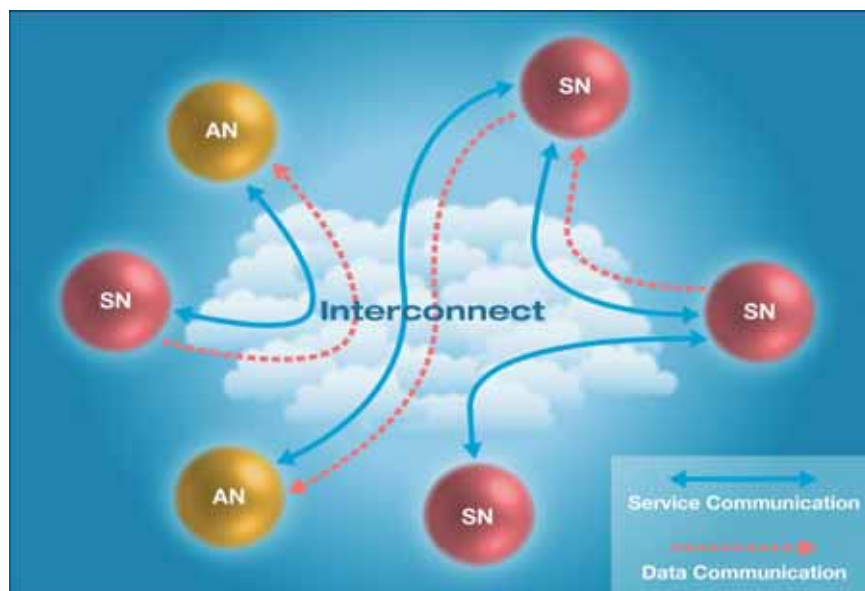


## Metodologia ESL: un esempio pratico di applicazione

**Kai Liu**  
TRIAS Mikroelektronik  
I Parte

Allo scopo di valutare appieno i benefici legati al passaggio a una metodologia basata su ESL, un gruppo di progettisti della divisione IA&DT (Industrial Technology and Drive Technologies) di Siemens, partendo dal progetto di uno switch Ethernet già esistente, ha generato una specifica a livello architetturale eseguibile. Al fine di poter passare dalla definizione del sistema alla fase di esplorazione architetturale è stato utilizzato un toolset ESL di CoFluent Design. Variando la mappatura del modello funzionale nel modello della piattaforma è stato possibile creare un certo numero di architetture adatte all'implementazione del progetto. Nel caso di un progetto complesso come quello di uno switch Ethernet, esistono molte variabili a livello architetturale che possono influenzare costi e prestazioni del sistema. In passato, per effettuare il confronto tra diverse architetture si ricorreva a calcoli e analisi di tabelle Excel, basati in larga misura sul know how acquisito dagli architetti di sistema. Fare affidamento sull'analisi degli spreadsheet e sull'esperienza acquisita può rappresentare un limite al numero delle configurazioni che

*L'impiego di una metodologia basata su ESL ha permesso a un team di progettazione di Siemens di effettuare un confronto tra diverse architetture di uno switch Ethernet 100 Mbps in modo rapido e accurato. In questa prima parte vengono descritte le fasi che hanno portato alla creazione dei modelli architetturali, mentre nella seconda verrà fornita l'analisi dei risultati*

