

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea Magistrale in Fisica

**Sistema di misura dei consumi di unità
di calcolo low-power per applicazioni
scientifiche presso il centro INFN-CNAF**

Relatore:

Prof.

Alessandro Gabrielli

Presentata da:

Davide Cammareri

Costantino

Correlatori:

Dott.

Daniele Cesini

Dott.ssa

Elena Corni

Sommario

Questo lavoro di tesi, svolto presso il CNAF (Centro Nazionale per lo sviluppo delle Tecnologie Informatiche e Telematiche) è inserito nell'ambito del progetto INFN COSA (Computing on SoC Architecture).

Si pone come obiettivo quello di testare i consumi e le prestazioni di sistemi computazionali a basso consumo energetico di tipo System on Chip (SoC), i quali stanno emergendo come unità di calcolo su cui eseguire e testare applicazioni scientifiche. A tal proposito si è realizzato un misuratore di corrente basato su Arduino UNO e un trasduttore corrente-tensione, la cui uscita analogica dipende linearmente dal valore della corrente in ingresso fornitagli dall'alimentazione del SoC di cui si vuole testarne il consumo. L'Arduino converte la tensione analogica in uscita dal trasduttore, tramite l'ADC a 10 bit incorporato, in valori di tensione digitali con un'incertezza di 5 mV e una durata temporale di 1 ms. Le misure sono state effettuate tramite una retta di calibrazione corrente-tensione ottenuta nel lavoro di tesi, la quale ha consentito di convertire le letture di tensione salvate nella memoria dell'Arduino, espresse come livello dell'ADC, in letture di corrente. In questo elaborato, si espongono alcune ragioni per cui oggi la comunità scientifica è sempre più interessata a scegliere come unità di calcolo architetture basate su SoC, piuttosto che architetture tradizionali. La tesi illustra tutti gli step che hanno consentito di ottenere la retta di calibrazione tramite la quale sono state effettuate delle misure, per alcune applicazioni fornite dal CNAF, su una scheda low power modello NVIDIA Jetson che monta un SoC Tegra K1. Successivamente sono stati confrontati i risultati del misuratore di corrente realizzato con quelli misurati tramite un multimetro digitale ad alta precisione in dotazione al CNAF.

Indice

1	Introduzione	1
1.1	Breve storia dei calcolatori	1
1.2	Aspetti Pratici	5
2	Progetto COSA	7
2.1	CPU e GPU: differenze architetture	8
2.2	Architetture System On Chip	11
3	Monitoraggio della potenza	13
3.1	Trasduttore corrente-tensione ad Effetto Hall	16
3.2	Conversione analogico-digitale	18
4	Metodo operativo	20
4.1	Materiale di lavoro	20
4.2	Sistema di misura	22
4.3	Misura tempo-corrente	23
4.4	Misura tempo-ADC	24
4.5	Retta di calibrazione	26
4.6	Misure effettuate con il sistema realizzato	29
4.7	Confronto	32
5	Conclusioni	35

*“La tecnologia è
o immatura o obsoleta.”
Clifford Kettborough*

Capitolo 1

Introduzione

In questo capitolo verranno analizzate le principali ragioni che hanno portato all'utilizzo delle architetture SoC *System on Chip*, utilizzate per il lavoro di questa tesi.

1.1 Breve storia dei calcolatori

Il primo esempio di macchina da calcolo risale al 1946 ed è rappresentata dall'E-NIAC (Electronic Numerical Integrator And Computer). Progettata da J. Presper Eckert e John Mauchly, fu realizzata mediante l'utilizzo di 18000 valvole termioniche e 1500 relé. Con un peso di circa 30 tonnellate occupava una superficie di 180 m^2 , la sua potenza dissipata era circa 140 kW ed era in grado di eseguire un'operazione ogni 0,5 ms. Questa macchina era basata sull'*architettura Harvard*, in cui i dati dei programmi e le istruzioni da eseguire erano memorizzati in spazi di memoria diversi [1].

Nel 1952 John Von Neumann, insieme al collaboratore Goldstine, svilupparono all'IAS (*Institute of Advanced Technology*), il calcolatore che rappresentò il prototipo per tutti i computer digitali futuri. L'innovazione consisteva nella caratteristica di immagazzinare all'interno dell'unità di memoria, sia i dati dei programmi che eseguiva, sia il codice e quindi le istruzioni di questi ultimi. Questo modello prende il nome di *Macchina di Von Neumann*, grazie alla quale tutti i successivi calcolatori si basarono su suddetta architettura, diventando così facilmente programmabili.

L'architettura di Von Neumann si compone di cinque parti fondamentali:

- CPU (Central Processing Unit o unità di lavoro): supervisiona tutte le funzionalità del calcolatore digitale. A sua volta è suddivisa in:
 - ALU: (Arithmetic Logic Unit) un'unità operativa, la cui componente fondamentale è costituita dall'unità aritmetica che esegue i calcoli;
 - un'unità di controllo, che ha il compito di leggere e decodificare tutte le istruzioni provenienti dalla memoria;
- Unità di memoria, RAM: (Random Access Memory) in cui vengono caricati i programmi che la CPU deve eseguire;
- unità di input, grazie alla quale vengono inseriti i dati per essere elaborati dal calcolatore;
- unità di output, per visualizzare i risultati delle elaborazioni e restituirli all'utente;
- l'interfaccia del Bus, che rappresenta la parte più importante di tale architettura, ovvero il canale che mette in comunicazione tutte le varie componenti.

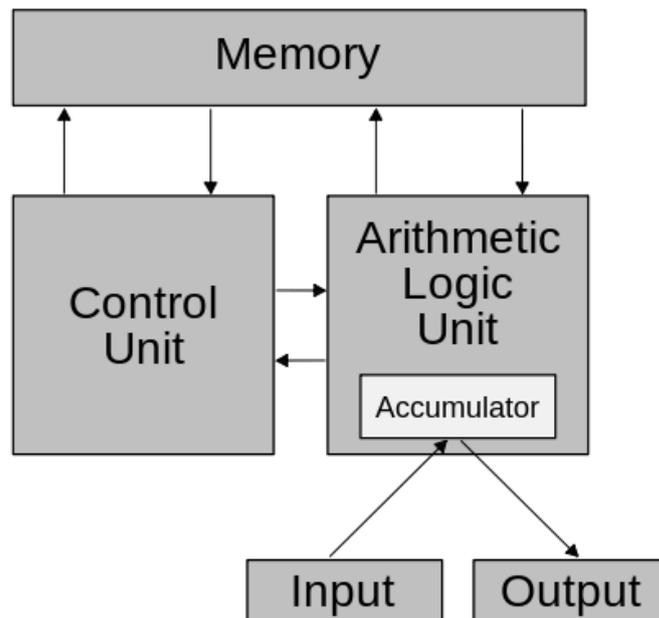


Figura 1.1: Schema a blocchi dell'architettura di Von Neumann [1].

Il Metal-Oxide Semiconductor Field-Effect-Transistor (MOSFET) è stata una delle invenzioni più importanti del secolo scorso. Si tratta di un componente a stato solido utilizzato come interruttore elettronico, pilotato da segnali elettrici, che ha consentito di sostituire le valvole nella costruzione di sistemi digitali. Grazie alla semplicità della sua struttura tecnologica, all'elevata densità di integrazione ottenibile e alla sua affidabilità¹, ha rivoluzionato il settore computazionale e permesso di aumentare la potenza di calcolo dei dispositivi, riducendone anche lo spazio fisico occupato.

Il 1958 ha visto la realizzazione, da parte di Werner Jacobi e Geoffrey Dummer, del *circuito integrato*, il quale ha dato un secondo contributo allo sviluppo computazionale, permettendo di realizzare un unico chip composto da più transistor e ridurre ulteriormente lo spazio fisico occupato, permettendo così un incremento del numero di componenti su ciascun chip, con costi di fabbricazione ridotti.

Grazie a questi dispositivi, si è assistito di anno in anno allo sviluppo tecnologico dei calcolatori che hanno permesso un notevole aumento delle operazioni logico-matematiche eseguibili per unità di tempo. L'efficienza di un sistema digitale si misura attraverso due parametri fondamentali: il tempo di calcolo e lo spazio in memoria occupato. Negli anni, per migliorare questi aspetti si è assistito ad un perfezionamento dell'hardware, diminuendo il tempo d'esecuzione di un'operazione e aumentando il numero di operazioni eseguibili contemporaneamente. Negli anni fra il 1956 e il 1965, l'imprenditore informatico cofondatore della Intel, Gordon Moore, si rese conto che la densità dei transistor all'interno dei chip raddoppiava ad intervalli di tempo regolari. Questa constatazione oggi è nota come *Legge di Moore*, ed è rimasta valida per più di 40 anni. Inizialmente fu formulata dallo stesso Moore come:

“La complessità dei componenti a costo minimo raddoppierà all'incirca ogni due anni.”

¹Si deteriora meno un transistor in trent'anni che una valvola in uno (cit. professor S. Mangione, corso di Fondamenti di comunicazioni elettriche all'UNIPA).

1.2 Aspetti Pratici

Il consumo di potenza e la velocità di calcolo sono limitate fisicamente dalle caratteristiche dei componenti, in gran parte transistor CMOS, che al di sotto un certo livello di miniaturizzazione, possono essere affetti da capacità parassite² fra le interconnessioni più vicine,³ dovute all'architettura e al materiale dei chip stessi, compromettendone le *performance* in frequenza [6]. Per questa ragione la velocità del clock⁴ dei processori, attualmente, è saturata intorno ai $3GHz$ (figura 1.3). Tutte le idee alternative, quali potrebbero essere interconnessioni ottiche, vedrebbero il loro limite nella termodinamica, in quanto il calore è diventato un problema rilevante con l'aumentare della densità dei transistor; quindi tenere il passo evolutivo in termini classici è diventato improponibile o comunque molto difficile.

La strada scelta da Nvidia per le GPU⁵ per lo sviluppo di sistemi più performanti, che può far rivalere la legge di Moore, è il concetto di *parallelismo*. La strategia da adottare sarà dunque quella di costruire microprocessori *multi-core*, ovvero che contengano più CPU e fornire di conseguenza una piattaforma che supporti la crescita economica e l'innovazione commerciale.

Secondo il leader scienziato di Nvidia William J. Dally l'obiettivo per il futuro è realizzare computer paralleli efficienti dal punto di vista energetico, nei quali molti core di calcolo, ognuno ottimizzato per l'efficienza e non solo per la velocità, lavorano insieme nell'elaborazione dei dati alla soluzione di un problema [7]. Raddoppiare il numero di processori, potrebbe duplicare le prestazioni di molti programmi. Incrementare di due volte i transistor in una CPU, invece porta un aumento modesto delle prestazioni, con conseguente consumo energetico molto elevato. Secondo William J. Dally infatti: *“Si potrebbe incrementare il numero dei transistor e dei core di quattro volte ogni tre anni. Facendo lavorare ogni core leggermente più lentamente e perciò in maniera più efficiente, possiamo triplicare*

²Capacità non previste in fase di progettazione di un determinato circuito, ma che si presentano in fase di realizzazione.

³Sovrapposizioni fra: gate/drain, source/drain.

⁴Parametro usato per indicare la velocità del processore ovvero la frequenza alla quale i chip lavorano.

⁵Graphic Processor Unit.

le prestazioni mantenendo lo stesso consumo totale” [7].

