono stati creati quattro: da 125 Mhz per l'interfaccia GMII/MII del MAC, da 66 Mhz, utilizzato dal data generator e dal MAC, un'altro da 80 Mhz per il circuito di configurazione ed anch'esso per il MAC ed infine un'altro ancora da 25 Mhz, utilizzato dall'interfaccia fisica esterna, nella fattispecie, il chip TI DP83848J, rappresentata nella parte destra della figura, al di fuori dell'entità principale.
Concludono lo schema, la presenza di un led per la generazione dei segnali visivi che ci consente di verificare l'effettiva velocità trasmissione e un tasto di reset, collegato a VDD, utilizzato per riavviare l'intero circuito e, conseguentemente l'invio dei dati.

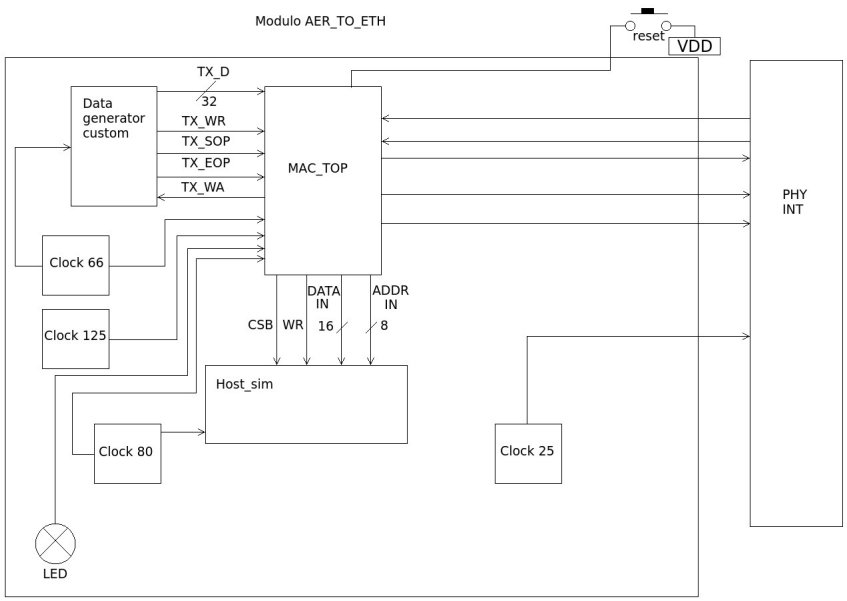

Figura 9: Schema a blocchi della Top-entity.

Come verifica per la corretta creazione della frequenza di clock centrale, l'uscita del modulo Clock 125 è stata verificata con l'oscilloscopio, come mostra l'immagine successiva, estratta dallo stesso, indicante, in basso a destra, la frequenza appunto di circa 125 Mhz richiesta dalla documentazione del core originale.

