

Una piattaforma flessibile per la robotica didattica

Giulio Vitale

Laboratorio Permanente Robotica Didattica

ITCS Erasmo da Rotterdam

Via varalli, 24 – 20021 BOLLATE (MI)

E-mail: gvitale@tiscali.it

Presentazione di una piattaforma didattica da impiegare in attività di robotica educativa per le scuole medie superiori. La piattaforma, pur potendo apparire sovra tarata per il livello di una scuola media superiore, si è dimostrata adeguata alle premesse e coerente con i presupposti di flessibilità e di apertura della sua struttura hardware e software. L'ambito applicativo scelto è stato quello della categoria di robot "rescue" che partecipano alla manifestazione Robocup Junior, che anche in Italia si svolge ormai da tre anni. Il sistema di base scelto è un "System on Chip" embedded in una FPGA che, restando sostanzialmente immutato nelle sue parti fisiche, per tre anni è riuscito a ri-modellarsi, plasmandosi su tre prodotti sperimentali diversi, seguendone appieno la loro evoluzione e riutilizzando, comunque, lo stesso hardware.

1. Introduzione

Nelle scuole tecniche, più che nei licei, percepiamo in modo più evidente lo stato diffuso di scarsa disponibilità dei nostri giovani allo studio e al sacrificio necessario ad avvicinarsi alle discipline tecnico-scientifiche. Pertanto, da tempo avvertiamo l'esigenza improrogabile di sperimentare e adottare nuove metodologie didattiche, capaci di risvegliare l'interesse verso la cultura, in generale, e verso la scienza e la tecnica, in particolare, con una rinnovata fiducia verso la scuola pubblica.

L'esperienza della robotica educativa si è dimostrata, nella nostra scuola, una delle possibili strade da percorrere per ribaltare un rapporto poco produttivo tra docente e studenti e trasformarlo in una opportunità di collaborazione, che stimoli la fantasia, la passione e la libera partecipazione. In questo contesto si realizzano più facilmente le condizioni per riscoprire la voglia di fare, lo spirito di sacrificio, la tenacia e la perseveranza indispensabili a sostenere lo sforzo necessario sia alla risoluzione dei problemi, sia al superamento di egoismi, stress da competizione o dissidi, per poter, finalmente, lavorare proficuamente in team.

Ma è dal punto di vista professionale che ho potuto osservare la crescita più evidente dei miei studenti, da numerosi punti di vista.

Per prima cosa essi scoprono che ogni progetto, algoritmo o soluzione adottata, non si trasforma automaticamente in un manufatto funzionante secondo le regole prestabilite. Gli oggetti prodotti, hardware o software che siano, esitano a comportarsi nella realtà così come sono stati pensati in teoria, e il pensiero, senza l'esperienza, non possiede immediatamente la chiave del successo. Si accorgono che l'esperienza matura proprio nello sforzo verso la concretizzazione delle idee, lottando con gli imprevisti derivanti dall'interazione con l'ambiente circostante.

Inoltre, imparano a mettere a frutto le diverse conoscenze, maturate in aule diverse e con vari docenti; scoprono nuovi collegamenti e la possibilità insospettata di mettere in pratica le varie tecniche e metodologie apprese, spesso, in modo non del tutto cosciente.

Il docente si trasforma da sorgente d'informazione destinata a uno "studente-contenitore" nel quale riversare conoscenze acquisite, a fornitore di servizi cognitivi verso un utente motivato, che chiede soluzioni a problemi concreti.

Insomma, lo studente cessa per un attimo di essere tale e s'immerge in un contesto molto più vicino alla realtà produttiva che lo attende fuori dalla scuola.

Infine, ma non per ultima, la gratificazione: per gli studenti nel vedere che il successo del proprio lavoro non è più semplicemente premiato con un giudizio ma con la soddisfazione derivante dall'aver vinto la temporanea complessità di un sistema e vederlo comportarsi secondo un proprio modello mentale; per il docente, che da questa soddisfazione trae la piacevole sensazione di aver speso bene le proprie ore d'insegnamento.

2. La piattaforma proposta

In questo contesto all'Erasmo da Rotterdam abbiamo pensato ad una procedura a due fasi nello sviluppo della realizzazione di robot didattici:

- nella prima, gli studenti utilizzano un sistema modulare standard, reperibile sul mercato, con il quale operano dovendo solo comporre parti pre-confezionate senza doverne creare di nuove;

- nella seconda, riprendono e rielaborano la piattaforma sviluppata internamente per farla evolvere rispetto alle esperienze pregresse, ogni anno in un modo diverso dal precedente, cercando di risolvere i problemi riscontrati.