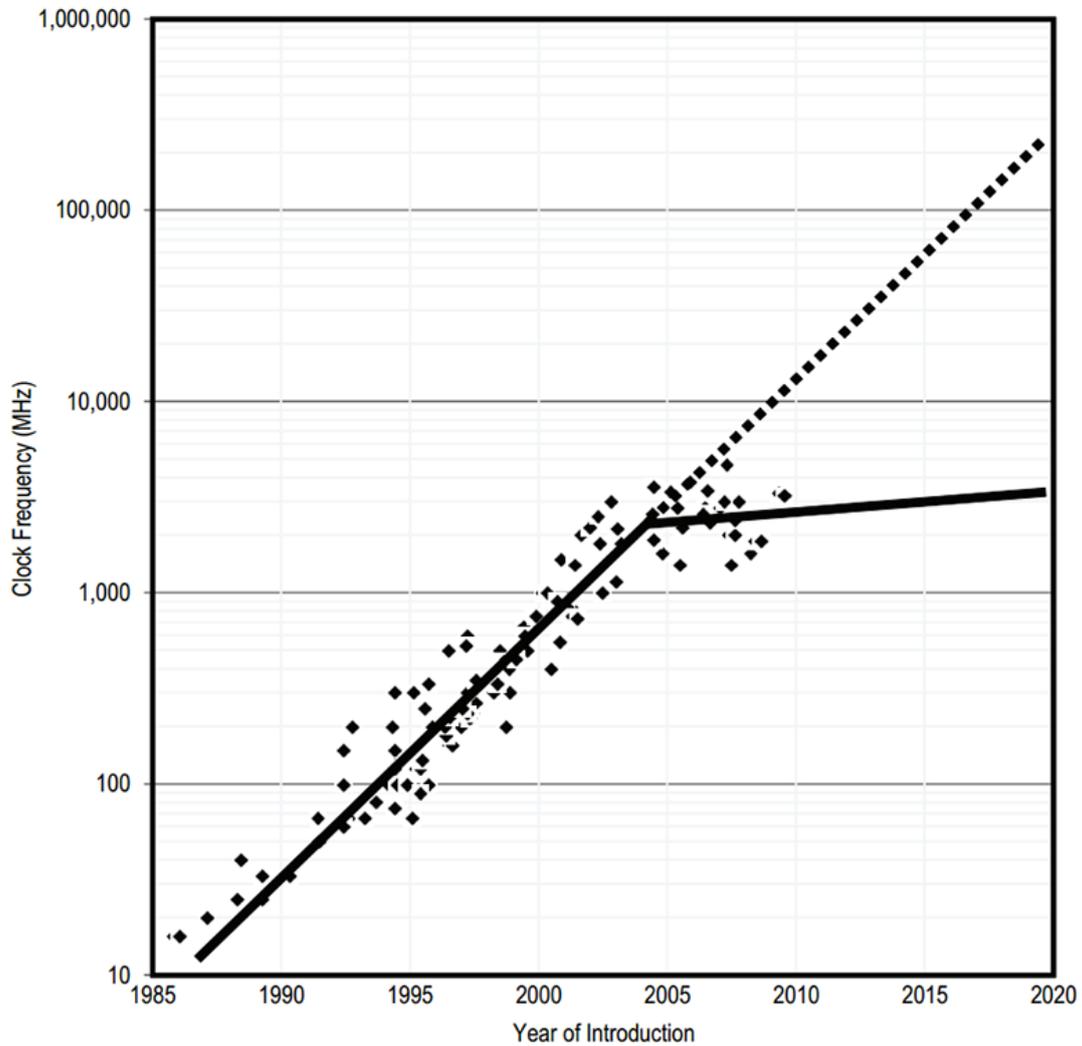


Figura 1.3: *Frequenza di clock per singolo processore in funzione dell'anno di produzione. Si evince la stagnazione intorno a 3GHz e il limite di prestazioni raggiunto in contrapposizione alla predizione di Moore (linea tratteggiata in alto)* [8].

Capitolo 2

Progetto COSA

Acronimo di: *Computing on SoC Architecture* è un progetto triennale (2015-2017) finanziato dalla Commissione scientifica V dell'Istituto Nazionale di fisica nucleare (INFN). L'obiettivo principale del progetto è quello di indagare le prestazioni, il costo totale e le possibilità offerte da sistemi informatici basati su architetture *System on chip* (SoC) che consistono di piattaforme low-power, non progettate per il calcolo scientifico, ma per dispositivi a batteria quali smartphone e tablet, nei quali non è importante la potenza di calcolo. Le prestazioni di calcolo saranno misurate grazie all'esecuzione di alcune applicazioni tratte dalla comunità di fisica dell'alta energia (HEP) appartenente all'Istituto.

Dato che il consumo energetico degli attuali sistemi HPC (*High Performance Computing*), relativo alla gestione delle applicazioni, è in aumento, si rende necessaria l'adozione di processori ad alta efficienza energetica. Ciò è rilevante dal punto di vista dei costi operativi e le grandi strutture di elaborazione, stanno valutando l'opzione di considerare non solo il tempo di esecuzione, ma anche la dissipazione di energia. Questo tema è diventato, negli anni, uno dei più importanti per i sistemi e applicazioni degli HPC, tendenza che si prevede continuerà in futuro. L'efficienza energetica sta diventando quindi sempre più importante nel campo degli HPC che consistono di processori di fascia alta progettati per il calcolo scientifico.

Diverse architetture multi-core, come le GPU integrate nei SoC, progettate appunto per soddisfare le esigenze del mercato low-power, sono in rapida evoluzione verso tecnologie sempre più avanzate orientate al risparmio energetico e moni-

toraggio della potenza. Inoltre questo tipo di unità computazionali, attraggono interesse per via delle loro crescenti capacità di calcolo, prezzi sempre più competitivi e per il basso consumo energetico se confrontati con architetture di tipo tradizionale [9].

Le piattaforme hardware SoC, in genere, integrano in un unico supporto fisico tutti i componenti necessari a un sistema computazionale multicore a bassa potenza in combinazione con una GPU. Questi processori dispongono di elevate prestazioni per-watt in rapporto all'efficienza energetica, ma allo stesso tempo richiedono un'attenta programmazione e ottimizzazione dei programmi da eseguire per essere anche efficienti nel calcolo. Inoltre, si possono presentare alcune limitazioni: piccole memorie della CPU, piccole dimensioni della RAM, interconnessioni ad alta latenza e poca memoria [10].

2.1 CPU e GPU: differenze architetturali

Le GPU (graphic processor unit), inizialmente pensate esclusivamente per elaborazione grafica, dal 2006 stanno emergendo come una piattaforma di successo dove testare applicazioni di largo utilizzo in ambito sia scientifico che ingegneristico [11].

La differenza sostanziale nell'architettura hardware, è ciò che differenzia maggiormente le CPU dalle GPU (figura 2.1). Nella CPU il processore è composto da vari core, ciascuno dei quali ottimizzato per l'esecuzione sequenziale delle istruzioni dei dati in arrivo dalla RAM. Ogni core si occupa di una determinata istruzione sino a che questa non è ultimata, generando così un flusso di lavoro sequenziale per ciascun core. Le istruzioni dei programmi sono così eseguite in sequenza multi-core. Sono strutturalmente ottimizzate per questo tipo di operazioni in quanto dotate di unità aritmetiche e logiche molto sofisticate e una memoria cache¹ molto grande, che consente di ridurre i tempi di accesso alle istruzioni e ai dati dei programmi.

Le GPU invece, operano con un metodo che si può definire duale: sfruttano la struttura multi-core e anziché eseguire le operazioni serialmente, ne eseguono di-

¹Dove vengono salvati programmi e istruzioni di recente accesso

verse centinaia contemporaneamente (in parallelo). Ciò quindi porta a una differenza sostanziale sulla loro componentistica: anziché avere pochi core ottimizzati per un lavoro seriale, la GPU ha diverse centinaia di unità di elaborazione, ciascuna delle quali con una frequenza minore rispetto a quella di una CPU tradizionale, ma ottimizzate per lavorare parallelamente. Le varie unità di calcolo infatti sono programmate per eseguire ciascuna piccole porzioni del codice del programma, che a sua volta quindi, dovrà essere ottimizzato per l'esecuzione in parallelo [11]. Oltre a questo, presentano una larghezza di banda notevolmente più ampia rispetto ai processori, arrivando a velocità di trasferimento dell'ordine di centinaia di GB/s, ovvero un ordine di grandezza più alto rispetto alle CPU che arrivano alla decina di GB/s.

Le GPU sono, in definitiva, dei coprocessori e in quanto tali, devono essere connesse ad un processore, generalmente chiamato host (figura 2.2).

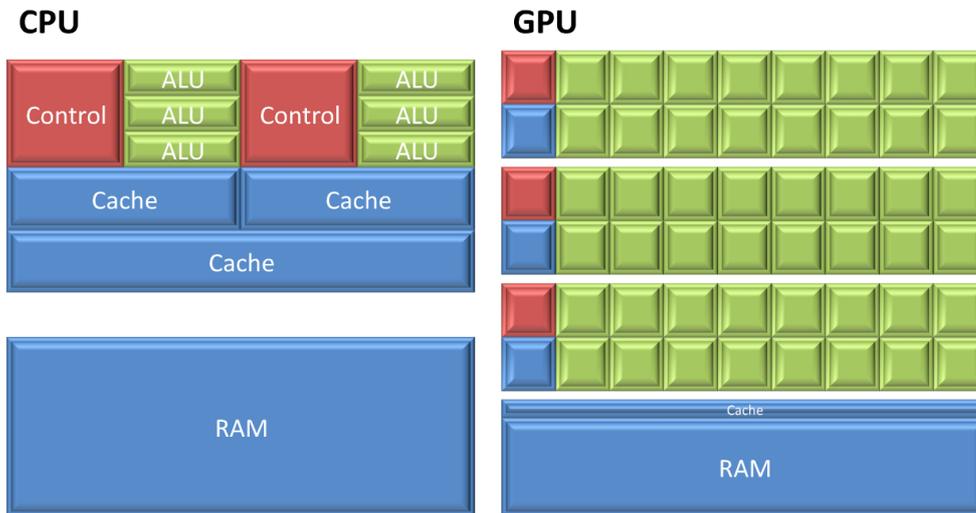


Figura 2.1: Schema a blocchi delle differenze strutturali tra architettura CPU e GPU, nella quale i blocchi verdi rappresentano non più unità ALU (Alogoritmie Logic Unit in blu) ma core di calcolo [12].

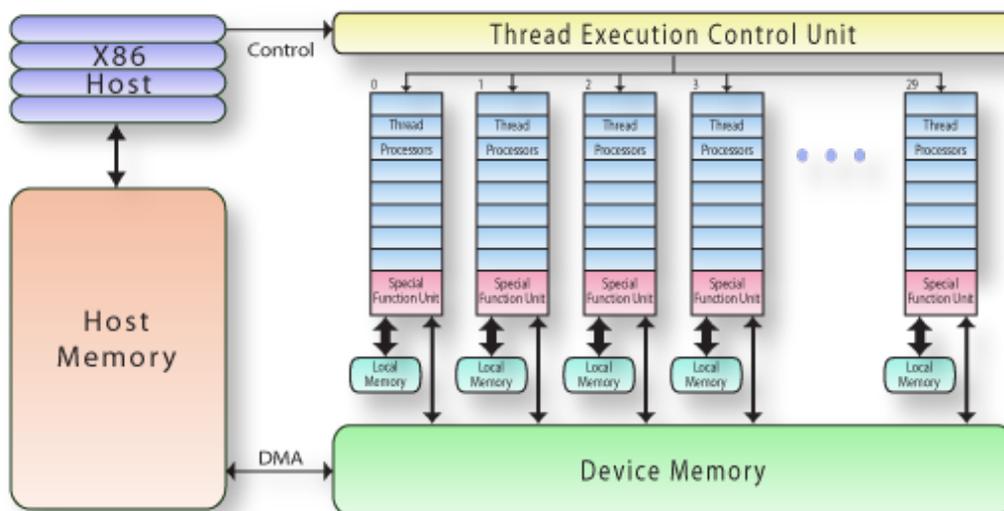


Figura 2.2: Schema di collegamento tra host e device [12].

2.2 Architetture System On Chip

Le architetture a cui si sta interessando sempre più la comunità scientifica sono quelle dei SoC (acronimo di System-On-Chip). Nell'ambito del computing low-power, le prime architetture basate sui SoC, con una potenza di calcolo sufficiente per essere utilizzabili nel settore scientifico sono: SoC ARMv7 based (32 bit), SoC ARMv8 based (64bit) , SoC x86 based (64bit), SoC HSA based (64bit). Questo tipo di architetture hardware, racchiudono in un solo supporto fisico, tutta la componentistica necessaria ad un sistema computazionale: processore centrale (CPU), controller grafico (GPU), blocchi di memoria (RAM, ROM) e interfacce di comunicazione input/output (USB, Ethernet, ecc); sono inoltre in grado di eseguire sistemi operativi embedded² o tradizionali. La differenza sostanziale fra una tradizionale CPU e un SoC sta quindi nell'elevato grado di integrazione di questi ultimi, che si traduce in una minor quantità di transistor rispetto ai processori tradizionali, rendendoli più economici e semplici. Nel 2001, Intel, la principale azienda produttrice al mondo di microprocessori, aveva tentato³ di sviluppare un SoC destinato a PC di fascia economica. Tuttavia per i SoC però si presentavano ancora alcuni limiti: frequenze di funzionamento più basse e generalmente prestazioni mediocri in modalità floating-point (modalità che è stata inserita solo nel 2005, con le generazioni ARMv7) rispetto ai dispositivi classici. Oggi I SoC sono noti soprattutto per il loro largo impiego nel settore mobile in cui l'autonomia, in termini di energia della batteria, è limitata. Questo perché hanno un'ottima efficienza energetica se confrontati con le classiche CPU e GPU. Gli ingenti investimenti, da parte delle aziende produttrici, destinati alla progettazione e realizzazione di SoC che garantiscano migliori rapporti performance/Watt sempre migliori, sono stati ricompensati grazie alla crescita che il mercato mobile ha visto negli ultimi due decenni. Ciò che ha contribuito a ridurre ulteriormente il divario di prestazioni tra SoC e CPU classiche, è anche la gran concorrenza fra le aziende produttrici, che ha anche permesso di mantenere i costi contenuti se paragonati ai tipici server high-end (server di fascia alta). La maggior parte dei SoC vengono acquistati assieme alle loro schede di sviluppo, su cui possono

²Sistemi operativi progettati specificatamente per determinati dispositivi

³Progetto successivamente abbandonato.

essere installati i sistemi operativi opportuni, le librerie e i driver necessari per realizzare un'unità di calcolo funzionante [10]. I SoC più utilizzati nel mondo mobile ed embedded sono basati su IP-ARM⁴, con un design organizzato secondo le architetture RISC (Reduced Instruction Set Computer). I sistemi tradizionali invece, come ad esempio x86⁵, sono meno efficienti dal punto di vista energetico poiché progettati per la performance, con un design organizzato secondo le architetture CISC (Complex Instruction Set Computer), penalizzando di conseguenza i consumi [13]. Visto quindi lo sviluppo di queste nuove tecnologie, oggi, realizzare prototipi di calcolo scientifico funzionanti basati sui SoC, può risultare una valida alternativa se si vuole contenere i consumi. Tuttavia però, ancora oggi, si ha una potenza di calcolo inferiore se paragonata ad un'equivalente combinazione CPU/GPU su tecnologie tradizionali. Per eguagliare la potenza di calcolo delle architetture tradizionali sarà quindi necessario connettere un certo numero di oggetti low-power.