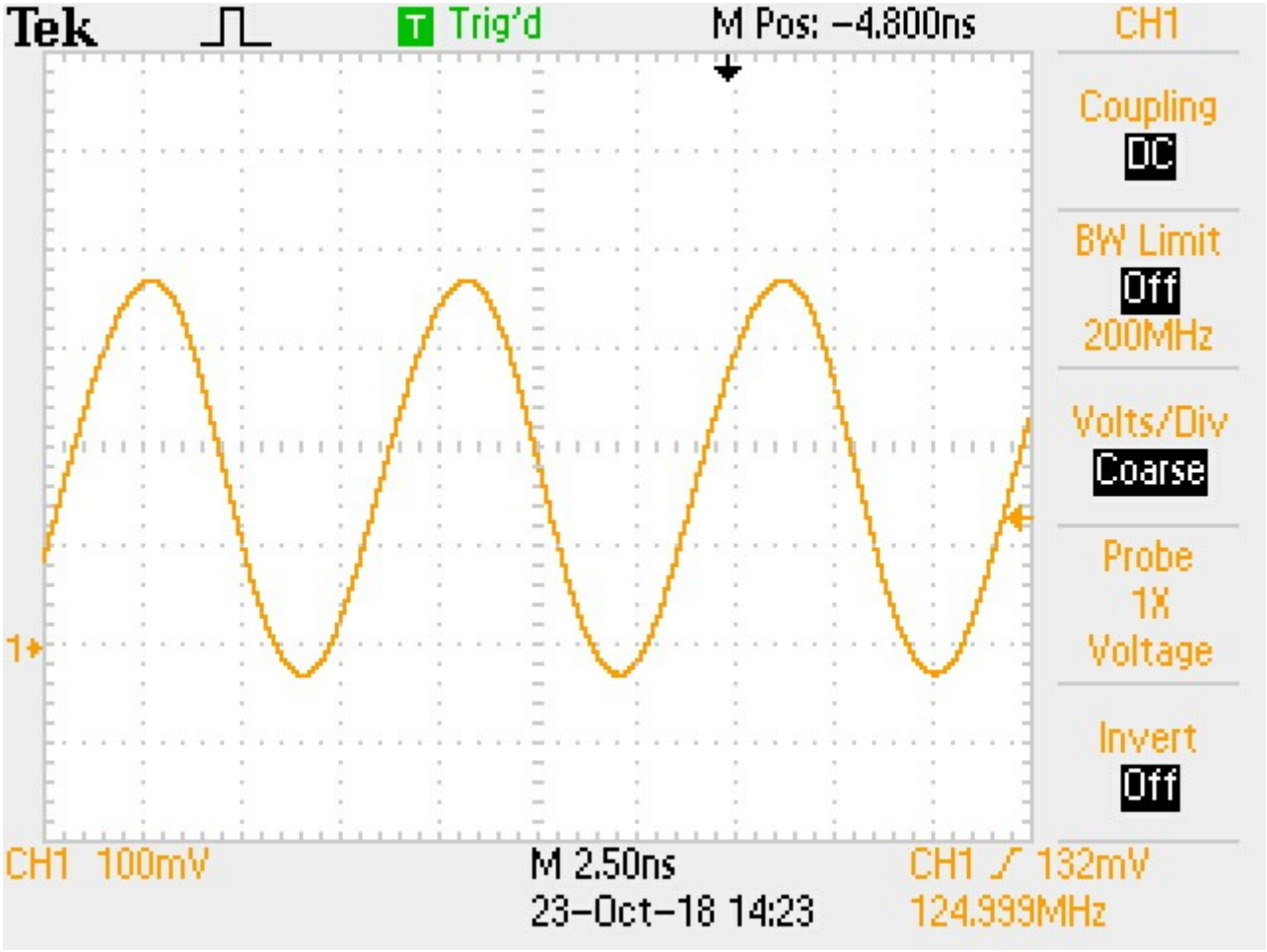

Figura 10: scansione oscilloscopio clock 125 Mhz

Sono stati utilizzati i led del connettore di rete di Arty, per verificare il corretto trasferimento dati e un altro tra quelli disponibili sul dispositivo per verificare l'impostazione della velocità di trasmissione.
Il passo successivo è stato progettare ed eseguire il test-bench, cioè la simulazione comportamentale del dispositivo sottoposto a stimoli; ci siamo serviti per questo scopo del simulatore Modelsim 6. 4ª.
Dopo aver preparato l'ambiente di lavoro, inserendo tutti i file relativi alla simulazione, abbiamo inserito anche qui il modulo Data Generator.

3.2 Realizzazione delle macchine a stati per finiti ed esecuzione dei test-bench.

Quest'ultimo è comandato da una macchina a stati finiti che "mima" il comportamento di un ricevitore di dati preposto a passarli al modulo MAC per la trasmissione Ethernet. Un semplice contatore è stato utilizzato per generare i dati da far arrivare all'uscita, in modo da testare il funzionamento del circuito.
Il comportamento della macchina a stati è descritto dal seguente diagramma:

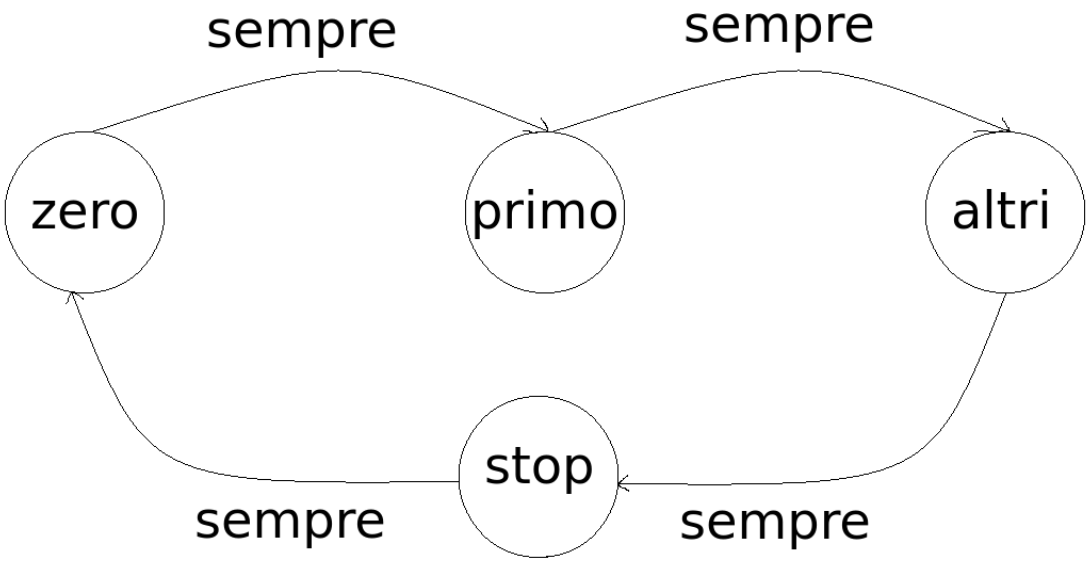

Figura 11:macchina di Moore generatrice di pacchetti

La macchina funziona può avere quattro possibili stati. Inizialmente la macchina si trova nella fase "zero", qui vengono impostati gli ingressi SOP, EOP e WR a zero, il dato in uscita indefinito, dopo di chè si passa nello stato "primo", nel quale il primo pacchetto viene inviato, SOP viene posto a 1, EOP a zero e WR a 1, il dato in uscita sarà valido ;lo stato successivo invece sarà "altri" dove SOP verrà posto a 1, EOP a 0 e WR a 1, nuovamente, anche qui, verrà visualizzato una sequenza di dati. Al passo successivo, lo "stop", i dati continueranno a essere inviati e SOP sarà posto a 0, EOP s 1 e WR.
Infine si ritornerà allo stato "zero", quindi l'invio dei pacchetti sarà concluso. Al prossimo invio, la macchina ripeterà lo stesso cammino.
E' stata realizzata e aggiunta, un'ulteriore macchina a stati finiti, che serviva anch'essa a "mimare" il comportamento di un'altra interfaccia, quest'ultima infatti era relativa all'HOST. Inizialmente, si era pensato a una struttura che rispettasse fedelmente tutto il processo di programmazione dei registri, ma per semplificare e rendere più leggero il codice, si è optato per una struttura simile alla precedente, "nascondendo" la parte prima della lettura del dato (infatti a noi interessa solo quest'ultima), quindi per ognuno degli stati, la macchina è stata impostata per eseguire la scrittura in appositi registri dedicati.
Vediamone il funzionamento in dettaglio:dopo il segnale di reset, la macchina entra immediatamente nello stato di Idle, dopodichè, attraversa i vari step, seguendo questo preciso ordine:imposta per prima cosa la velocità del protocollo, l'indirizzo MAC, il MAC DATA, seleziona MAC WR HIGH, dopo MAC WR LOW e qui vi è una parte, in cui si devono attendere sei cicli di loop, per premettere la scrittura dei dati, secondo le specifiche dell'autore del core (vengono ripetuti gli ultimi quattro step).

Dopo ciò, vi è l'abilitazione del MAC, il MAC enable, ed infine, si giunge allo stato di stop, dove si continua a rimanere, sino a quando, viene effettuato un nuovo reset della macchina, dopo il quale essa riprende il normale funzionamento.

Nel seguente diagramma, si può osservarne il funzionamento.

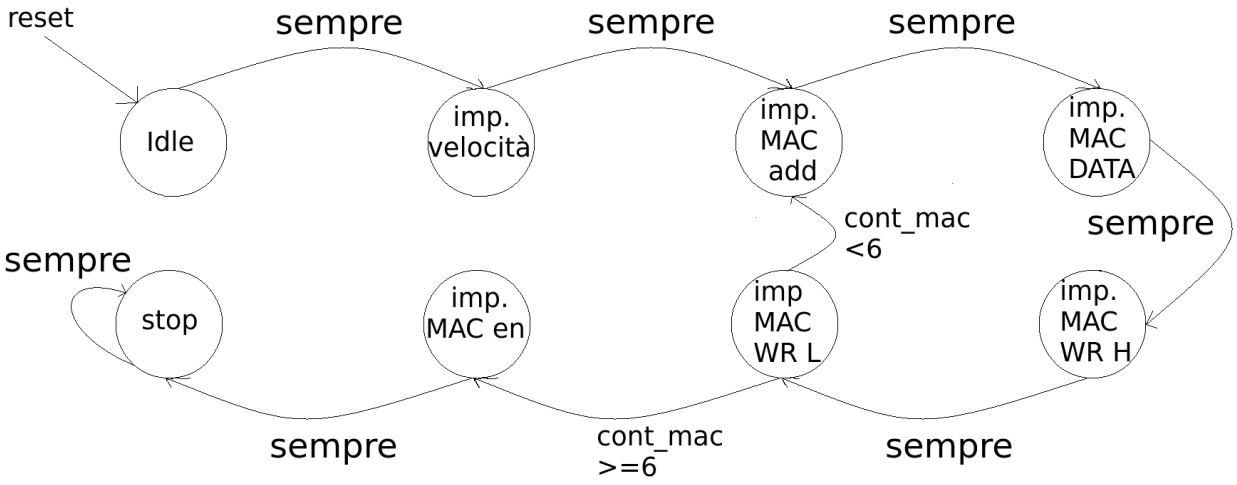

Figura 12: macchina di Moore per impostazione registri e velocità.