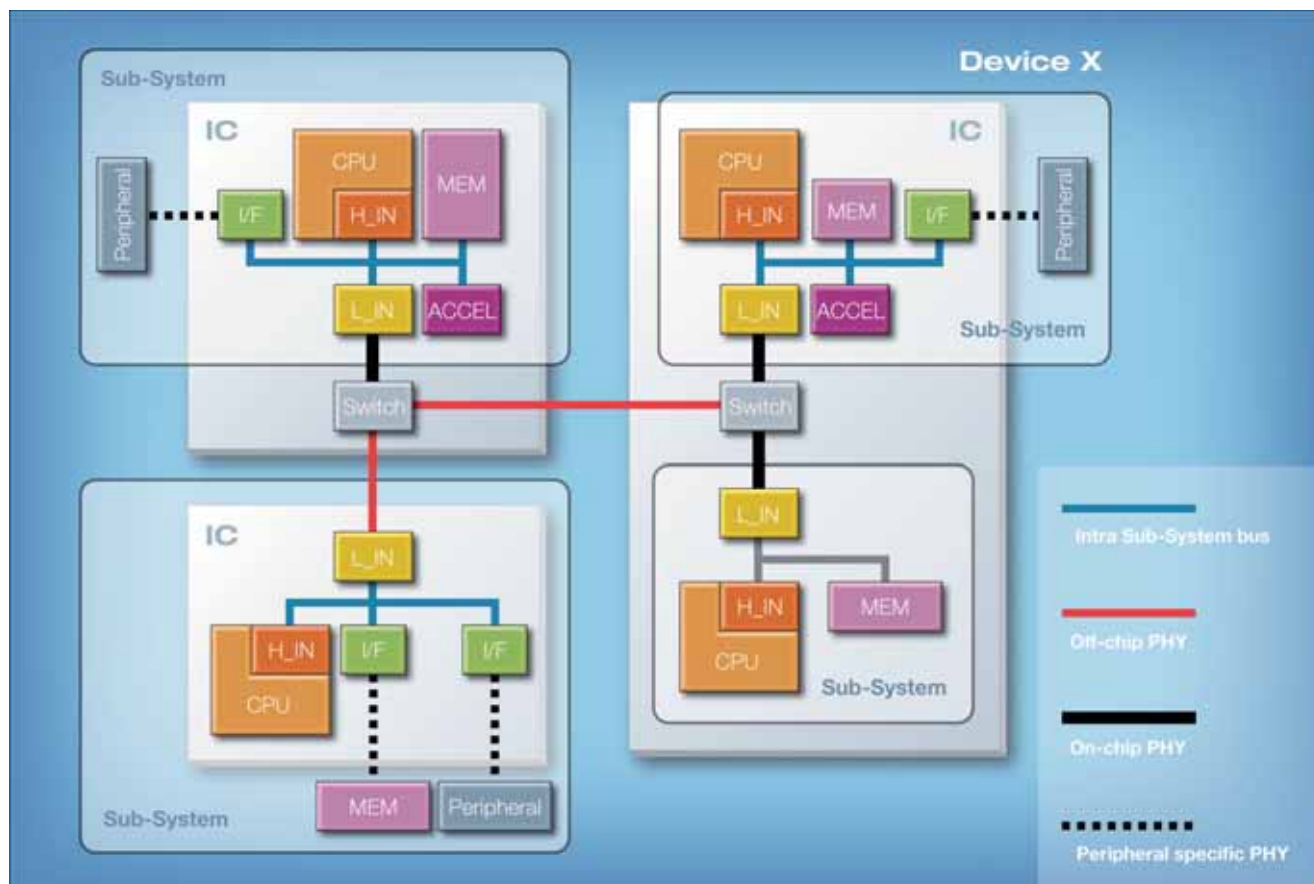


**Fig. 1 – Flusso di progettazione ESL**

è possibile analizzare. L'adozione di un processo automatizzato per le operazioni di generazione e confronto permette di valutare in maniera accurata un numero sicuramente maggiori di casi di test (test cases). L'obiettivo finale è pervenire a una combinazione ottimale di variabili di progetto che permetta di ottenere le migliori prestazioni senza nessun tipo di ridondanza (fattore che si riflette favorevolmente sui costi di progetto).