⁴Famiglia di processori commercializzati da ARM Holdings

⁵Famiglia di processori commercializzati dell'azienda Intel

Capitolo 3

Monitoraggio della potenza

Anche se negli ultimi anni i produttori di processori hanno lavorato molto sull'ottimizzazione dei consumi energetici e si sono sviluppati e implementati diversi sistemi di monitoraggio della potenza ad alta risoluzione temporale per cercare di valutare quanto sia conveniente in futuro sostituire le attuali unità di calcolo basate su sistemi tradizionali con sistemi di tipo low-power. Questi sistemi di monitoraggio sono grado di fornire letture di energia per un'applicazione generica. Il primo di questi sistemi, è un software che sfrutta un'interfaccia PAPI¹. Questo software consente alle applicazioni di avviare e arrestare direttamente le misurazioni utilizzando interfacce specifiche dell'architettura, come il RAPL (Running Average Power Limit) per le CPU Intel e la gestione NVIDIA Library (NVML) per le GPU NVIDIA [14]. Tale sistema è particolarmente utile nei sistemi di fascia alta, dove i processori comunemente implementano questi registri.

Il secondo sistema personalizzato, realizzato per il seguente progetto di tesi e di seguito descritto, richiede hardware dedicato e rappresenta una soluzione praticabile ogni volta che le interfacce software citate precedentemente non sono disponibili o mostrano difficoltà nella lettura. Ad esempio, questo misuratore di potenza personalizzato può essere utilizzato per monitorare la potenza delle schede di sviluppo SoC.

La configurazione personalizzata utilizza un convertitore analogico corrente-tensione (trasduttore di corrente LTS 6-NP) e una scheda Arduino UNO, la quale grazie

¹Performance Application Programming interface: si tratta di un'interfaccia portatile che consente di misurare le prestazioni hardware e software dei microprocessori moderni

all'ADC a 10 bit incorporato, può acquisire campioni di corrente con granularità di 1 ms, per digitalizzare le letture della tensione e salvarle in memoria. La scheda Arduino UNO è sincronizzata con la scheda che ospita il SoC attraverso la connessione USB, grazie alla quale è possibile accedervi da remoto. Questa configurazione è in grado di effettuare misurazioni di corrente durante l'esecuzione delle applicazioni sul SoC Tegra K1 (scelto per questo lavoro di tesi, Cap. 4.1) con un'accuratezza di pochi millisecondi. L'applicazione, durante l'esecuzione, può avviare e interrompere le misure in automatico, scrivendo le letture su un file di testo nella memoria dell'Arduino UNO.

Un'altra apparecchiatura per la misurazione della corrente utilizzata per misurazioni di corrente continua, e successivamente, per avere un confronto sulle misure, è costituita dal multimetro digitale ad alta precisione Tektronix DMM4050 (figura 3.1).



Figura 3.1: *Multimetro Tektronix DMM4050 utilizzato per le letture d'intensità.*

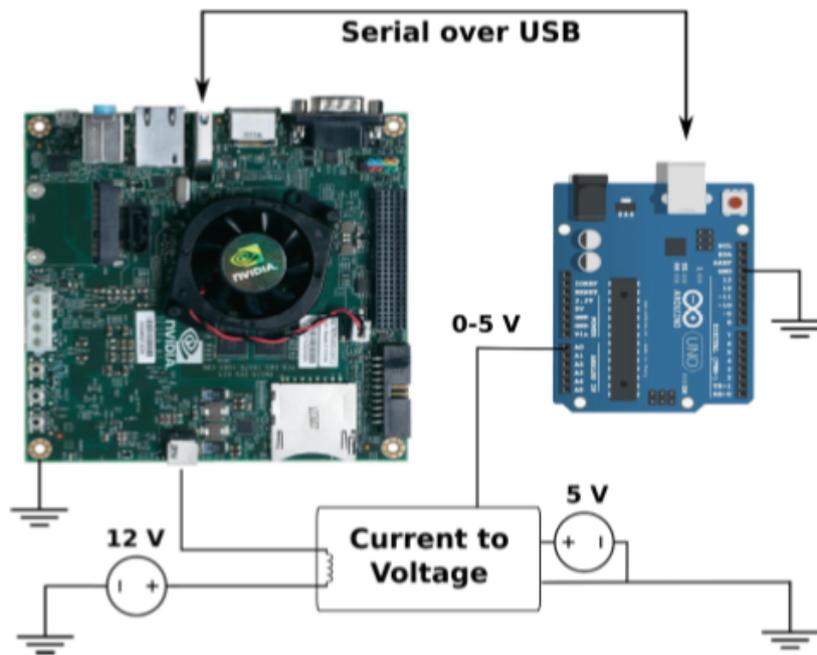


Figura 3.2: Schema circuitale del misuratore di corrente realizzato [15].

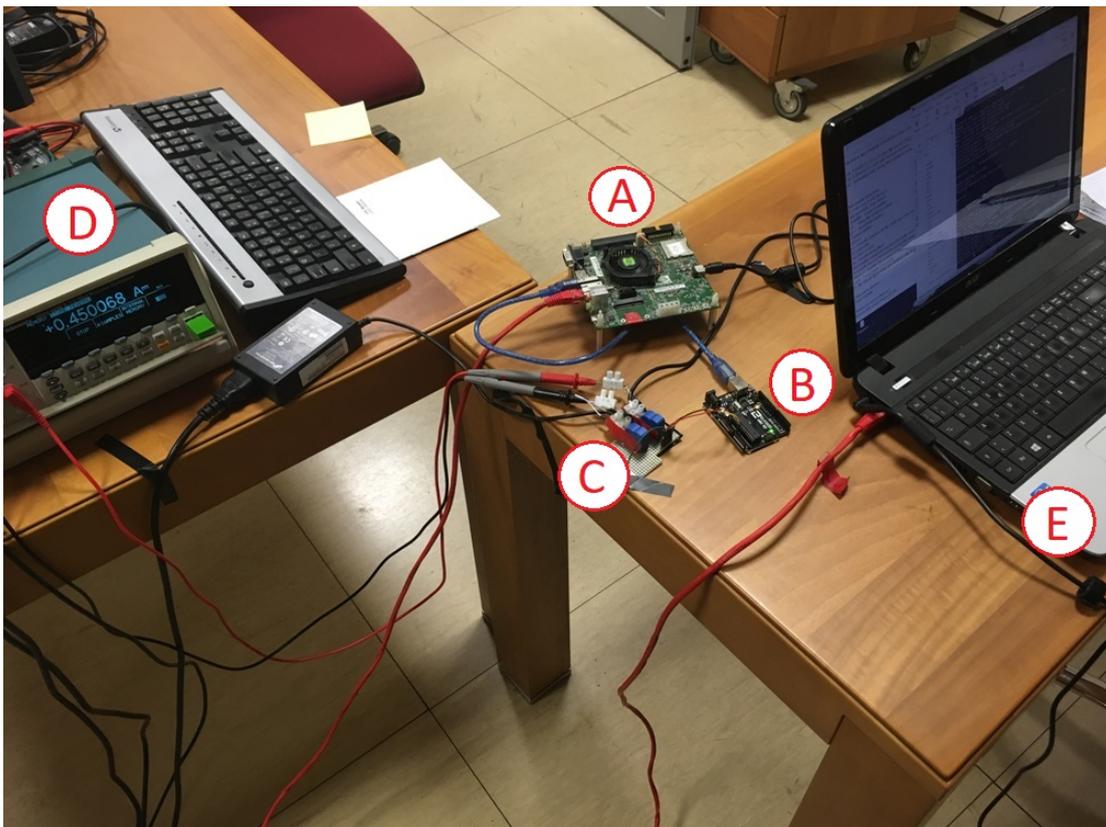


Figura 3.3: Foto realizzata in laboratorio del sistema personalizzato. (A) La Tegra k1, (B) L'arduino UNO, (C) il trasduttore, (D) Il multimetro, (E) il PC utilizzato per raccogliere e analizzare i dati.

3.1 Trasduttore corrente-tensione ad Effetto Hall

Il trasduttore di corrente, utilizzato a questo scopo, è di tipo ad anello chiuso multi-intervallo con utilizzo di tecnologia effetto Hall. Questo trasduttore di corrente viene utilizzato per la misurazione elettronica delle correnti (DC, AC, pulsata, mista) con separazione galvanica ² tra il circuito primario e quello secondario [16].

L'effetto Hall fa riferimento alla tensione misurabile attraverso un conduttore (o semiconduttore), quando una corrente elettrica I che vi scorre nell'elemento sensibile è influenzata da un campo magnetico \vec{B} perpendicolare ad esso. Il campo magnetico applicherà una forza sugli elettroni in conduzione chiamata *Forza di Lorentz*. La forza di Lorentz agisce quindi sugli elettroni in movimento all'interno della lamina conduttrice, determinando fino ad un certo momento la separazione delle cariche, facendo sì, che oltre al campo magnetico presente fin dall'inizio, si venga a creare anche un campo elettrico. Dopo un tempo abbastanza lungo³, le forze fra il campo elettrico che crea una tensione trasversale perpendicolarmente alla corrente applicata e la forza di Lorentz, raggiungeranno una condizione di equilibrio. Grazie al bilanciamento della forza di Lorentz e quella del campo elettrico venutosi a creare, viene generata quella che viene chiamata tensione di Hall, la quale ha direzione perpendicolare sia al verso di scorrimento della corrente che alla direzione del campo magnetico. Tale tensione è quella in uscita del dispositivo di conversione corrente-tensione e dovrà successivamente essere digitalizzata tramite l'Arduino.

Questo tipo di tecnologia ha trovato spazio in molte applicazioni, in particolare nell'ambito degli elettrodomestici moderni e delle applicazioni automotive. L'implementazione di sensori Hall migliora la durata nel tempo e l'affidabilità, in quanto elimina l'usura delle movimentazioni non essendoci stress meccanico.

²Condizione in cui fra due punti a diversa differenza di potenziale non vi è circolazione di cariche, ma l'energia elettrica può essere scambiata tramite altri fenomeni quali: induzione, accoppiamento capacitivo o effetto Hall in questo caso.

³In rapporto al tempo impiegato da un elettrone a percorrere l'elemento sensibile in lunghezza.

In definitiva, il sensore ad effetto Hall, rileva il campo magnetico e produce un segnale (analogico o digitale) opportunamente convertito in uno standard, secondo i requisiti del sistema elettronico [17].

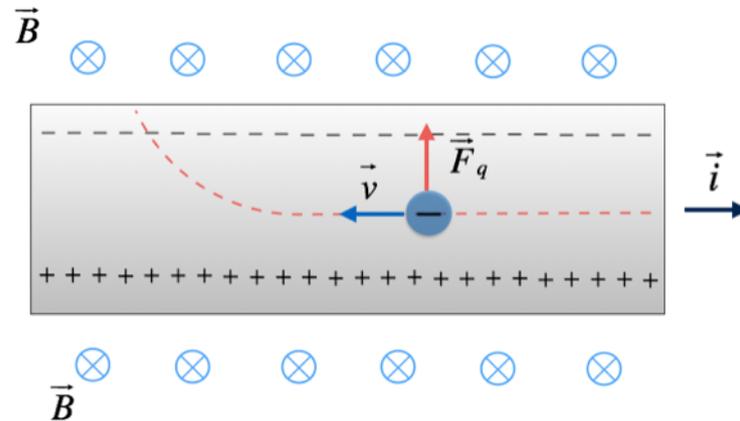


Figura 3.4: Rappresentazione dell'effetto Hall: traiettoria dell'elettrone che a causa dell'effetto della forza di Lorentz, tende deviarlo dalla sua traiettoria originale.