Come accennato il core per il progetto, è stato ricostruito ad hoc, rispettando tutte le specifiche utilizzate dall'autore, per interfacciarsi correttamente ad esso, anche per eseguire i test-bench (ha utilizzato uno script automatico ma in ambiente diverso), quindi, prendendo spunto da quelli presenti lui, sono stati modificati, inserendo la parte che interessava.
Il test-bench, una volta completato è un sistema automatico di verifica, che, una volta fatto partire visualizza su schermo, le forme d'onda relative ai vari segnali.
I test-bench, risultano molto utili, in quanto permettono di avere una previsione della trasmissione dei dati, in modo eventualmente, da modificare il Data generator per ottenere una corretta trasmissioni dei dati e livellare eventuali anticipi o ritardi.
In figura 13 e 14, si possono osservare il diagrammi wave del test-bench.
Nel primo si osservano, in modalità decimale, per facilità di lettura i dati emessi dal modulo DATA GENERATOR, sulla MAC_DATA e il dato nella riga Speed, impostato al valore 010 in binario, indicante la velocità di trasmissione di 100 Mb/s

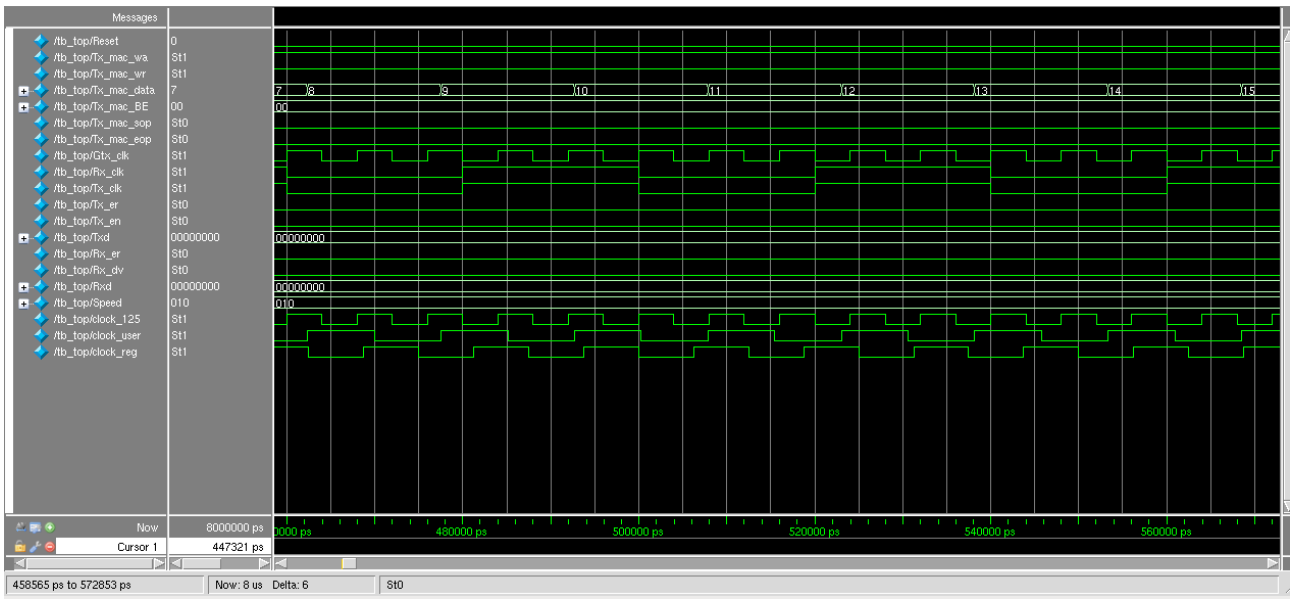

Figura 13: test-bench, grafico wave. Dati emessi dal modulo Data Generator

Nel secondo si osservano, anche qui in decimale, i dati emessi, sulla riga Txd e il dato nella riga Speed, impostato al valore 010 in binario, indicante la velocità di trasmissione di 100 Mb/s