### Flusso di progettazione

Il flusso di progettazione utilizzato viene riportato in figura 1. Nella prima fase i progettisti di sistema devono prendere in esame i requisiti dei clienti e considerare tutti i possibili casi di utilizzo (use case). La modellazione comportamentale temporizzata si basa sulle specifiche provenienti dalla prima fase prevista dal flusso ESL. In questa fase viene dichiarata la struttura interna e il flusso di trasferimento dei segnali mediante la definizione degli accoppiamenti e delle funzioni necessarie allo scopo. La messa a punto



della piattaforma reale avviene nella fase di modellazione della piattaforma stessa. Per questa modellazione è anche possibile utilizzare il modello strutturale: esso è di tipo grafico e gerarchico. Una piattaforma semplificata per il modello funzionale viene progettata seguendo lo stesso metodo di trasformazione successive: la soluzione cui si perviene è la classica "scatola nera" che contiene l'insieme completo di funzionalità del sistema.

Sono tre gli elementi con i quali è possibile descrivere qualsiasi piattaforma fisica: processori, memorie e nodi di comunicazione. Per poter utilizzare il modello nella fase successiva, ovvero quella della modellazione architetturale, il significato di modello di piattaforma in questo contesto viene ampliato prendendo in considerazione il significato a livello sia funzionale sia di piattaforma di una struttura. Il metodo più comunemente adottato

**Fig. 2 – Rappresentazione grafica del sistema di commutazione**

prevede la mappatura dal modello funzionale in quello della piattaforma. A livello di progettazione architetturale vengono aggiunte ulteriori informazioni al modello comportamentale. Poiché il diagramma delle attività funzionali utilizza la piattaforma, è necessario prevedere l'allocazione e il rilascio di uno spazio di memoria per una memoria specificata. I valori assegnati per l'interfacciamento tra le funzioni e le piattaforme rappresenta la durata temporale. Tutti questi elementi, ovviamente, influenzano le prestazioni. Le prestazioni dell'intero sistema – utilizzo del bus, requisiti di memoria, dissipazione di potenza, costi – sono ricavati attraverso la simulazione.

## Modellazione funzionale

La modellazione funzionale prevede le seguenti operazioni:

- analisi e modellazione dell'ambiente dell'intero sistema;
- definizione dei limiti del sistema con i suoi ingressi e le sue uscite;
- definizione del comportamento del sistema;
- sintesi del comportamento mediante un diagramma di attività e successiva verifica.

Le funzioni previste per lo switch comprendono la trasmissione dei frame, la ricerca della porta di destinazione in base alla tabella degli indirizzi, la memorizzazione dell'indirizzo della sorgente e del numero di porta associato e la valutazione della rete VLAN (Virtual Local Area Network). La fase successiva definisce i confini del sistema e determina gli ingressi e le uscite. A questo stadio di