Gli studenti che frequentano il Laboratorio di Robotica Didattica, infatti, hanno una provenienza longitudinale, convivono, cioè, studenti di classi ed età differenti. I più giovani, iniziano con il diffuso sistema Lego Mindstorm, programmato in Java, che li mette subito nelle condizioni di ottenere risultati concreti e soddisfacenti, ma, contemporaneamente, vivono da vicino l'esperienza dei più grandi che si stanno cimentando in parallelo con un robot totalmente auto-costruito, programmabile in C, e che devono risolvere i problemi emersi in precedenza dal punto di vista meccanico, elettronico e informatico, per farlo migliorare ed evolvere.

2.1 Presupposti

Un robot didattico si propone come un sistema relativamente complesso, nel quale convivono problemi di organizzazione generale con problemi specifici relativi ai singoli “task” associati a ognuno dei suoi organi.

Le funzioni caratterizzanti di una piattaforma flessibile, adatte a fornire un ambiente adeguato ad affrontare questa dualità di obiettivi, sono le seguenti:

- **essere totalmente riconfigurabile**, sia dal punto di vista hardware che da quello software, per dare il massimo di libertà nella scelta della soluzione da usare e garantire la possibilità di riciclaggio dell’hardware di base della piattaforma verso altre applicazioni;
- **fornire prestazioni soddisfacenti**, in contesti diversi, pur mantenendo costi contenuti;
- poter **aggiungere periferiche** senza modificare nella sostanza la struttura del sistema.

Una possibile implementazione, legata all’esperienza maturata internamente nel corso di Controlli Digitali, è quella di usare una FPGA (Field Programmable Gate Array) economica, come la famiglia Spartan della Xilinx, all’interno della quale realizzare l’intero sistema hardware: CPU, memorie e controller delle periferiche necessarie a interagire con il mondo esterno.

La flessibilità è garantita proprio dalla possibilità di scegliere la CPU, di ritagliare la quantità di memoria necessaria, di definire e realizzare le interfacce verso l’esterno, sia fruendo di moduli preconfezionati standard, scritti in un qualche linguaggio descrittivo dell’hardware, sia realizzando i propri co-processori esattamente come richiesto dagli organi sensoriali e attuatori scelti.

2.2 La scelta della CPU

In questo contesto, la CPU può essere scelta utilizzando una delle due diverse filosofie d’implementazione:

- **“hard core”**, in questo caso il processore è già fornito come una macro pre-cablata nella FPGA; ad esempio in casa Xilinx è attualmente fornita una CPU PowerPC, ma in futuro potrebbe esserci un core Arm.

“soft core”, in questo ambito sono possibili due ulteriori scelte. La prima, conduce all’impiego di core forniti dal costruttore; la seconda, porta a scegliere “cores open source”, disponibili con la stessa modalità di condivisione del “software open source” (vedi sito <http://opencores.org>).

Nel nostro caso abbiamo scelto la soluzione fornita da Xilinx, ovvero il core RISC a 32 bit con un’architettura Harvard, denominato Microblaze (fig.1) e dotato di un elevato grado di configurabilità e di ritagliabilità delle singole funzioni, in modo da renderlo perfettamente calzante alle esigenze. Il motivo di tale scelta è legato alla disponibilità di un sistema di sviluppo completo e integrato, che Xilinx ci ha messo a disposizione gratuitamente, all’interno del quale condurre il flusso completo di progetto, sia hardware che software, fruendo di strumenti, quali wizard assistiti, ricche librerie e quant’altro necessario a un rapida e semplificata messa a punto del sistema di elaborazione centrale.

Nel mondo “open cores”, comunque, sono disponibili moltissimi cloni di processori standard, così come risultano avviati tanti progetti aperti di CPU didattiche, che emulano diverse architetture molto diffuse nei sistemi di elaborazione commerciali, nelle quali il “do it yourself” raggiunge il massimo livello ma richiede più attenzione e competenza.

Dal punto di vista didattico questo è il livello supremo di libertà; non ci sono vincoli precostituiti all'architettura da scegliere e all'indirizzo didattico da dare al proprio corso, ma soltanto considerazioni di opportunità nella scelta tecnologica da fare.