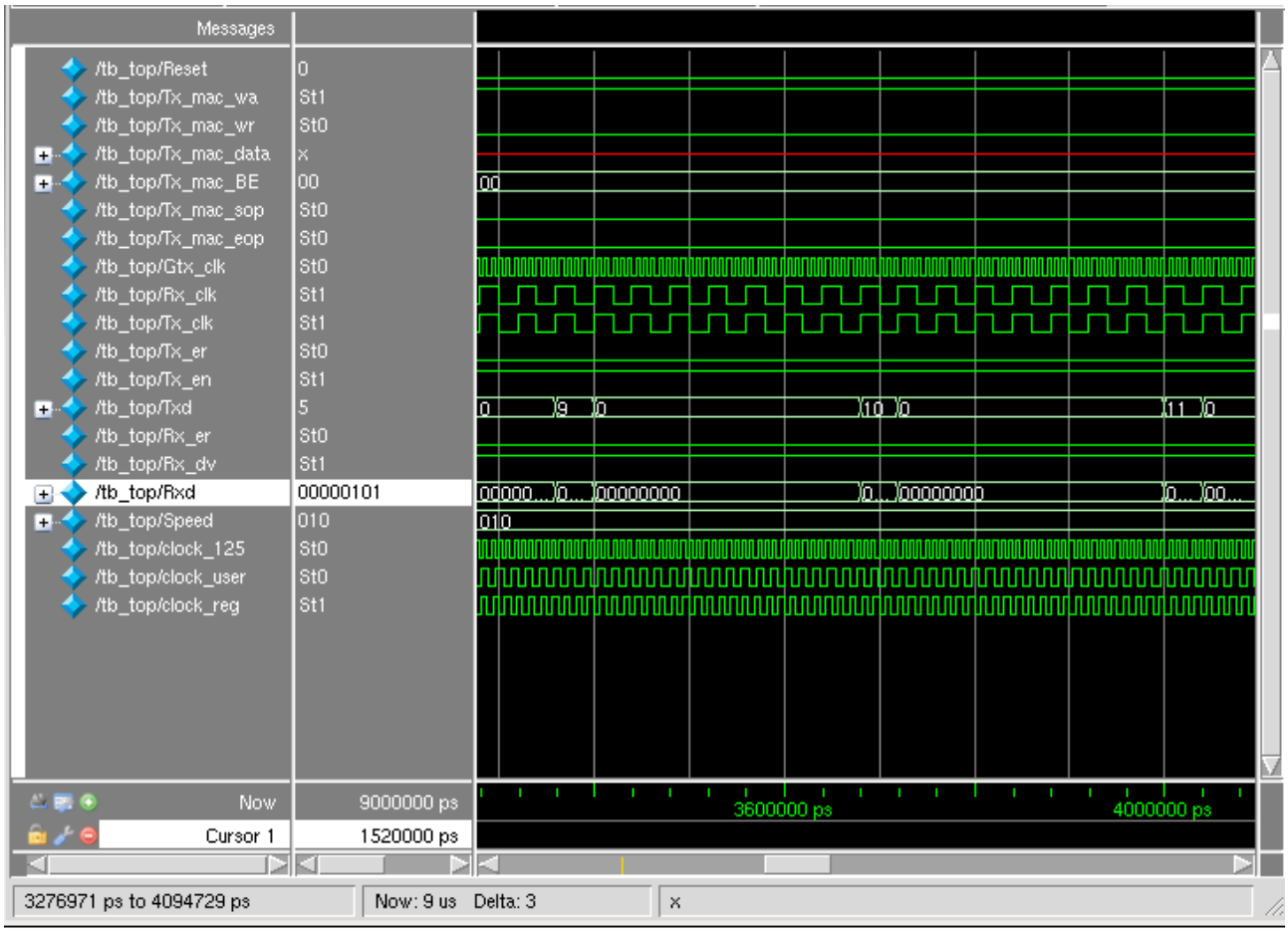

Figura 14: Test-bench, grafico wave. Dati emessi dal circuito MAC compatibili con l'interfaccia fisica

Non è stata eseguito il test per connessione 1 Gb, in quanto il sistema di connessione AER, da test eseguiti dall'autore, si attesta al massimo a 100 Mbps (2. 9 Meps con eventi da 32 b), di conseguenza, non necessario.
In seguito si doveva caricare sul kit di sviluppo Arty, tramite cavo usb e software Vivado il core opportunamente modificato e validato.
Dopo il caricamento, è stato misurato all'oscilloscopio il segnale di clock richiesto al funzionamento dell'interfaccia di rete, come descritto in precedenza.
Una volta eseguito ciò, era necessario testare tramite il software Wireshark, uno sniffer, che i dati inviati dal pc, tramite data generator, venissero trasmessi