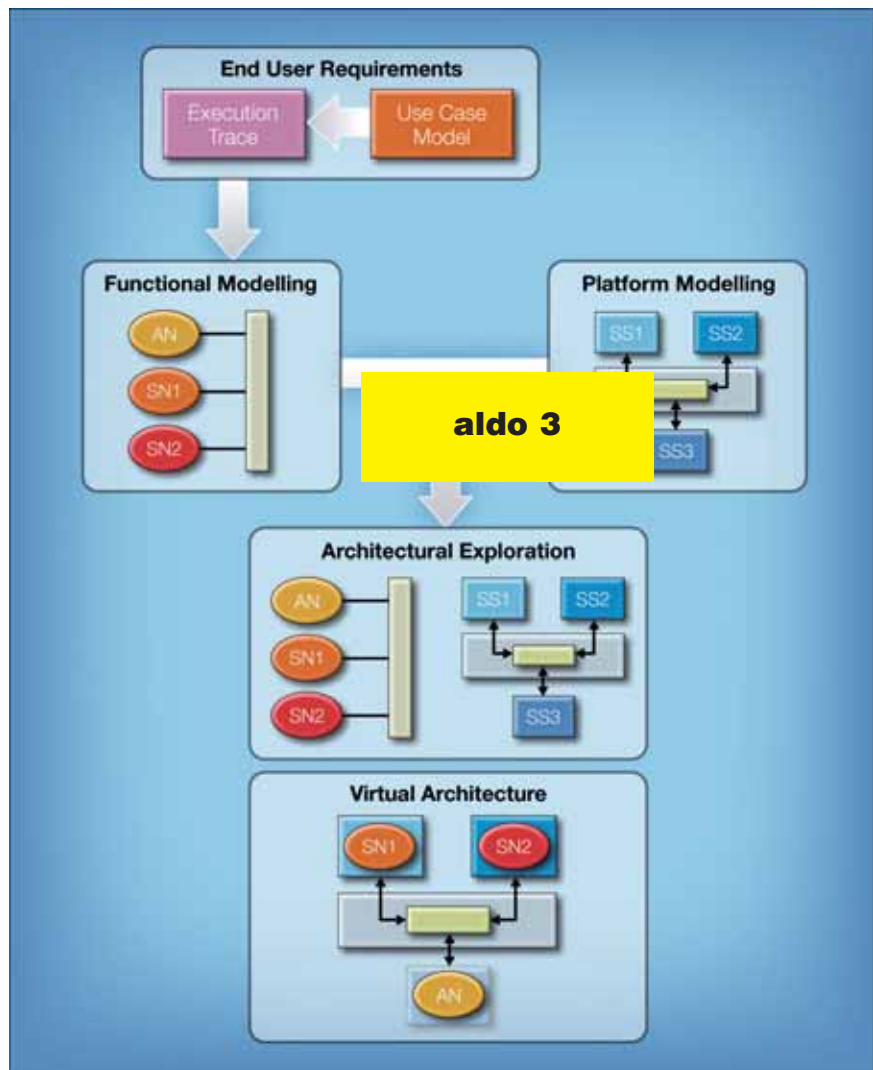
**Fig. 3 – Modello della piattaforma: SimplestPlatform**

progettazione funzionale le specifiche relative alla tecnologia di implementazione non vengono prese in considerazione poiché in questa fase è necessario ottenere come risultato una soluzione indipendente da una particolare tecnologia. La modellazione ambientale prende in considerazione le interfacce a livello di sistema, in particolare modo le porte di ricezione (Rx), quelle di trasmissione (Tx) e l'interfaccia dello switch. Il progetto che si sta prendendo in considerazione è in grado di ospitare quattro porte Rx e altrettante Tx. L'interfaccia dello switch, dal canto suo, si preoccupa della configurazione dello stesso, compresa l'inizializzazione della tabella di indirizzi e della tabella della rete VLAN.

Al fine di realizzare un sistema scalabile, viene specificato un parametro generico denominato PortNo. Mediante i parametri generici è possibile la messa a punto dei parametri del sistema in modo da far girare le simulazioni senza dover procedere alla ricompilazione del modello. Una caratteristica di questo tipo permette di semplificare la misura delle prestazioni e il relativo confronto. La configurazione stabilisce gli attributi per configurare il sistema. Questi attributi sono caratterizzati da un proprio tipo definito in conformità alle funzionalità dello switch. Una volta completata la configurazione, lo switch dà avvio all'istradamento dei frame. Producer ha il compito di generare i frame e di collocarli nelle corrispondenti code di messaggio FrameIn. Quando lo switch termina l'elaborazione dei frame, li colloca nelle corrispondenti code di messaggio Frame Out.

### Comportamento del sistema di commutazione

Una volta definito l'ambiente del sistema è possibile studiare il comportamento dello stesso sotto svariati punti di vista. In primo luogo l'intero sistema viene descritto senza apportare alcuna miglioria. Da un punto di vista esterno, lo switch riceve i frame da un generatore



(Producer) e li invia al ricevitore appropriato. Si tratta come si può vedere di una descrizione algoritmica molto astratta. La valutazione delle prestazioni (benchmarking) è basata su questo sistema astratto e può essere riutilizzata per il perfezionamento del sistema. Essa assicura la correttezza del primo prototipo dell'intero sistema di commutazione così come quello delle migliorie apportate nelle fasi successive.