2.2 Il progetto delle interfacce

Nel caso della nostra categoria di robot, i “rescue”, ovvero robot soccorritori, devono necessariamente essere dotati di una vasta gamma di sensori e di attuatori per svolgere appieno alla loro funzione. Quindi anche un semplice robot didattico può porre sfide progettuali interessanti.

Per questa categoria di robot, è stata individuata la necessità di poter disporre dei seguenti moduli di comunicazione:

- sensori di luminosità per seguire una traccia prestabilita; più di uno per misure spot singole, almeno uno in caso di sensore lineare o a matrice di punti. Nell'ultimo prototipo realizzato ne sono presenti tre del tipo luce-frequenza.

- Sensori di colore; le vittime da cercare possono essere identificate da un colore, pertanto occorre avere l'abilità di discernere le singole componenti cromatiche.

- Ostacoli vicini; il robot deve essere in grado di rilevare eventuali macerie che possono ostacolare il suo cammino e che risulta più conveniente evitare invece che scavalcare. In questo caso gli infrarossi sembrano la soluzione ottimale, per quanto sostituibili con semplici sensori di prossimità.

- Oggetti lontani; deve essere necessario poter individuare oggetti voluminosi, posti a una certa distanza e che si staccano nettamente dal contesto. Per questa funzione possono rivelarsi utili sensori a ultrasuoni.

- Inclinazione e scavalcamento; può essere utile la presenza di una piattaforma inerziale, con funzioni di giroscopio/accelerometro, per identificare un tragitto in piano o la presenza di una inclinazione o di un dislivello nel territorio.

- Pinza; alcune macerie richiedono la rimozione oppure l'eventuale vittima richiede di essere sollevata e portata in salvo in un luogo sicuro.

- Motori; devono essere controllati in velocità e deve essere possibile rilevare l'entità di uno spostamento o di una rotazione.

Ogni sensore e ogni attuatore richiede un'interfaccia progettata ad hoc e non è detto che sia la stessa da un anno all'altro.

3. Il modello generale

Dalle precedenti analisi ne deriva un modello generale di base come quello indicato in fig. 2.. Da esso emerge la qualità di un sistema flessibile. I moduli d'interfaccia sono un potpourri di diverse tipologie di interfacciamenti,

corrispondenti a una variegata rappresentanza di organi periferici da controllare, alcuni standard (UART, LED, dip-switch, pulsanti, memorie) altri del tutto disegnati ad hoc. Per i primi si possono scegliere delle IP prefabbricate, offerte dall'ambiente di sviluppo o reperibili sempre come “open cores”, per i secondi si possono creare le proprie “custom IP” in relazione alle scelte fatte di anno in anno e di robot in robot.

Il progetto procede come la confezione di un abito e il sistema che ne deriva evolve verso una ottimale condizione di parallelismo, nel senso che **i dati acquisiti sono elaborati dove serve** e non semplicemente trasferiti in forma digitale verso un'unica unità centrale di elaborazione.

Con poca spesa ma con un buon lavoro didattico si possono raggiungere ragguardevoli prestazioni. Il sistema ottenuto tende a comportarsi come un sistema organico in cui ogni singolo organo periferico non invia dati grezzi al “cervello” ma li pre-elabora, estraendone informazioni più sintetiche e più facilmente trattabili dalla CPU. Esso tende ad assomigliare sempre di più a una struttura “data flow” che a una classica macchina sequenziale.

4. L'implementazione del sottosistema di elaborazione

Il sistema ritagliato all'interno della FPGA assume la forma classica di un'architettura Harvard, con BUS separati non solo tra istruzioni e dati, ma suddivisi su diversi livelli gerarchici, in funzione del grado d'interazione tra software e periferica. Nel caso del Microblaze si può disporre di almeno tre diverse infrastrutture standard di BUS:

- un BUS locale a latenza nulla per il collegamento tra CPU e memoria centrale; la velocità di elaborazione equivale al tipico livello L1 nei sistemi basati su una organizzazione gerarchica della memoria;
- un BUS principale “cacheable” per il collegamento di tutte le periferiche di comunicazione, interne ed esterne, e l'interfaccia verso memorie esterne a maggiore latenza ma con una più elevata capienza;
- un link veloce, FSL (Fast Simplex Link), per connettere co-processor specializzati con i quali far dialogare la CPU a livello di linguaggio macchina; per questo tipo di BUS sono disponibili istruzioni del tipo “put” e “get” che effettuano trasferimenti su registri specializzati con l'esecuzione di una sola istruzione.