3.3 Caricamento del progetto su FPGA e test trasmissione dati tramite pc.

Come accennato, è stata eseguita la sintesi con successo ed anche l'implementazione su Vivado, quindi, dopo alcuni passi, come la generazione del bitstream e il caricamento sul device, si può verificare direttamente su FPGA, il circuito realizzato.
Dopo aver programmato il dispostivo Arty, e collegato l'interfaccia di rete ad un adattatore ethernet/usb aggiuntivo, in quanto è stata impostata come non gestita dall'interfaccia, l'accensione della spia verde sul dispositivo (appositamente impostata, seguendo lo schema rosso, 10 Mb/s, osservabile durante il reset del sistema, verde 100 Mb/s e blu 1 Gb), indicando la corretta selezione della velocità di utilizzo del core, impostata a 100 Mb/s. Nella fotografia in 15 è osservabile appunto il led in questione, in basso, durante la fase di test.

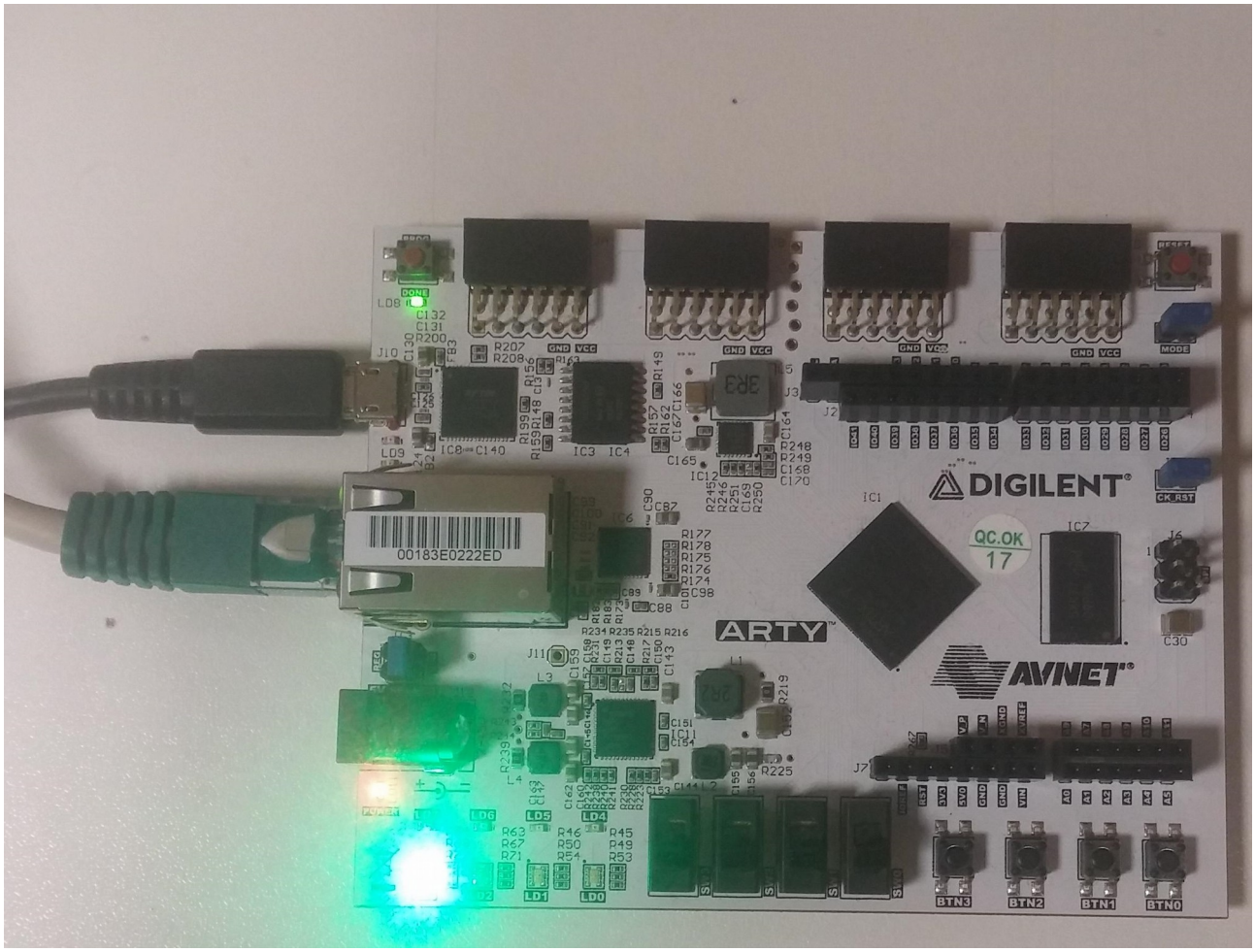

Figura 15:led verde impostazione velocità 100 Mb/s

Per quanto riguarda la corretta trasmissione dei dati, è visibile dal led destro del connettore LAN di Arty, lampeggiante di colore arancione, controprova della corretta impostazione della velocità, come mostrato in figura 16.