Una volta verificato il modello astratto, questo viene perfezionato passo dopo passo: nella figura 2 viene rappresentato graficamente il sistema di commutazione finale. Ciascuna istanza di Input Processing e Output Processing dispone del relativo Input Buffer e Output Buffer. La funzione Routing si preoccupa di istradare il frame dall'InputBuffer della porta sorgente all'OutputBuffer della porta di destinazione. SharedBuffer dal canto suo

viene impiegato per la configurazione che richiede una memoria centrale per memorizzare tutti i frame e non prevede il ricorso a Input Buffer o a Output Buffer.

In base alle caratteristiche di traffico di una tipica comunicazione Industrial Ethernet sono stati generati numerosi casi di test (test cases), realizzati usando un file Excel al fine di collaudare tutte le varie tipologie di situazioni di traffico. Il file XML viene generato dal file Excel e quindi letto dal testbench del modello ESL. Nel corso del presente articolo vengono presentati alcuni dei casi di test di utilizzo più frequente.

Il traffico Ethernet si distingue fra traffico burst e traffico base. Il traffico del primo tipo contiene i frame più corti, 64 byte, ed è contraddistinto da un fattore di carico pari al 100%. Il gap all'interno dei frame più corto è di 12 byte. Il traffico

base, invece, contiene frame caratterizzati da lunghezza e fattore di carico variabili. I casi di test mettono in evidenza gli scenari nel caso peggiore (worst-case), come accade quando il traffico burst arriva contemporaneamente a uno switch da parte di tutti gli altri switch della rete, situazione questa particolarmente onerosa per la memoria. Una volta definiti il diagramma funzionale e gli algoritmi per ciascun blocco funzionale, è possibile ricavare parecchie configurazioni. Il toolset ESL consente di configurare le diverse soluzioni in conformità ai requisiti dell'applicazione. Oltre a ciò è possibile inserire differenti algoritmi per le configurazioni. Al fine di ottenere diverse configurazioni allo stesso modello grafico vengono applicati vari schemi di pianificazione dello switch.

### Esplorazione architetturale e analisi delle prestazioni

La piattaforma rappresenta la soluzione fisica presa in considerazione per il progetto al fine di supportare la soluzione funzionale. La ricerca della struttura appropriata richiede la progettazione di un'architettura fisica. La piattaforma risulta composta dai processori e dalle relative interconnessioni, che possono prendere la forma di nodi di comunicazione e memoria condivisa. Un processore è in grado di far girare le funzioni, siano esse hardware oppure software.

Dapprima si descrive la struttura della prima piattaforma che rappresenta i componenti fisici, senza prendere in considerazione nessuna miglioria a livello della piattaforma hardware (Fig. 3). Questa piattaforma astratta consente di effettuare la prima (e più semplice) mappatura dal modello funzionale su quello della piattaforma. Essa garantisce il corretto comportamento sia di Producer e Receiver sia di FrameIn e FrameOut dopo la mappatura. L'attenzione è concentrata sull'architettura dello switch centrale. Al fine di creare strutture di piattaforme

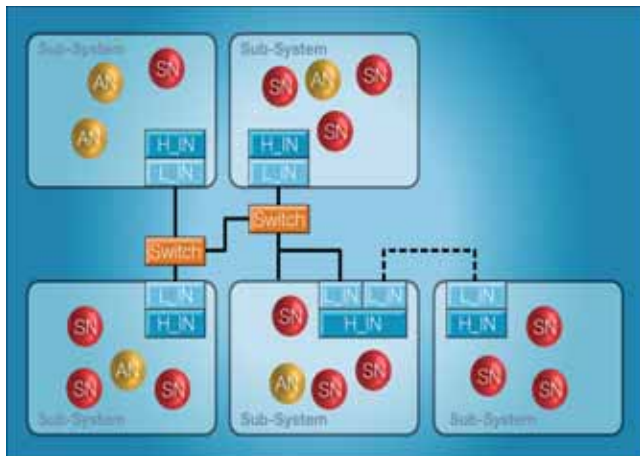
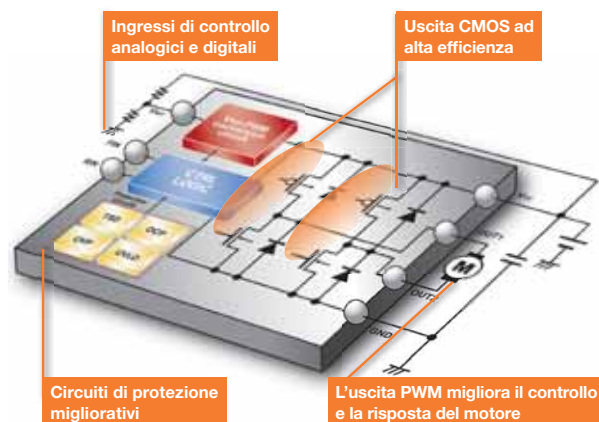