La piattaforma che viene qui presentata è visibile nella Fig.3

5. Il livello di coinvolgimento degli studenti

Nelle piattaforme usualmente proposte a livello standard, gli studenti possono intervenire con due modalità di atteggiamento:

- nella prima (es. Lego Mindstorm) essi sono coinvolti nella scelta creativa dell'assemblaggio di parti modulari, con una limitata possibilità d'intervenire nel funzionamento dei vari organi ma con una grande scelta di opzioni prefabbricate da comporre; dal punto di vista

professionale di una scuola tecnica, l'attività principale riguarda il software;

- nella seconda (es. Parallax e simili) gli studenti dispongono di kit più aperti, con moduli preconfezionati disponibili ma con un'ampia possibilità di creare interfacce particolari per moduli personalizzati; sempre dal punto di vista di una scuola tecnica, l'attività si estende dal software all'hardware.

Nel caso specifico della piattaforma qui presentata, l'attività di coinvolgimento è del secondo tipo ma ancora più spinta: essi possono essere coinvolti anche nella definizione della struttura del sistema di elaborazione. Ovviamente la funzione del docente è più importante nella fornitura dei servizi essenziali alla messa a punto del sistema, ma esso nasce "più vicino" ai suoi fruitori e più aperto dei moduli standard disponibili sul mercato, che spesso corrono il rischio di essere acquisiti come scatole chiuse.

Nella mia esperienza di questi tre anni di Laboratorio di Robotica Didattica ho visto nascere e crescere tra i miei studenti sia una maggiore percezione dell'organizzazione interna dei sistemi di elaborazione "real time", sia un maggiore senso di paternità verso il prodotto finito.

Pur non controllando interamente il processo che porta alla completa realizzazione della piattaforma, gli studenti sono comunque più coinvolti e partecipi dell'implementazione di tutti i "task" specifici, eseguiti da ognuno degli organi sensoriali e di attuazione, non fosse altro perché devono scrivere i driver software per la loro gestione, per inserirli agevolmente all'interno della strategia complessiva del sistema, necessaria a concludere l'intero percorso di salvataggio previsto dalle regole internazionali nelle gare di "rescue".