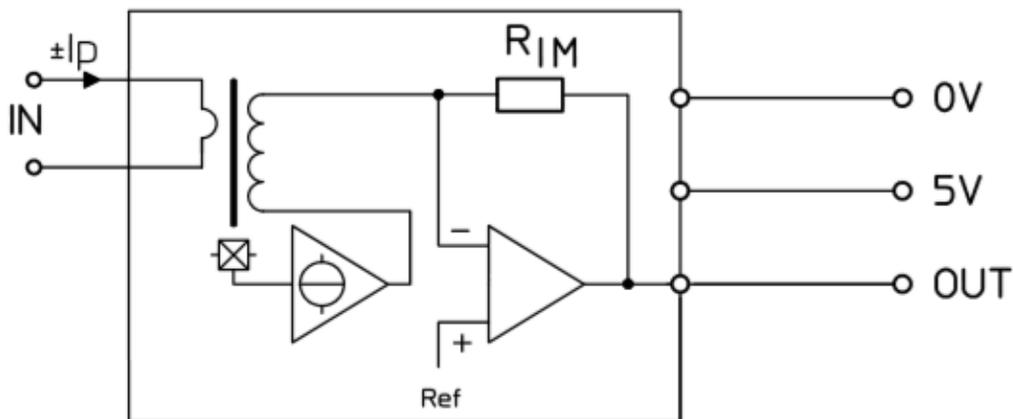


Figura 3.5: Schema circuitale del trasduttore corrente-tensione utilizzato: a sinistra il circuito primario in input, a destra il secondario che genera il campo magnetico con relativo output a destra, $R_{IM} = 208,33\Omega$ internal measuring resistance [16].

3.2 Conversione analogico-digitale

Molti dei fenomeni propri del mondo fisico sono continui, ovvero assumono tutti i possibili valori all'interno di un intervallo definito, tali fenomeni vengono rappresentati da grandezze definite *analogiche*. Ci sono grandezze che invece possono essere espresse da segnali che assumono un insieme di valori finito all'interno di un intervallo. Questi segnali vengono chiamati *digitali*. L'idea che sta alla base della digitalizzazione è la seguente: qualsiasi grandezza fisica di interesse (tensione in questo caso) viene misurata e il valore della sua misura codificato come numero binario; si può ulteriormente pensare di esprimere i segnali, ciascuno dei quali può assumere solo due valori chiamati *bit*: chiaramente un grande numero di questi bit può rappresentare un grande numero di segnali discreti. Se la grandezza assume diversi valori nel tempo, essa sarà misurata a intervalli regolari, dando luogo ad una sequenza di numeri [18].

In questo caso la scheda Arduino UNO utilizzata è in grado di rappresentare i campioni con una profondità di 10 bit. Grazie al processo del *campionamento*, si misurano a intervalli di tempo regolari (in questo caso ogni ms) i valori di tensione in uscita dal trasduttore. I campioni così ottenuti vengono convertiti in numeri binari attraverso un processo chiamato *quantizzazione*. La risoluzione della quantizzazione è funzione del numero di bit usati per rappresentare il valore digitale ottenuto dalla conversione e cresce come 2^n , con n numero di bit utilizzati per rappresentare il valore di ciascun campione. Dell'intervallo di tensione in uscita dal trasduttore, che in questo caso va 0 a 5 V [16], viene suddiviso in 2^n con $n=10$ bit=1024 livelli possibili e per tanto a ogni campione corrisponde un'incertezza di circa 5 mV. Il processo di conversione analogico-digitale, assegna a ciascun campione il numero binario corrispondente all'intervallo di quantizzazione intero più vicino. All'uscita del processo di quantizzazione, quindi, i valori di tensione letti dalla scheda Arduino vengono trasformati in numero binario.



Figura 3.6: Schema del processo di conversione analogico digitale: il segnale analogico $x(t)$ viene suddiviso in x_n campioni di ugual lunghezza temporale a ciascuno dei quali viene assegnato un livello b_n [19].

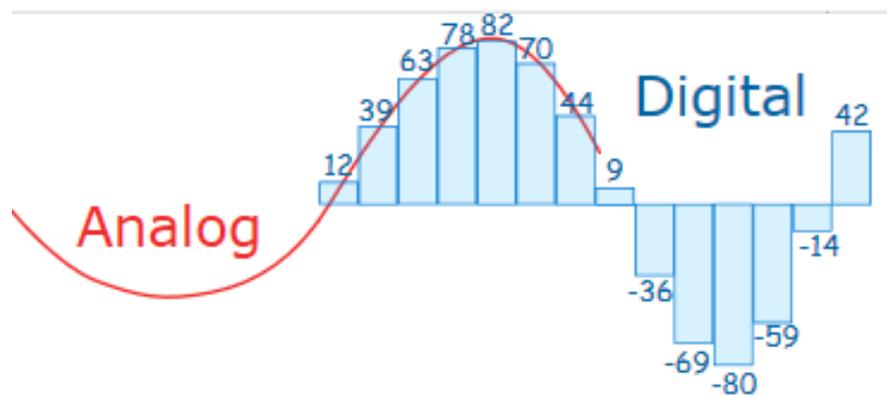


Figura 3.7: Confronto fra segnale analogico a sinistra e digitale a destra.

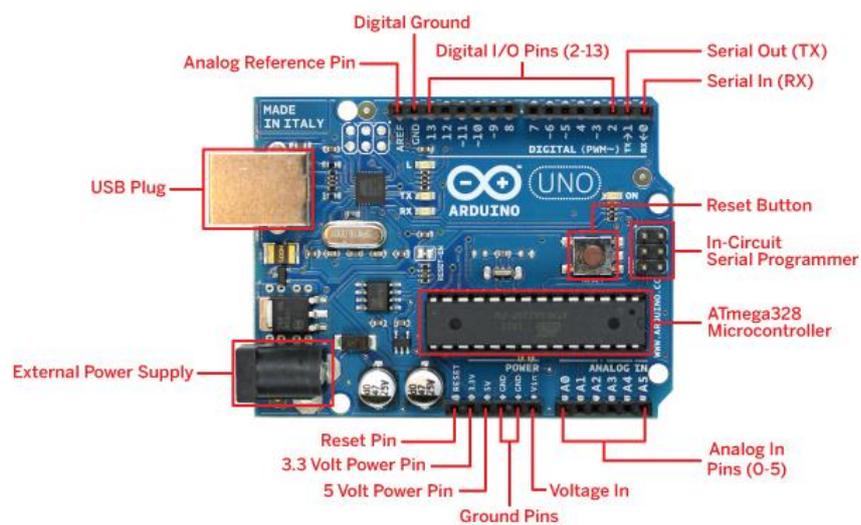


Figura 3.8: Scheda Arduino UNO utilizzata per conversione ADC [20].

Capitolo 4

Metodo operativo

In questo capitolo si descriveranno i passaggi e i metodi utilizzati per trovare la retta di calibrazione corrente-tensione(ADC), grazie alla quale si è misurata l'intensità della corrente assorbita dalla scheda NVIDIA Jetson TK1.

4.1 Materiale di lavoro

Il SoC low-power di cui si è misurata la corrente, durante il questo lavoro di tesi, è l'hardware NVIDIA Jetson TK1. Questo SoC è stato progettato per creare un'importante discontinuità nelle prestazioni dai processori mobili precedenti, ed è stato il primo al mondo ad utilizzare un chip per smartphone, con basso consumo di potenza, che gli consente di avere le stesse caratteristiche avanzate e l'architettura di una GPU desktop [21]. Ne esistono due versioni: una dual-core a 64-bit e un'altra quad-core a 32-bit (fornita per questo lavoro di tesi). Tale dispositivo monta una CPU con 4+1 core ARM Cortex-A15 con frequenza di clock di 2.3 GHz, quattro dei quali usati per l'esecuzione delle applicazioni assegnate e l'ultimo, *shadow core* a basso consumo, dedicato solo alla gestione del sistema quando non serve potenza di calcolo e tiene la scheda accesa quando gli altri quattro core non eseguono nulla. L'hardware della Jetson TK1 include inoltre, delle caratteristiche e alcune funzionalità che la rendono *PC-oriented* come: connessioni SATA¹, mini-PCI, ISP² e una ventola di raffreddamento per consentirne il funzionamento continuo, anche con carichi di lavoro che richiedono un grande impiego della CPU. Dispone inol-

¹Interfaccia per computer utilizzata per connettere hard disk o driver ottici.

²*Internet service provider*: fornitore di servizi internet.

tre di una GPU con architettura Kepler da 192 core CUDA ottimizzati per l'uso efficiente dell'energia, garantendo comunque funzionalità grafiche avanzate [21]. Questa scheda viene fornita con preinstallato il sistema operativo Linux4Tegra OS (fondamentalmente Ubuntu versione 14.04).

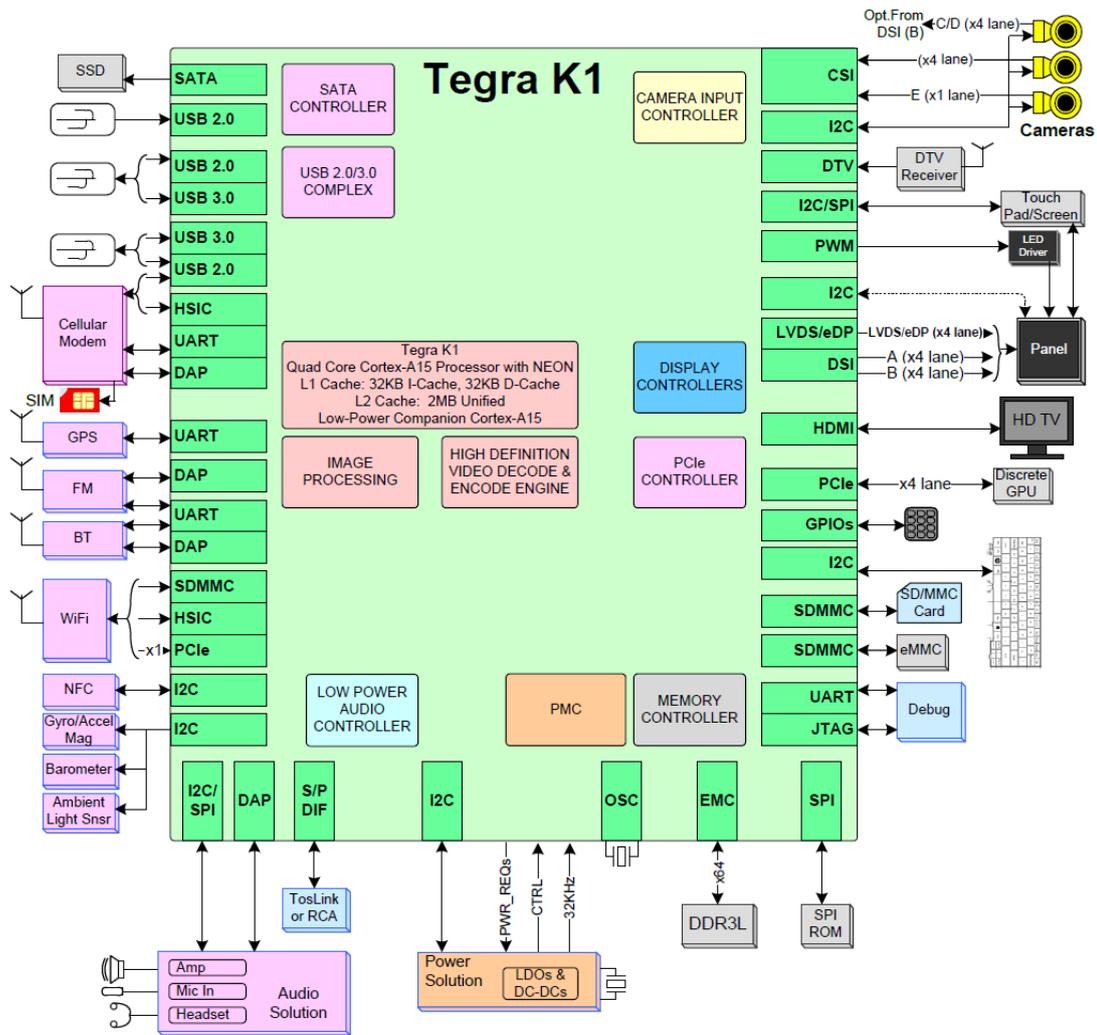


Figura 4.1: Schema a blocchi della scheda utilizzata con tutte le relative possibili connessioni con altri hardware [22].