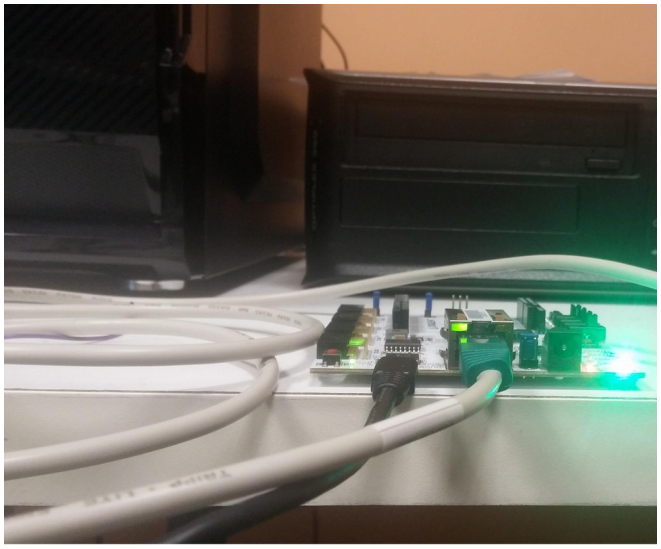

Figura 16: Led destro arancione lampeggiante stato dati e 100 Mb/s, fra una trasmissione e l'altra.

Successivamente si è utilizzato il software Wireshark per il controllo dei pacchetti ed una volta impostato correttamente è stato avviato, visualizzando così l'avvenuta comunicazione tra dispositivo e pc.