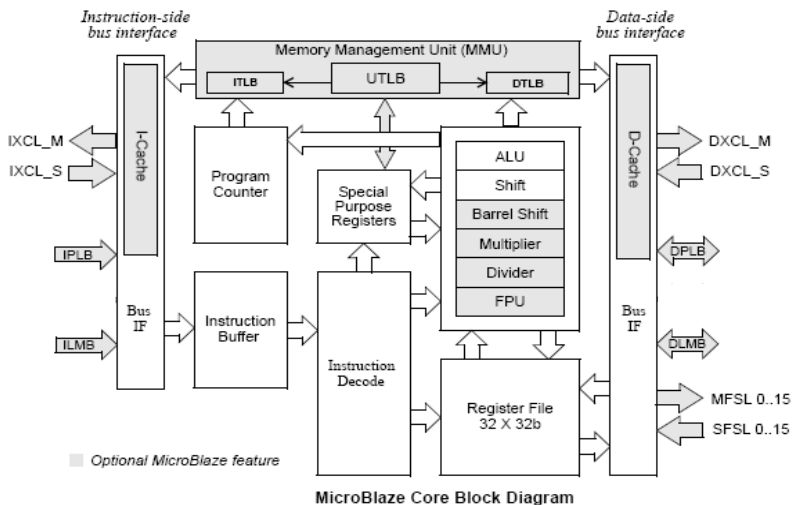


Fig.1 – Architettura del soft core configurabile Microblaze

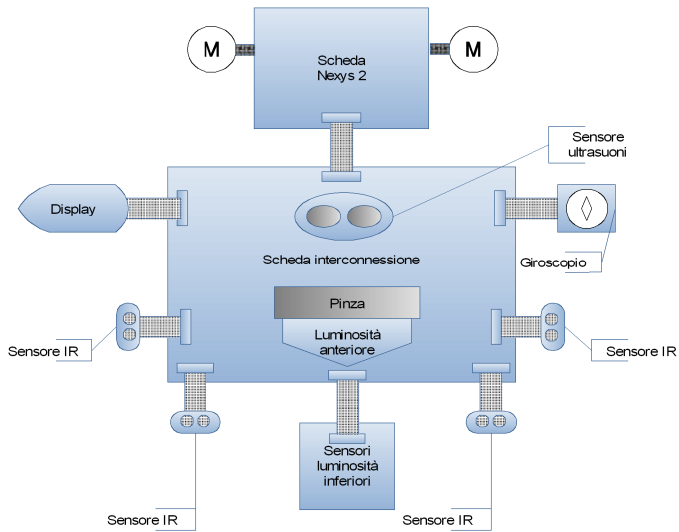


Fig.2 – Schema a blocchi della piattaforma proposta

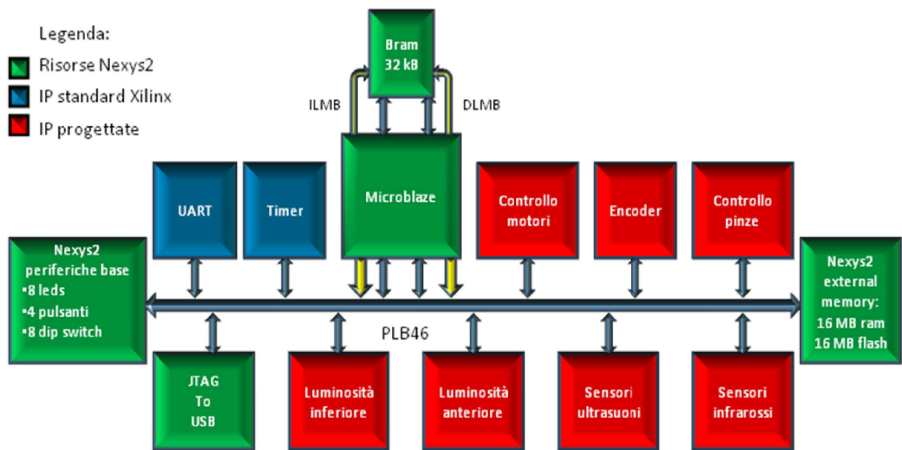


Fig.3 – L'implementazione su FPGA

Conclusioni

La piattaforma di lavoro proposta potrebbe apparire troppo complessa per il livello delle scuole medie superiori ma nei fatti è risultata perfettamente adeguata alle esigenze e, in tre anni consecutivi, ha prodotto quattro robot, simili nella tipologia, ma in forte evoluzione l'uno rispetto all'altro, con l'architettura originaria che si è riadattata di anno in anno ai miglioramenti necessari.

Ciò che noi ci auguriamo è che nasca la possibilità di estendere questa esperienza, mettendola in "Rete", contando sulla eventuale formazione di gruppi di lavoro che scambiano le proprie esperienze e condividono alcune soluzioni tecnologiche di base, lasciando ai singoli laboratori l'arbitrarietà della strategia generale, esattamente come avviene con i kit preconfezionati in cui si assemblano parti standard, disponibili a tutti.

La scuola, deve poter fruire di tutti i vantaggi proposti dalle tecnologie correnti per raggiungere i propri obiettivi formativi; nel fare ciò non deve necessariamente proporre sempre soluzioni facili ma deve cercare d'insegnare anche ad affrontare quelle difficili.

Lo studio delle tecnologie più avanzate, necessarie a risolvere i problemi più complessi, è parte integrante dell' "Insegnare Futuro" e quanto sperimentato nel Laboratorio di Robotica Didattica si presenta come una possibilità concreta per produrre uno scenario adeguato e in grado di promuoverlo.

Naturalmente tutto il lavoro finora svolto nel Laboratorio è stato possibile soltanto perché si è basato sull'impegno volontaristico e passionale di alcuni docenti, con fondi economici e spazi curricolari molto angusti, se non addirittura inesistenti, ma con un grande successo tra gli studenti e le loro famiglie, che ha fatto aumentare il numero di iscritti all'indirizzo Elettronico-Informatico, da quando l'attività laboratoriale è stata resa visibile sul territorio con "microstage di robotica" verso i ragazzi delle scuole medie primarie.

Se, però, queste esperienze non si sedimentano in una pratica curricolare più stabile, capace di estendere il numero di docenti partecipanti e di creare una sorta di continuità didattica, tutto ciò rischia di rimanere un vano tentativo di rinnovamento che si spegnerà con lo spegnersi della presenza degli attuali propositori.