4.2 Sistema di misura

All'alimentazione a 12 V in corrente continua della scheda NVIDIA Jetson TK1 si è collegato il multimetro ad alta precisione e il trasduttore corrente-tensione LTS 6-NP, come mostrato in figura 3.1. Il trasduttore converte, tramite l'effetto Hall descritto nel capitolo precedente, l'intensità di corrente in entrata in un segnale analogico di tensione all'uscita, secondo una funzione lineare (figura 4.2).

L'uscita in tensione dal trasduttore viene digitalizzata dalla Scheda Arduino UNO tramite un apposito script che converte questo range in livelli ADC con un incertezza di 5 mV ciascuno.

Le misure del livello dell'ADC, durante l'esecuzione delle applicazioni, vengono salvate nella sua memoria, alla quale è possibile accedervi da remoto. Sempre da remoto, si ha accesso alle istruzioni delle applicazioni da eseguire sulla Tegra K1 per testarne i consumi.

Output Voltage - Primary Current

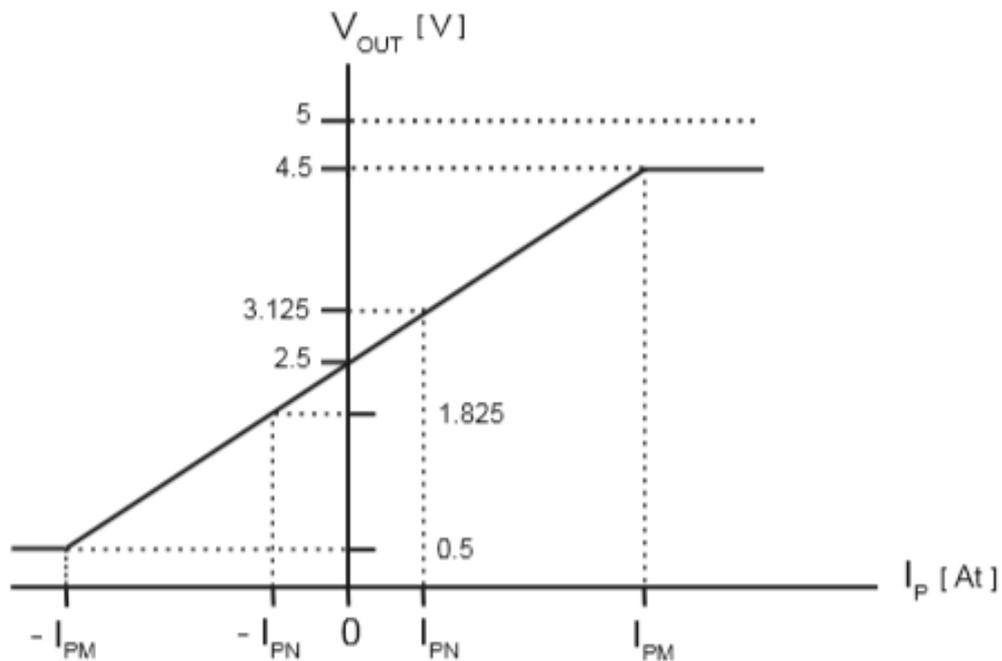


Figura 4.2: Grafico di Funzionamento del trasduttore di corrente: $\pm I_{PM} = \pm 6$ A [16]. Nel caso in questione si è lavorato fra 0 e $I_{PN} = 1,825$ A

4.3 Misura tempo-corrente

Le letture in corrente sono state effettuate tramite il Multimetro Tektronix DMM4050, collegato in serie all'alimentazione a 12 V in corrente continua della Tegra.

Come primo step, per studiare l'andamento della corrente in funzione del tempo assorbita dalla Tegra K1, si è scritto un codice in linguaggio C che ha consentito di scegliere quali applicazioni eseguire e di impostare alcune opzioni durante l'esecuzione come: `sleep`³ e `loop`⁴. Si è scelto di eseguire un'applicazione che conta quanti numeri primi ci sono nei primi cento milioni⁵ di numeri interi. Poco prima di avviare l'applicazione, si è dato lo `start` al multimetro, il quale ha iniziato a raccogliere i dati per scriverli in un file di testo su chiavetta USB inserita in esso. Subito dopo si è avviata l'applicazione tramite terminale, la quale è stata eseguita per misurare l'intensità di corrente in assorbimento, usando rispettivamente, da uno fino a tutti e quattro i core montati sulla Tegra K1. L'applicazione è stata eseguita mandandola in `loop` più volte, per essere sicuri di acquisire un campione sufficientemente grande di dati. Questo è stato possibile vederlo grazie allo `sleep`, impostato a 30 secondi, sulla finestra del terminale nel monitor del PC. Dopo esser certi di aver acquisito un campione abbastanza grande di dati, si blocca l'applicazione tramite il terminale e si dà lo `stop` sul multimetro per interrompere l'acquisizione dei dati. La misura effettuata ha coperto un arco temporale di circa 16 minuti (956 secondi), durante i quali, il multimetro ha effettuato un totale di 3824 conteggi ($3824/956=4$ misurazioni al secondo, per cui si sono acquisiti campioni d'intensità di corrente con una granularità di 0,25 secondi ciascuno). Successivamente i dati sono stati analizzati tramite il programma Microsoft Excel (figura 4.3 pagina seguente).

Nel grafico ottenuto si può notare che, dopo un picco iniziale tipico del primo avvio di ogni applicazione⁶, si osservano quattro *plateau*, con intensità di corrente crescenti, che corrispondono rispettivamente all'utilizzo di: 1, 2, 3, 4 core della Tegra K1, lo `sleep` e il successivo riavvio.

³Tempo che intercorre fra il termine di un'applicazione e l'avvio della successiva.

⁴Quest'impostazione consente di ripetere l'esecuzione dell'applicazione.

⁵Successivamente ridotto a dieci milioni.

⁶Dovuto alle conformazione delle celle di memoria volatile realizzate a MOS che hanno bisogno di caricare tutte le capacità di gate.

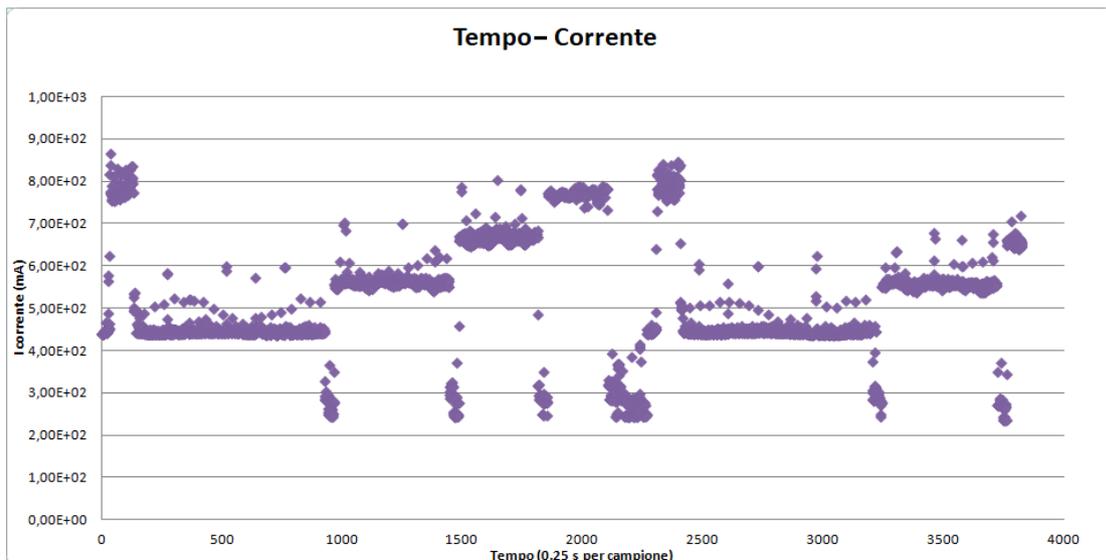


Figura 4.3: Grafico della corrente in funzione del tempo: si notano i quattro plateau, lo sleep e il successivo riavvio.

4.4 Misura tempo-ADC

Per misurare la relazione tra l'intensità di corrente in entrata e la tensione in uscita dal trasduttore in funzione del tempo, l'Arduino UNO ha effettuato le misure e le ha scritte su un file di testo, in cui si trovano i valori in canali ADC al variare del carico della CPU della Tegra K1. Questo procedimento è stato effettuato contemporaneamente all'esecuzione della stessa applicazione descritta precedentemente. In questo caso, i dati sono stati acquisiti con una granularità di un millesimo di secondo, per cui il file di testo in memoria dell'Arduino conteneva oltre 2,4 milioni di elementi. Per questo motivo è stato necessario ridurli tramite uno script Python, che in questo caso⁷, ha mediato le misure dell'ADC a gruppi di 900 ciascuno. Questo procedimento, ha consentito di ottenere, un numero di elementi dello stesso ordine di grandezza dei conteggi presenti sul file delle misure della corrente in funzione del tempo e ha reso gestibili i dati tramite il programma Microsoft Excel⁸.

⁷Veniva utilizzato ogni qualvolta i dati non erano gestibili tramite Excel.

⁸Tale programma non consente di graficare consecutivamente più di 32000 punti.

```

# TempoADC
with open("TempoADC.txt", "r+") as pf: #apre il file e crea puntatore
    k=0 #inizializza righe file
    for line in pf: #calcola tutte le righe presenti nel file
        k=k+1
    print (k)
    k=k/900 #dividile per 900
    print ("Numero di Gruppi[%d]"%(k)) #ora k è il numero gruppi

M_array={} #lista di gruppi
with open("TempoADC.txt", "r+") as pf: #apre il file e crea puntatore
    for q in range(0,int(k)): #varia q da 0 a k=900
        array=[] #dichiara il vettore da riempire con 900 valori
        for i in range(0,900):
            array.append(float(pf.readline())) #concatena
        M_array[q]=array #assegna numero gruppo nella lista

with open("MediaTempoADC.txt", "w+") as pf: #scrive su un nuovo file la media dei 900
    for i in range (0,len(M_array)):
        sum_GRP=0 #inizializza la media gruppi
        for j in range (0,900): #qua somma tutti gli elementi di ciascun gruppo
            sum_GRP=sum_GRP+M_array[i][j]
        print ("Gruppo [%d] Media Gruppo[%f]"%(i,sum_GRP/900)) #stampa tutte le medie e le divide per 900
        pf.write(str(float(sum_GRP/900))+'\n') #riempie il file coi dati medi
print("End")

```

Figura 4.4: Script Python utilizzato per comprimere i dati delle misure dell'ADC.

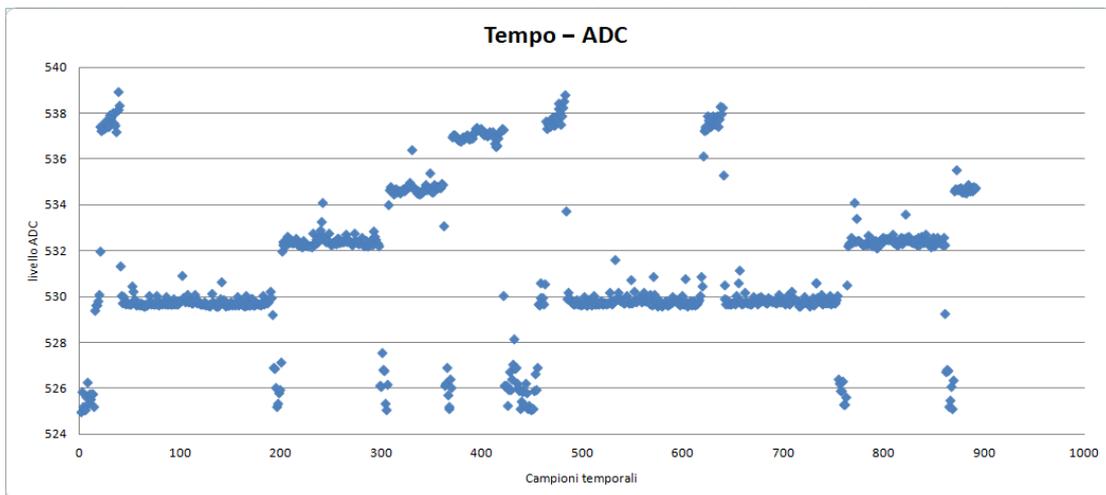


Figura 4.5: Grafico dei valori dell'ADC in funzione del tempo. Si noti l'andamento pressocchè identico al grafico precedente.