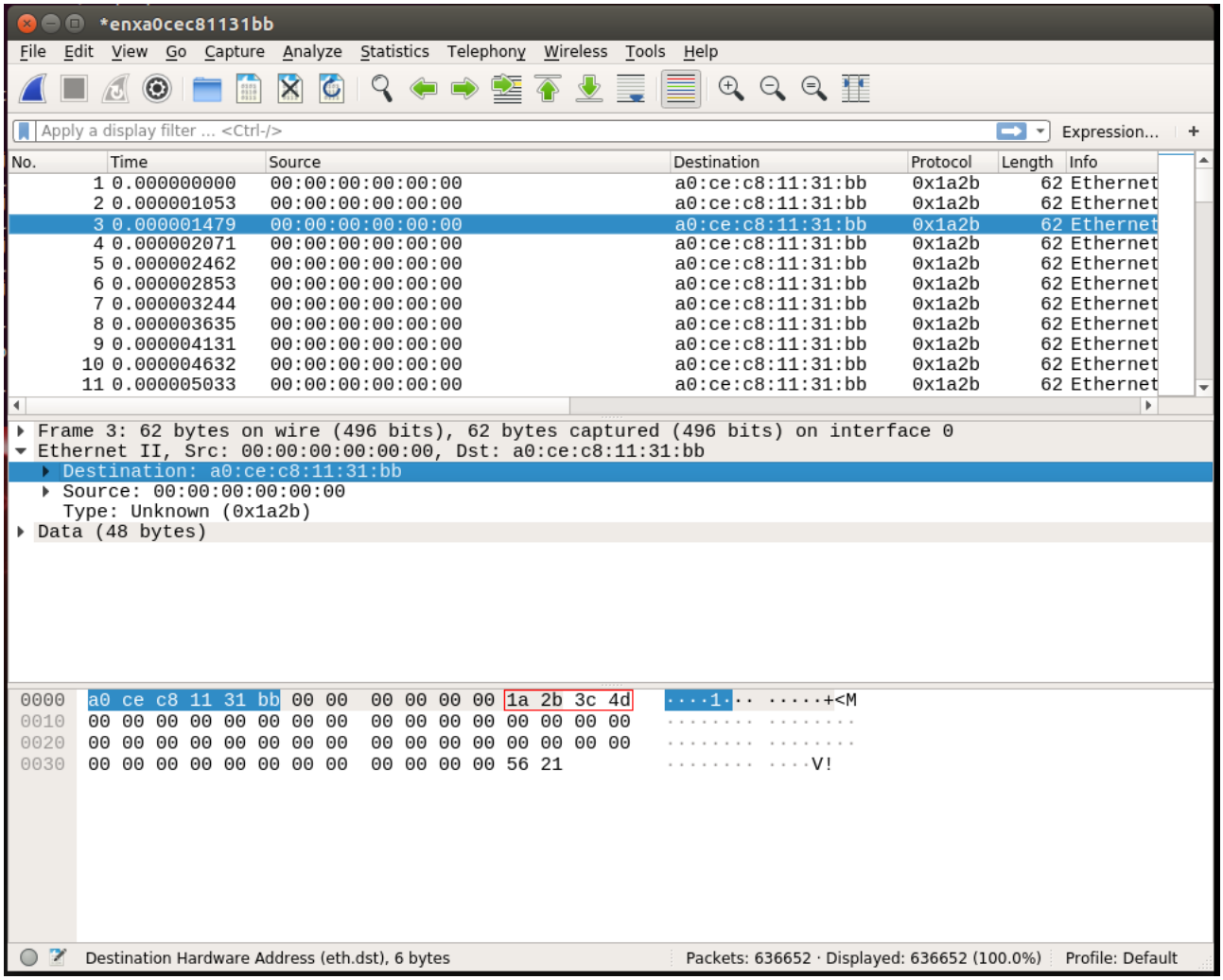

Figura 17: Cattura Wireshark. In blu il MAC address, in rosso un pacchetto ricevuto.

Nell'immagine (fig.17) si può osservare una comunicazione, in cui è evidenziato in blu, nella parte superiore della schermata, il numero del pacchetto ispezionato e dettagli nell'intestazione. Nella parte sottostante, invece, sempre in blu, a sinistra, l'indirizzo MAC del destinatario, in rosso invece, i dati appena trasmessi.

4 Conclusioni

Il progetto assegnatomi per questa tesi, rappresentava un'interessante sfida riguardo l'ottimizzazione di un protocollo molto semplice, ma dai molteplici utilizzi, come già ampiamente descritto nelle prime pagine di questa tesi.
La scelta di utilizzare un core già realizzato, deriva dal fatto di avere una base di partenza solida e funzionante (come indicano i test eseguiti dall'autore) e il fatto di avere licenza libera non ha posto problemi nel suo utilizzo per questa nuova realizzazione.
Senza dubbio, è stato realizzato un corposo lavoro di adattamento per l'utilizzo con il protocollo AER, in quanto l'IP originale era stato ideato per l'utilizzo con Ethernet a 10/100/1000 Mb/s, compreso il protocollo, quindi si è provveduto a selezionare con rigore, solo ciò che era necessario, ottenendo un sistema semplice, per un' utilizzo nell'ambito preposto.
Come descritto nelle pagine precedenti, sia i test-bench che le verifiche sul circuito ne hanno dimostrato il funzionamento, con il corretto invio di dati alla velocità desiderata.
Di conseguenza, siamo in presenza di una piattaforma trasmissiva già pronta all'uso e di facile implementazione anche in altri kit di sviluppo con le opportune modifiche. Altresì, questo approccio progettuale è interoperabile: futuri utilizzi, oltre le workstations potrebbero comprendere unità robotiche o altro, quindi sarebbe necessario ripartire dai sorgenti, apportare le modifiche necessarie alle necessità e dopo le fasi di validazione e testing ed ulteriori verifiche, il progetto qui realizzato, potrebbe essere sintetizzato e prodotto in serie, per essere installato sulle unità che ne avessero bisogno.

5 RIFERIMENTI.

I) https://upload. wikimedia. org/wikipedia/commons/2/23/Innorobo_2015_-_NAO_(cropped). JPG

II) https://upload. wikimedia. org/wikipedia/commons/3/39/ASIMO_4. 28. 11. jpg

III) https://3c1703fe8d. site. internapcdn. net/newman/gfx/news/hires/2017/6-theimportanc. jpg

IV) Parmiggiani, Alberto & Maggiali, Marco & Natale, Lorenzo & Nori, Francesco & Schmitz, Alexander & Tsagarakis, Nikos & Santos-Victor, José & Becchi, Francesco & Sandini, Giulio & Metta, Giorgio. (2012). The Design of the iCub Humanoid Robot. International Journal of Humanoid Robotics. 9. 10.1142/S0219843612500272.

V) https://upload. wikimedia. org/wikipedia/commons/f/f3/PCI-104_EP405. JPG

VI) https://ieeexplore. ieee. org/document/8457469

VII) Point-to-Point Connectivity Between Neuromorphic Chips Using Address Events Kwabena A. Boahen

VIII) https://reference. digilentinc. com/reference/programmablelogic/arty/startredirect=1

IX) http://blog. brixandersen. dk/wp-content/uploads/Part0-Arty-0. png

X) https://opencores. org/projects/ethernet_tri_mode - Jon Gao - 10_100_1000 Mbps tri-mode ethernet MAC