Fig. 4 - Modello della piattaforma: RefinedPlatform 1

## Drivers per motori a ponte H

Alta efficienza ■ Alta integrazione ■ Linea di prodotto completa

### Massime prestazioni e flessibilità



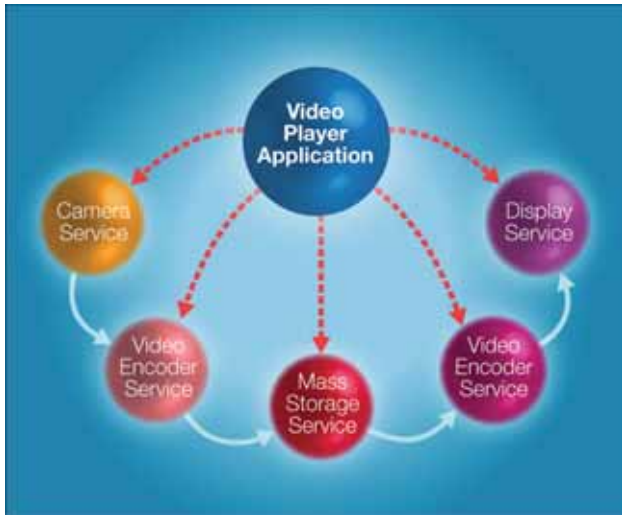
Tensione nominale	Numero di canali	Corrente di uscita		
		Classe 0.5 A	Classe 1.0 A	Classe 2.0 A
Bassa alimentazione 7V	1 ch	BD6210 HFP / F	BD6211 HFP / F	BD6212 HFP / FP
	2 ch	BD6215 FV / FP	BD6216 FP / FM	BD6217 FM
Media alimentazione 18V	1 ch	BD6220 HFP / F	BD6221 HFP / F	BD6222 FM
	2 ch	BD6225 FV / FP	BD6226 FP / FM	BD6227 FV / FP
Alta alimentazione 36V	1 ch	BD6230 HFP / F	BD6231 HFP / F	BD6232 FM
	2 ch	BD6235 FV / FP	BD6236 FP / FM	BD6237 FM

### Caratteristiche:

- Sistema di conversione PWM controllato in tensione
- Stadio di uscita a MOSFET complementari
- Protezione interna contro lo shoot-through in corrente
- Controllo PWM diretto in ingresso
- Basso consumo
- Circuiti di protezione migliorativi
- Circuito power saving
- Protezione ESD fino a 4kV
- Controllo indipendente per ogni singolo canale
- Linea di prodotto completa



readerservice.it n.24417



**Fig. 5 - Modello della piattaforma: RefinedPlatform 2**

più complesse è necessario procedere a una migliore definizione di SwitchProc. Le prime migliorie apportate alla piattaforma vengono illustrate in figura 4: qui CentralProc viene considerata un'unità di elaborazione hardware e sono state create due memorie condivise.