Anche in questo caso si possono notare i quattro *plateau*, in cui i livelli di ADC crescenti corrispondono al variare del carico della CPU della Tegra K1, che ricordiamo, elabora la stessa applicazione utilizzando da uno fino a tutti e quattro i core in sua dotazione.

4.5 Retta di calibrazione

Per costruire la retta di calibrazione fra intensità di corrente e livello dell'ADC si è fatto ricorso alla costruzione di due istogrammi, riportati nella pagina seguente. Il primo (figura 4.6) corrisponde ai conteggi della ricorrenza di una determinata misura d'intensità di corrente, con un'ampiezza del bin di 25 mA. L'altro istogramma (figura 4.7), rappresenta la ricorrenza media del livello di ADC misurato durante l'arco temporale di nostro interesse. In tali istogrammi si possono notare cinque picchi, quattro dei quali molto pronunciati, corrispondenti alle misure temporali al variare del carico della CPU su ciascuno dei 4 *plateau* (figura 4.3 e 4.4); rispettivamente per intensità di corrente e livello dell'ADC, misurati all'aumentare del numero dei core con cui si eseguiva l'applicazione. Grazie alla corrispondenza di questi picchi, è stato possibile ricostruire una tabella che assegna dei punti nel piano corrente-tensione.

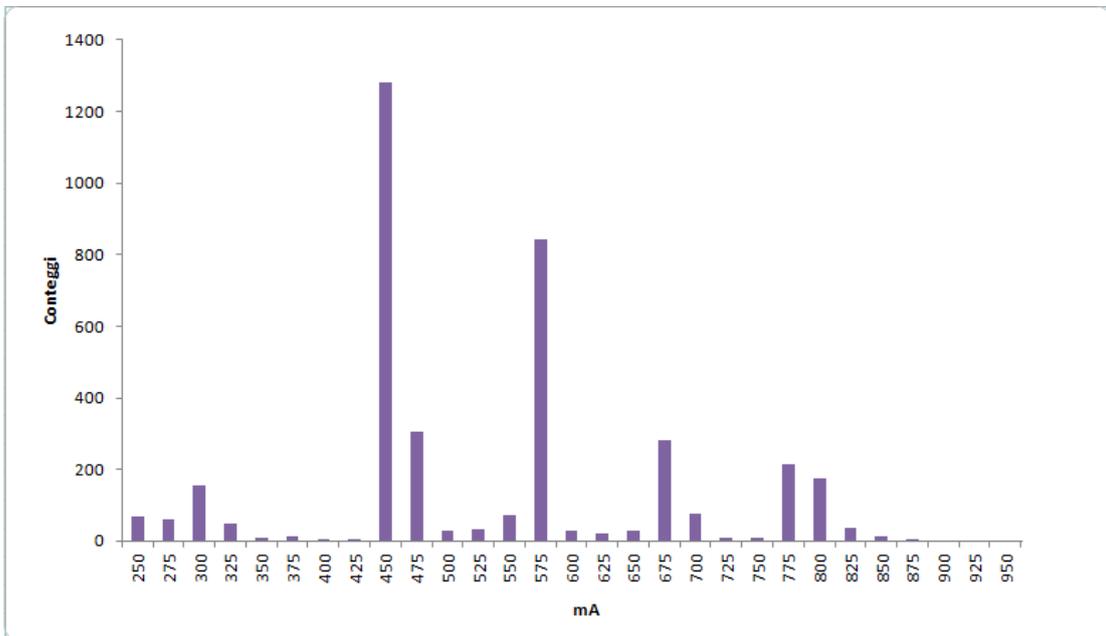


Figura 4.6: *Istogramma che descrive la ricorrenza delle intensità misurate durante l'arco temporale d'interesse.*

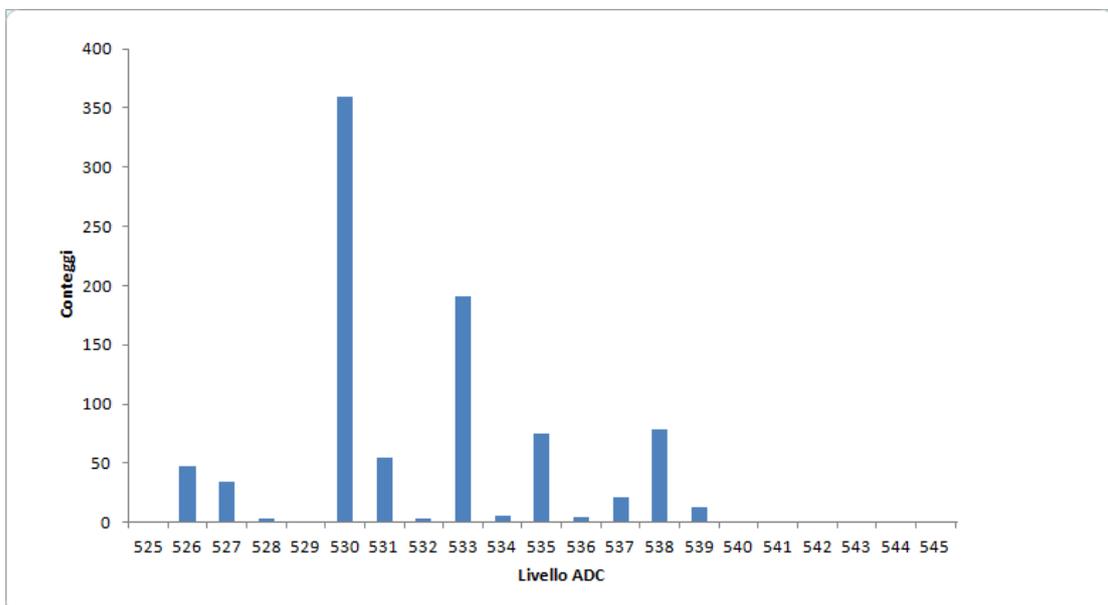


Figura 4.7: *Istogramma che descrive la ricorrenza dei livelli dell'ADC durante l'arco temporale d'interesse.*

Si riportano in tabella i punti utilizzati nella costruzione della retta di calibrazione corrispondenti ai picchi dell'istogramma :

	Intensita (mA)	livello ADC
1 picco 1core	450	530
2 picco 1core	475	531
picco 2core	575	533
picco 3core	675	535
picco 4core	775	538

Rappresentando i valori ottenuti come punti nel piano corrente-tensione, dove in ordinata è rappresentato il livello dell'ADC e in ascissa l'intensità di corrente, è stato possibile ottenere la retta di calibrazione mostrata in figura 4.8.

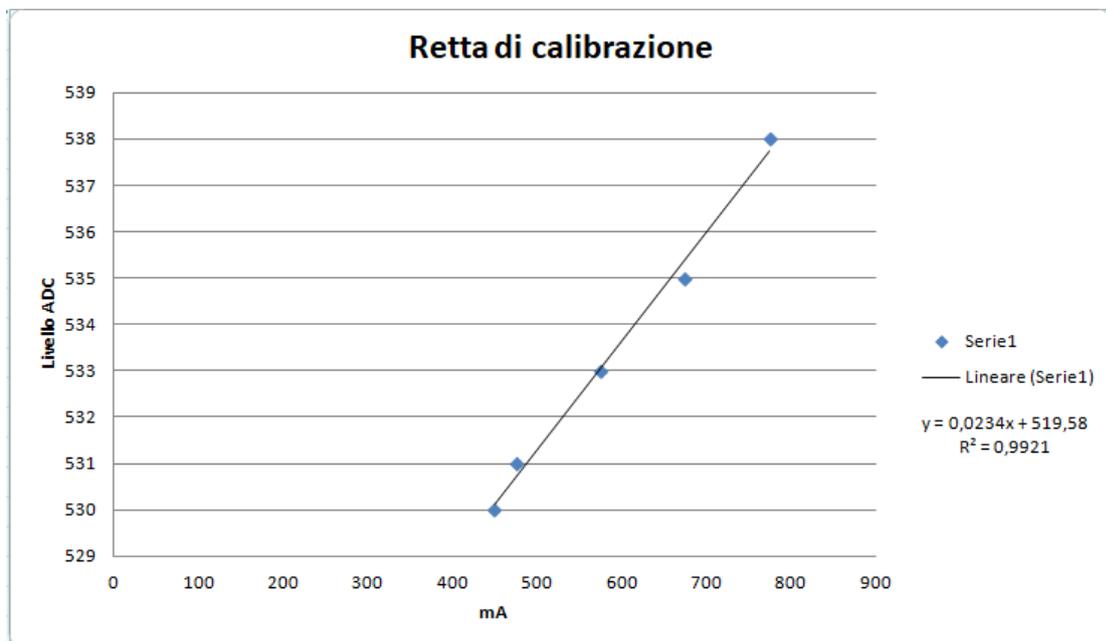


Figura 4.8: Retta di calibrazione ottenuta rappresentando i picchi dell'istogramma come punti nel piano.

Eseguendo, tramite Excel, il fit lineare dei punti rappresentati, la retta che lega intensità assorbita dalla Tegra K1 e tensione in uscita dal trasduttore, risulta avere equazione:

$$y = 0,0234x + 519,18$$

dove si ribadisce che: y corrisponde al livello dell'ADC misurato e x all'intensità di corrente espressa in mA. Per validare il risultato della retta di calibrazione ottenuta, si è eseguito il test del coefficiente di determinazione, o più comunemente R^2 ,

che nel caso delle rette di regressione è il quadrato del coefficiente di correlazione fra due grandezze (corrente e livello dell'ADC in questo caso). Tale coefficiente, indica quale frazione di varianza totale è dovuta alla dipendenza lineare fra due grandezze, ovvero quanto il modello della regressione lineare è attinente al fenomeno in studio. Per questo motivo viene utilizzato per stabilire la bontà di un modello lineare: tanto è più prossimo a 1, migliore sarà la bontà del fit. Avendo ottenuto come risultato 0,9921, che è molto vicino all'unità, la regressione lineare esprime, nel range dei dati sperimentali, molto accuratamente il legame fra le due grandezze, per cui può essere utilizzata per effettuare misure. Grazie a tale retta si possono effettuare le misure di corrente, conoscendo soltanto il livello dell'ADC, senza più l'utilizzo del multimetro.

4.6 Misure effettuate con il sistema realizzato

Per effettuare alcune misure con il sistema realizzato, si sono eseguite tre applicazioni fornite dal CNAF, da eseguire sulla Tegra K1. La prima applicazione, vista in precedenza, conta quanti numeri primi ci sono tra i primi dieci milioni di numeri interi. La seconda, ricostruisce una slice tomografica utilizzando l'algoritmo filtered back-projection (FBP) [23], la terza applicazione si trova in letteratura: stress [24], la quale impone determinati compiti da svolgere alla Tegra K1, anche se in questo caso non si evincono scostamenti significativi dal fondo/vuoto durante l'esecuzione di quest'ultima. Lo script con il quale si è effettuata l'esecuzione è stato impostato con uno `sleep` di 30 secondi fra il termine di un'applicazione e l'avvio della successiva, per consentire di distinguere il vuoto/fondo dal carico della CPU durante l'esecuzione. All'avvio di queste applicazioni, l'Arduino inizia in automatico a creare un file di testo in memoria dove scrive i livelli dell'ADC registrati, al quale è stato possibile accedervi da remoto, copiarlo sul PC, per poi successivamente analizzarne i dati tramite Excel.

A questo punto, per conoscere l'intensità di corrente, espressa in mA, assorbita dall'alimentazione durante l'esecuzione delle applicazioni, si è invertita la retta di calibrazione:

$$I = \frac{ADC_i - 519,27}{0,0234}$$

dove le ADC_i rappresentano tutte le misure del livello dell'ADC presenti nel file di testo.

Per graficare i 64536 punti presenti nel file di testo, si sono dovuti costruire tre grafici (per il motivo citato prima), rispettivamente per l'ADC e per le misure di corrente ottenute, corrispondenti a sequenze temporali consecutive. I grafici appaiono molto rumorosi a causa dell'alta risoluzione temporale. Si evidenziano sempre i quattro plateau per la prima applicazione, corrispondenti al crescente numero dei core utilizzati dalla Tegra K1 (figura 4.10 grafico1 e grafico2). Al termine di questo processo si vede un un fondo/vuoto a circa 0,25 A, corrispondente allo **sleep** (grafico 2 figura 4.10). Successivamente a questo **sleep** si nota un incremento della corrente, dovuto all'esecuzione della seconda applicazione, corrispondente a un plateau fra 0,6 e 0,7 A. Al termine di questa applicazione, si può notare un altro **sleep** dopo il quale si vedono 3 picchi non oltre 0,5 A dovuti alla terza applicazione. Nel grafico 4.11 in rosso, che rappresenta l'ultima sessione temporale delle misure, si ha sostanzialmente fondo/vuoto fra 0,2 e 0,25 A.

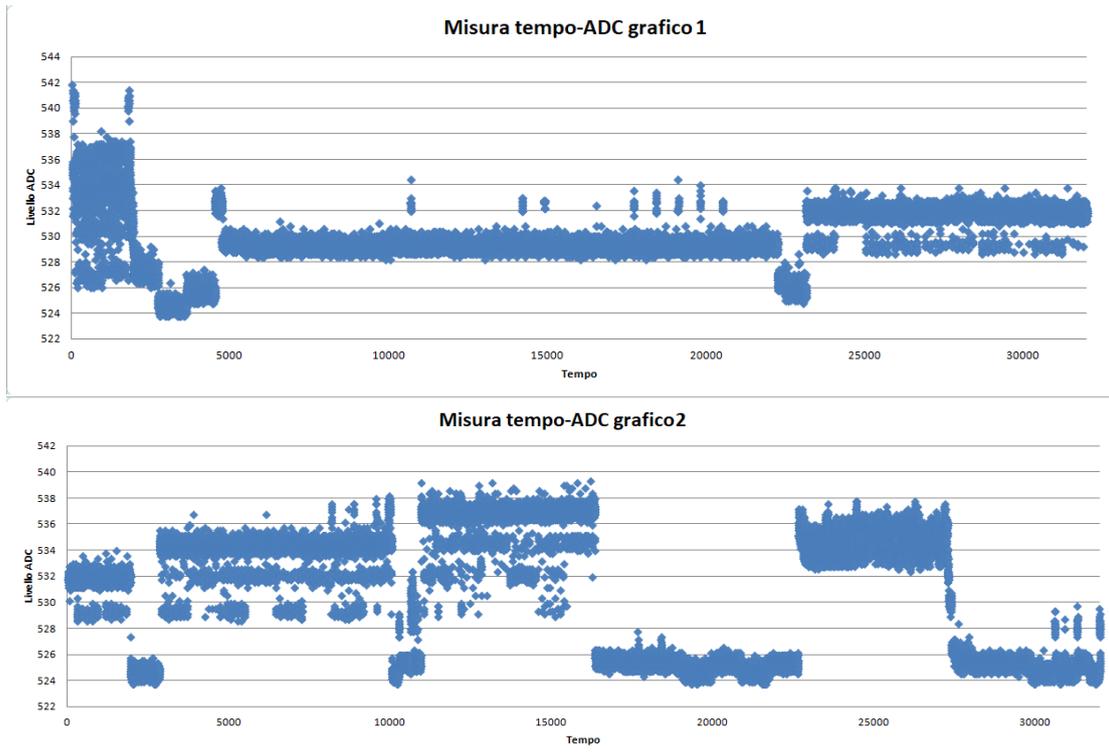


Figura 4.9: *Misure dell'ADC effettuate.*

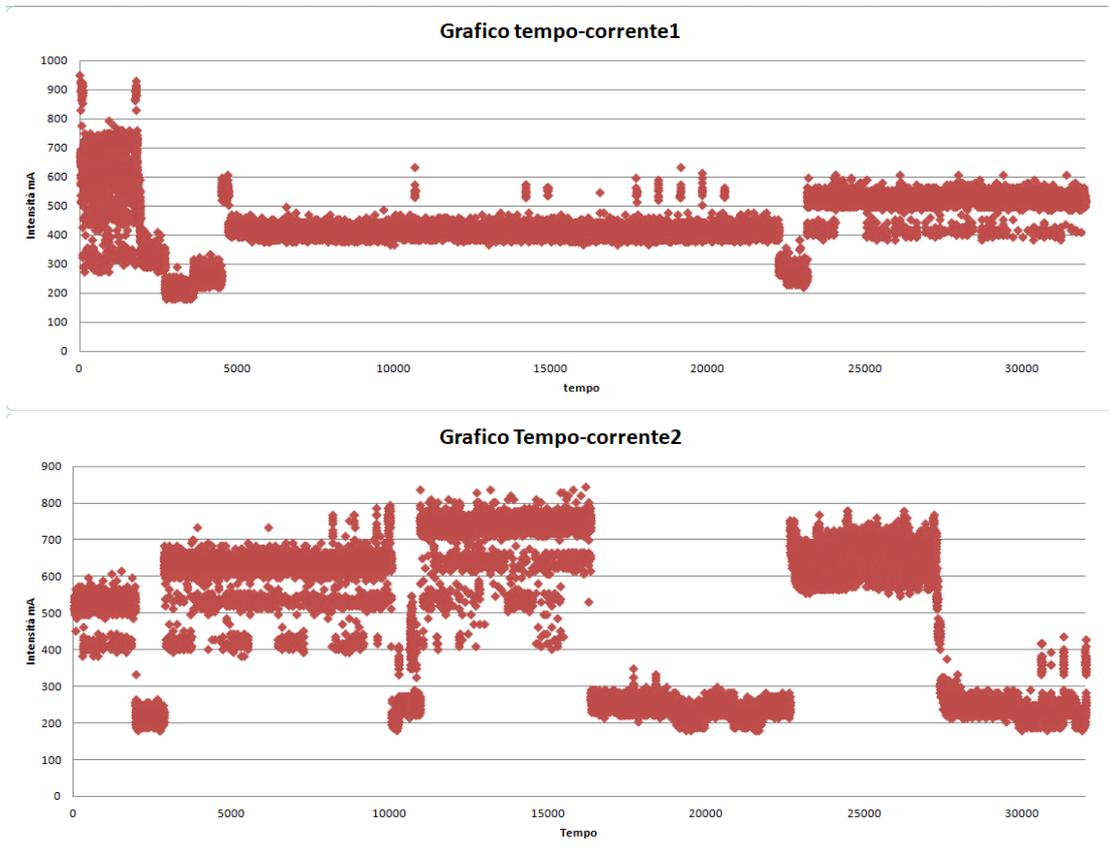


Figura 4.10: *Misure di corrente ottenute invertendo la retta di calibrazione.*

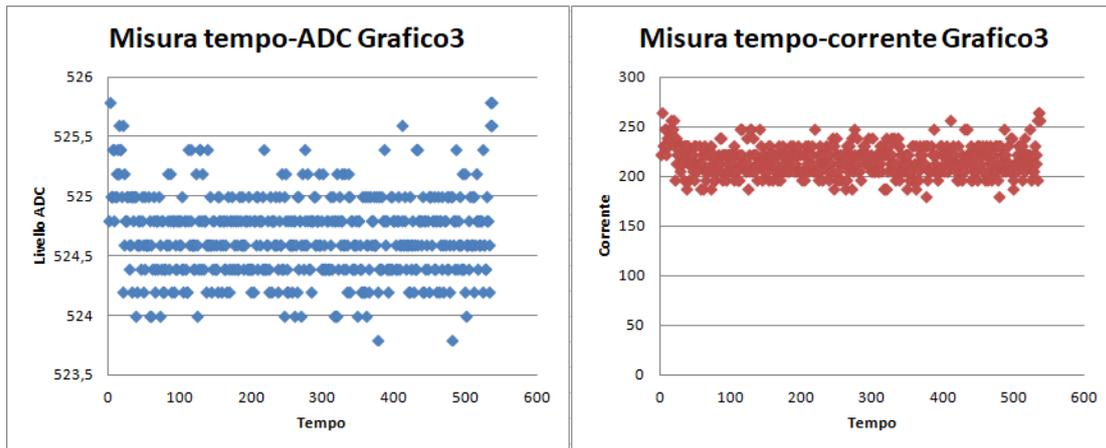


Figura 4.11: *Ultima sezione temporale dei grafici sia dell'ADC e relativa misura di corrente.*

4.7 Confronto

Per avere un'ulteriore verifica dell'affidabilità e della precisione del sistema realizzato, si sono effettuate due misure contemporaneamente: sia dal multimetro ad alta precisione che dal sistema realizzato. Per costruire i rispettivi grafici si è operato come descritto in maniera precedente: analizzando i dati in uscita dal multimetro scritti in un file di testo su chiavetta USB e copiando sul PC il file di testo scritto in memoria dell'Arduino, dal quale si sono ricavati i corrispondenti valori in corrente, con il procedimento descritto nel paragrafo precedente. Dalle misure sono emersi i seguenti risultati: Grafici figura 4.12-4.13 a pagina seguente. Si evidenziano sempre i quattro plateau di intensità crescente, corrispondenti all'uso di uno, fino a tutti e quattro i core della CPU. Al termine di questo processo si nota lo **sleep**, dopo il quale si nota un incremento della corrente misurata su un secondo plateau, corrispondente all'esecuzione della seconda applicazione. Quello che appare evidente dalle due figure è, come nelle ultime misure, da parte del sistema realizzato, c'è una sovrastima sistematica della corrente misurata. Nelle misure effettuate dal multimetro la corrente non supera mai 0,8 A, mentre per quelle ottenute invertendo la retta di calibrazione si arriva anche a 1,2 A. Per quanto riguarda l'andamento globale è pressochè identico, se non fosse per il fatto che le misure ottenute tramite la retta di calibrazione hanno una risoluzione temporale ben superiore rispetto a quelle registrate dal multimetro, per cui appaiono più rumorose.

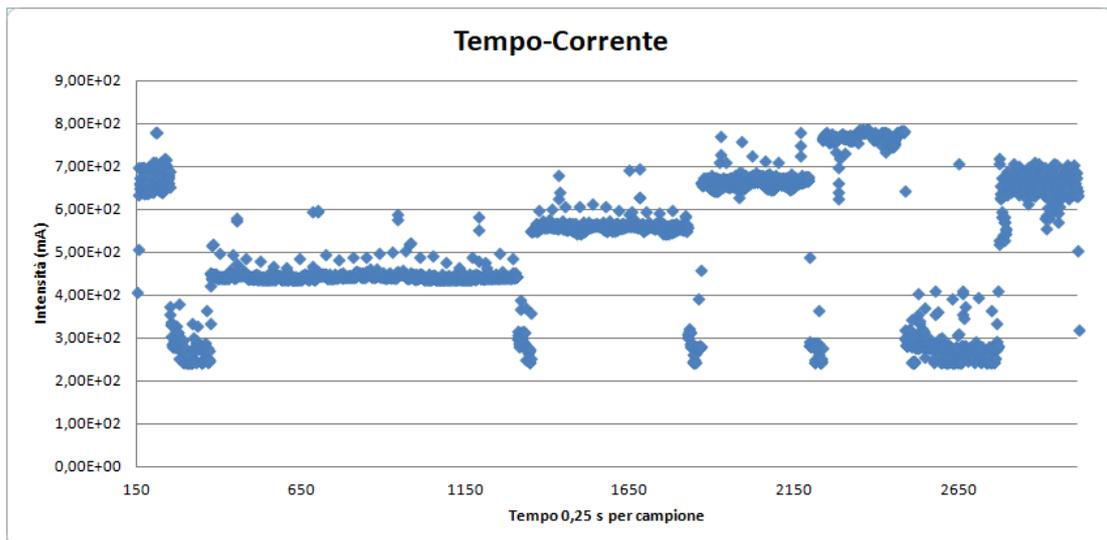


Figura 4.12: *Corrente misurata tramite il multimetro.*

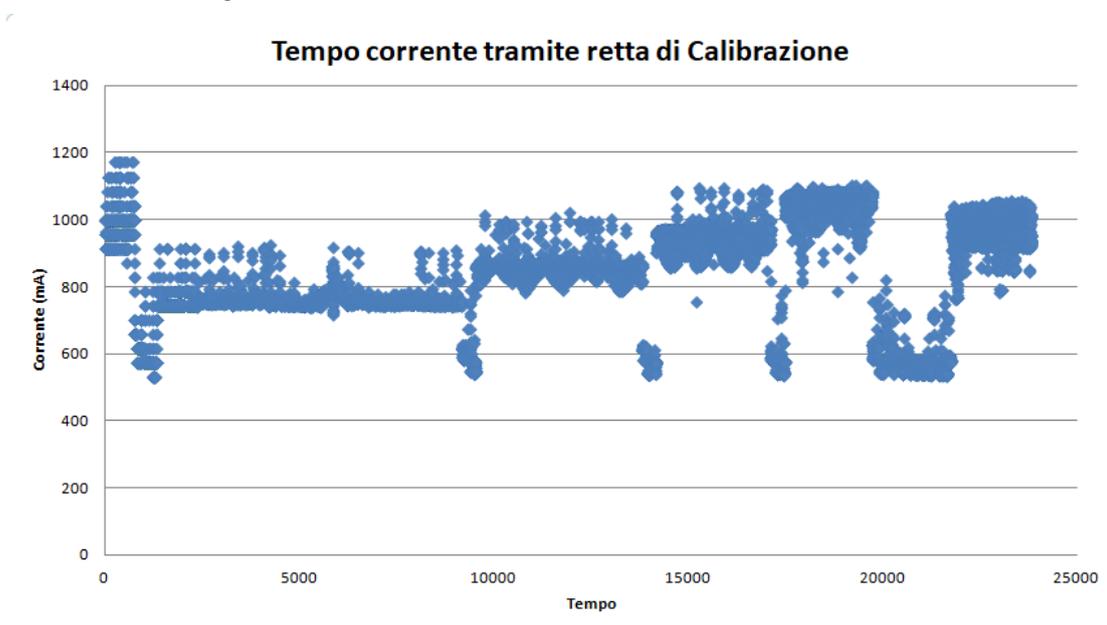


Figura 4.13: *Corrente misurata Tramite retta di Calibrazione.*

Tuttavia, da un confronto fra i grafici in figura 4.14 e 4.15, si nota come le misure relative al livello dell'ADC in figura 4.14 (le ultime effettuate), siano sistematicamente più alte rispetto alle misure effettuate in precedenza (con la stessa applicazione in esecuzione che contava i numeri primi fra i primi dieci milioni). Da un confronto fra le figure 4.5 (misura dell'ADC all'inizio) e 4.9 (misure effettuate solamente col sistema realizzato), si nota che i livelli dell'ADC sono sempre compresi fra 524 e 542; mentre, misurando sia dal multimetro che dal sistema realizzato, i valori dell'ADC salgono fra 530 e 548. Cosa che non avviene invece nelle figure 4.9

e 4.10 (dove si utilizza soltanto il sistema realizzato), in cui i valori dell'ADC sono tutti compresi fra 522 e 544, con conseguenti valori d'intensità compresi fra 0,2 A che rappresenta il fondo/vuoto e 0,8 A a carico massimo. Lo stesso andamento lo si può notare nelle figure 4.3 (misura della corrente effettuata solamente dal multimetro) e 4.10 (Misura effettuata solo con il sistema realizzato).