Il toolset ESL può impostare differenti tipologie per un nodo di comunicazione: un bus, una rete di instradamento, un collegamento punto-punto oppure un modello esterno definito dall'utente. Se la tipologia di un nodo di comunicazione è un bus, l'interfaccia è allocata al nodo. È possibile far ricorso a una modalità di simulazione specifica relativa a un bus con caratteristiche di concorrenza e non viene generata alcuna interfaccia. Il risultato che si ottiene non è così accurato come quello che prevede un bus con un'interfaccia, ma assicura una velocità di simulazione più spinta. CentralBus, che collega SharedMem e CentralProc, è un bus allocato con interfacce. Il buffer condiviso, la tabella degli indirizzi e la tabella della VLAN sono mappate nelle memorie per analizzare l'influenza dell'ampiezza del bus. SimplestPlatform è stata perfezionata in un altro modo per creare RefinedPlatform2 rappresentata in figura 5.

In questa piattaforma vi sono tre memorie, identificate come InputMem, OutputMem e TableMem.

## Modellazione architetturale

I modelli architetturali vengono creati mediante la mappatura del modello funzionale nel modello della piattaforma. Il tool di mappatura analizza le possibili soluzioni per ciascuna comunicazione funzionale in maniera automatica.

L'utente sceglie le modalità della loro mappatura sui nodi di comunicazione sulla piattaforma e definisce i loro corrispondenti attributi.

### SMMMap\_1Mem

Il primo modello architetturale è il risultato della mappatura del modello della memoria condivisa in RefinedPlatform1. L'obiettivo è individuare la corretta ampiezza del bus e i requisiti di memoria del sistema. In questo modello architetturale, l'accesso alle tabelle degli indirizzi (AddrT) e della VLAN (VLANT) e al frame buffer (SharedBuffer) viene eseguito attraverso il singolo bus condiviso (CentralBus). Esiste solo una memoria condivisa (SharedMem), ragion per cui il traffico su CentralBus dovrebbe essere abbastanza intenso.

### SMMMap\_2Mem

Per la configurazione della memoria condivisa può essere generata un'altra memoria al fine di immagazzinare i dati in ingresso (entry) delle tabelle degli indirizzi e della VLAN. In questo modello architetturale AddrT e VLANT sono mappati in TableMem invece che in SharedMem.

### OQMap\_2Mem

Il modello architetturale è basato sullo schema di pianificazione Output Queue configurato nel modello funzionale e nel modello della piattaforma RefinedPlatform2. L'utilizzo dello schema Output-Queue prevede che i frame vengano istradati immediatamente al corrispondente OutputBuffer.

### OQMap\_3Mem

Se si analizza la precedente architettura si può notare che il traffico su InputBus è ancora intenso in quanto InputProcessing e Routing devono utilizzarlo per effettuare tutti gli accessi alla memoria, compresi quelli relativi a AddrT, VLANT e Input Buffer. Un'opzione possibile sarebbe quella di rimuovere AddrT e VLANT da InputMem e impostare un'altra memoria indipendente per memorizzare i dati in ingresso delle tabelle degli indirizzi e della VLAN in modo del tutto simile a quanto fatto con SMMMap\_2Mem. In questo caso l'accesso alle tabelle degli indirizzi e delle VLAN avviene attraverso TableBus e non tramite InputBus.

## Altri tipi di architetture

Utilizzando la stessa metodologia, i medesimi modelli della funzione possono essere mappati nel modello della piattaforma RefinedPlatform2. Un'altra possibilità è quella di mappare AddrT e VLANT in InputMem o in TableMem. InputBuffer è mappato in InputMem e Output Buffer in OutputMem. Alla fine è stato possibile ottenere sei soluzioni architetturali: IQMap\_2Mem, IQMap\_3Mem, VOQMap\_2Mem, VOQMap\_3Mem, PMVOQMap\_2Mem e PMVOQMap\_3Mem. Questi modelli architetturali sono stati verificati utilizzando il medesimo testbench come nel caso del modello funzionale. I dati definitivi relativi alle prestazioni sono ricavati da questi modelli architetturali al fine di procedere alla soluzione ottimale.

**TRIAS Mikroelektronik**  
www.trias-micro.com