In definitiva, utilizzando solamente il sistema realizzato, le misure coincidono con quelle effettuate separatamente dal multimetro durante l'esecuzione della stessa applicazione di riferimento: figure 4.3 e 4.10, in cui la corrente rimane sempre fra 0,2 e 0,8 A. Utilizzando sia Multimetro che sistema realizzato, si nota una discrepanza di almeno 0,2 A in eccesso da parte di quest'ultimo.

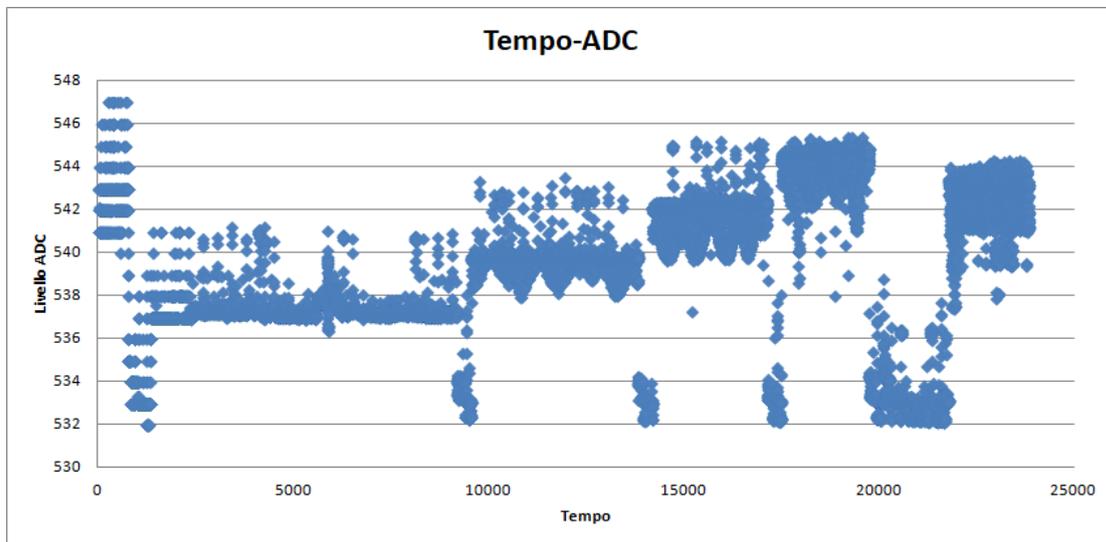


Figura 4.14: *Ultime misure dell'ADC prese nell'ultimo test.*

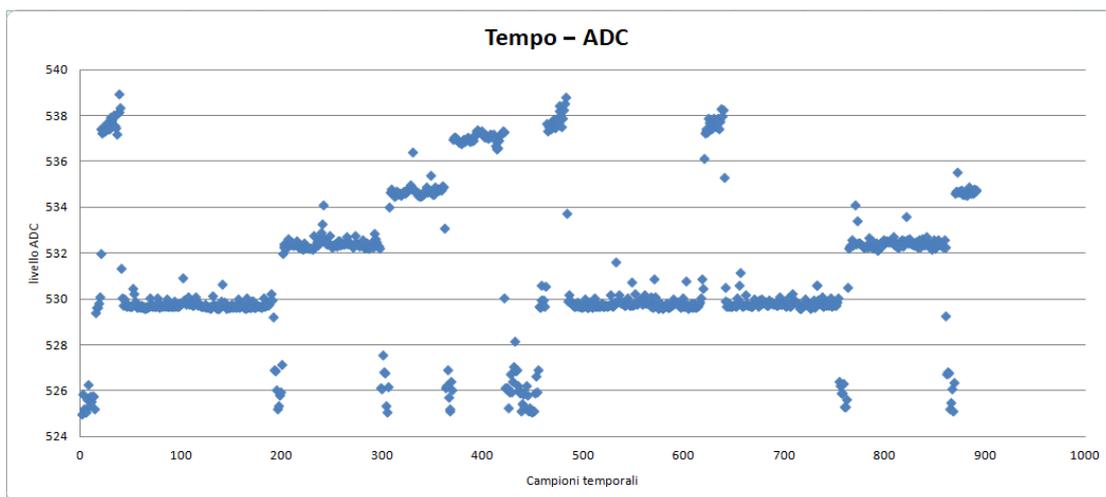


Figura 4.15: *Misure dell'ADC prese in precedenza.*

Capitolo 5

Conclusioni

L'obiettivo del lavoro di tesi è stato quello di realizzare un sistema di misura di corrente, con hardware dedicato, affidabile, ad alta risoluzione temporale e comandabile da remoto. Tale sistema permette, mediante la retta di calibrazione, di effettuare misure durante l'esecuzione di applicazioni su dispositivi low-power.

Come primo step per la realizzazione della retta si è eseguita un'applicazione sulla Tegra K1 che utilizzava da uno fino a tutti e quattro i core in sua dotazione. La corrente veniva misurata all'uscita dell'alimentazione del SoC tramite un multimetro, mentre il trasduttore corrente-tensione, posto anch'esso all'uscita dell'alimentatore, provvedeva a restituire un valore in tensione secondo una funzione lineare, la cui uscita analogica, posta in uno dei pin d'ingresso dell'Arduino, veniva convertita in digitale. Queste misure hanno portato alla realizzazione di due grafici in funzione del tempo, rispettivamente per la corrente e per l'ADC al variare del carico della CPU della Tegra K1. Andando a rappresentare graficamente, mediante l'utilizzo di istogrammi, i valori di maggior ricorrenza temporale, ovvero i quattro *Plateau*, osservati sia per le misure in corrente, sia per il livello dell'ADC si sono ottenuti due istogrammi. Graficando i valori dei cinque picchi dei rispettivi istogrammi nel piano corrente-tensione ed eseguendo un fit lineare dei dati, si è ottenuta la retta di calibrazione, validata dal test R^2 che ha restituito come valore 0,992, quindi esprime molto accuratamente, nel range dei dati sperimentali, la regressione lineare fra corrente misurata dal multimetro e il livello dell'ADC.

Una volta convalidata la retta si sono eseguite tre applicazioni software dedicate,

una delle quali era la stessa utilizzata nel primo step. Questa volta si sono effettuate le misure solo del livello dell'ADC misurato tramite l'Arduino e sostituendo i valori digitali della tensione nella retta di calibrazione, si ottenevano i valori di corrente corrispondenti.

Si è verificato che, mediante l'uso del multimetro o del sistema di misura realizzato durante questo lavoro di tesi, i risultati erano compatibili con l'esecuzione dell'applicazione software di riferimento. Tuttavia, misurando con entrambi i sistemi contemporaneamente, si sono ottenute delle discrepanze nella lettura della corrente non ancora del tutto giustificabili.

Da un'analisi più approfondita dei risultati, ci si è resi conto che le misure del livello dell'ADC, presenti nel file di testo, presentavano dei valori decimali, grazie ai quali è stato possibile mantenere le misure coerenti fra di loro. Nei grafici riportati, quindi, a ogni livello dell'ADC misurato corrisponde un'incertezza di 5 mV, ma al livello reale corrisponderà necessariamente un'incertezza minore, in quanto le grandezze digitali devono essere necessariamente intere. Per conoscere il livello di ADC reale, si dovrebbe effettuare un'opportuna conversione che probabilmente giustificerebbe anche la discrepanza nella misura finale. Un possibile proseguimento futuro di questo lavoro, sarà quello di caricare il sistema on-line nella rete del CNAF, renderlo leggibile e comandabile tramite remoto per effettuare misure su tutti i dispositivi connessi alla rete.

Ringraziamenti

Persone da ringraziare: Elena Corni per la disponibilità, gentilezza e soprattutto pazienza mostrata nei miei confronti; Carmelo Pellegrino, che grazie alla sua ironia ha reso piacevoli le ore di laboratorio. Daniele Cesini e Alessandro Gabrielli per la disponibilità e fiducia mostrata nei miei confronti e per avermi permesso di realizzare un progetto di tesi che, seppur breve, data l'imminente scadenza dei termini; ha portato a dei risultati. Ing. Salvatore Bruccoleri per dei consigli tecnici suggeriti durante la stesura dell'elaborato.

Bibliografia

- [1] http://www.semanticwikisun.sdf.unige.it/mediawiki/index.php/Architettura_di_von_Neumann
- [2] I. Mudawar. “*Assessment of High-Heat-Flux Thermal Management Schemes*”; IEEE Trans. Comp., (2001).
- [3] L. B. Kish. “*End of Moore’s Law: Thermal (Noise) Death of Integration in Micro and Nano Electronics* ”; Phys. Lett., (2002)
- [4] V. V. Zhirnov “*Limits to Binary Logic Switch Scaling - A Gedanken Model*”; Proc. IEEE 91, (2003).
- [5] https://upload.wikimedia.org/wikipedia/commons/0/00/Transistor_Count_and_Moore%27s_Law.-_2011.svg
- [6] http://www.diee.unica.it/barbaro/eolab2/ESD/07/03ed_cmos.pdf
- [7] Santa Clara SA. “*Nvidia Names Stanford’s Bill Dally Chief Scientist, VP Of Research*”; Press release, (2009).
- [8] Samuel H. Fuller and Lynette I. Millett. “*The Future of Computing Performance*”; The National Accademy Press, (2011)
- [9] <http://www.cosa-project.it>
- [10] Power-Efficient Computing from the COSA project
- [11] <http://www.nvidia.com/object/what-is-gpu-computing.html>
- [12] <https://jmmxtech.wordpress.com/2014/06/11/understanding-the-gpu>

- [13] Andrew Stuart Tanenbaum, *“Livello del linguaggio assemblativo, in Architettura dei calcolatori. Un approccio strutturale”*; Pearson Education, (2006).
- [14] E.Calore, S.F. Schifano R.Tripicciono; *Energy to solution towards energy-aware HPC application*
- [15] E.Calore. *Energy-Performance Tradeoffs for HPC Applications on Low Power Processors*
- [16] Datasheet del trasduttore.
- [17] R. H. Warring, S. Gibilisco *“Trasduttori, principi e applicazioni”*; Jackson, (1985)
- [18] M. M. Mano, C. R. Kime *“Reti Logiche”*; Pearson Prentice Hall, (2008)
- [19] Slides del corso: *“Laboratorio di progettazione elettronica”*, prof. A. Gabrielli (A.A. 2015/16)
- [20] https://www.researchgate.net/figure/fig-Fig-34-Arduino-Pin-Diagram-298882952_fig6
- [21] http://www.nvidia.it/content/PDF/tegra_white_papers/tegra-K1-whitepaper.pdf
- [22] <http://linuxgizmos.com/com-express-type-6-module-runs-linux-on-tegra-k1/>
- [23] E. Corni; *“Implementazione dell’algoritmo Filtered Back-Projection (FBP) per architetture Low-Power di tipo Systems-On-Chip”*, (A.A. 2013/14).
- [24] <https://linux.die.net/man/1/stress>