LOGICHE PROGRAMMABILI: PREFAZIONE

1. INTRODUZIONE

Diverse aree dell'industria elettronica stanno ponendo sempre più interesse alle logiche programmabili. Le logiche programmabili ad alta densità offrono un numero di gate di logica configurabile da utente sempre crescente e stanno rapidamente divenendo uno dei prodotti più usati dai progettisti.

Un progetto può essere realizzato e verificato in alcuni giorni, mentre seguendo lo stesso processo sono richieste diverse settimane adottando i gate array. Inoltre non esistono i costi di mascheratura (definiti in gergo NRE acronimo di non recurring engeneering cost) e i tempi di attesa per i prototipi.

A causa del fatto che i dispositivi sono configurati via software attraverso una programmazione quasi istantanea, le modifiche sono meno rischiose e possono essere effettuate in qualsiasi momento, invece nei gate array occorre attendere alcune settimane per la rimascheratura creando ritardi sui tempi schedulati. Questo influisce di nuovo sui costi del progetto e su quelli di produzione.

Uno studio compiuto negli Stati Uniti da una società di consulenza si conclude con l'affermazione che sei mesi di ritardo nell'entrata del prodotto sul mercato possono costare un terzo del potenziale profitto del prodotto nel suo tempo di vita.

Mentre l'uso delle logiche programmabili sta rapidamente incrementandosi, c'è ancora molta confusione intorno alla terminologia e ironicamente le compagnie fornitrici del silicio e i relativi responsabili di marketing contribuiscono a questa confusione.

Il nodo della questione sta nella differenza tra una PLD e un FPGA.

FIGURA 1

2. EVOLUZIONE STORICA

Le logiche programmabili sono nate nel 1984 e offrivano la possibilità di manipolare 1000 gate di logica programmabile nelle varie tecnologie allora presenti. Questi dispositivi si indirizzavano ai progettisti che ambivano ad integrazioni logiche e volevano evitare i rischi connessi alle rigidità dei gate array ove potevano farne a meno, inoltre il tempo di ingresso nel mercato, dalla fase di stesura delle specifiche, era divenuto un requisito critico per la competitività sul mercato stesso.

Sfruttando la riduzione del ciclo progettuale che ne scaturiva con questo metodo di lavoro, i vari progettisti inoltre, riuscivano ad introdurre nuovi prodotti sempre più rapidamente.

In questi primi anni il termine usato per queste logiche era high density PLD un'estensione logica delle logiche programmabili allora presenti (PAL e GAL). Tutti questi nuovi dispositivi erano usati in modo simile, cioè integrazione di logica sparsa (glue logic) precedentemente implementata usando circuiti integrati TTL e sostituzione di gate array di bassa densità

Con il passare del tempo e l'introduzione di nuove famiglie di componenti, nel tentativo di creare una distinzione tra differenti architetture di PLD alcune compagnie introdussero il termine field programmable gate array (FPGA) come una

sottocategoria delle logiche programmabili ad alta densità, nome desunto da qualche similarità nell'architettura ai gate array, altre come una categoria a parte.

Una simile e fuorviante strategia di marketing ha indotto i progettisti a pensare che FPGA significhi elemento riprogrammabile ad alta densità, mentre non-FPGA significhi elemento riprogrammabile a bassa densità Come risultato immediato di questa nuova caratterizzazione, e probabilmente ovvio, quelle che venivano definite high density divennero simple PLD, con il beneficio di essere le uniche a non essere orfane di una collocazione classificativa.

Deve essere chiaro che densità e velocità di funzionamento sono terminologie indipendenti dal mercato, ciò che conta nella distinzione è l'architettura. Nessuno confonderebbe una DRAM con una SRAM in quanto caratterizzate da un'architettura differente pur realizzando la stessa funzione. Analogamente nel mondo dei microprocessori le classificazioni avvengono in base al set di istruzioni e non alle sigle RISC o CISC che sono microprocessori che realizzano le stesse funzioni con architetture sostanzialmente diverse.

FIGURA 2

3. ELEMENTI DISTINTIVI

Le logiche programmabili sono caratterizzate da due elementi architetturali: interconnesioni e elementi logici.

Gli elementi logici noti come macrocells, logic cells, configurable logic blocks, sono le parti in cui viene realizzata la logica del progettista. Le strutture di interconnessione determinano come questi elementi logici sono connessi tra di loro all'interno del dispositivo.

Questi due elementi architetturali definiscono le prestazioni e la capacità delle logiche programmabili.

4. CLASSIFICAZIONE DELLE FPGA E PLD

In seguito alla crescente affermazione delle logiche programmabili del tipo FPGA, le compagnie produttrici usavano un diverso assortimento di strutture logiche per implementare funzioni booleane e ram nonchè diverse strategie di piazzamento dei canali di interconnesione che avevano lo scopo di connettere queste strutture. Tipicamente una funzione booleana in cui veniva inserita la parte combinatoria del progetto veniva realizzata attraverso un generatore di funzioni di quattro o cinque ingressi. Questa, definita con il termine look-up table (LUT), pilotava un elemento sequenziale eventualmente bypassabile.

Nelle PLD la struttura di interconnessione realizzata era continua e non segmentata come negli FPGA, questo consentiva trasmissioni dei segnali veloci e predicibili. Gli elementi logici inoltre erano costituiti da flip-flop pilotati da somme di prodotti.

In generale la ripartizione tra risorse di logica combinatoria e logica sequenziale non era la stessa. Infatti il rapporto combinatorio-sequenziale era favorevole alle PLD mentre il rapporto sequenziale-combinatorio era favorevole agli FPGA.

5. L'ARCHITETTURA IMPATTA SULL'APPLICAZIONE

I progettisti si sono trovati spesso di fronte alla scelta del giusto componente da selezionare tra quelli disponibili per le proprie esigenze. Queste selezioni sono basate sul raggiungimento degli obiettivi progettuali quali velocità, capacità e prezzo. Ma questi ultimi fattori sono conseguenza dell'architettura.

In generale è scorretto pensare che per progetti ad alta densità occorra usare un FPGA e per progetti ad alta velocità ma di bassa densità una PLD. È la combinazione di elementi quali implementazione logica e sfruttamento delle risorse di interconnessione che definisce la scelta.

Storicamente gli FPGA erano caratterizzati da celle piccole, da basso fan-in, e un elevato numero di flip-flop rispetto alle PLD. Le interconnessioni che nascevano tra una cella e l'altra necessarie per unire i livelli di logica necessari alla realizzazione di una data funzione booleana degradavano rapidamente le prestazioni. Per cui venivano usate per applicazioni lente e pipelined.

Le PLD invece, a causa dell'elevato fan-out, delle buone risorse di logica combinatoria e delle interconnessioni continue garantivano alta velocità Per cui venivano usate per le logiche di controllo, funzioni di decodifica, macchine a stati.

Oggi è possibile con l'introduzione delle nuove famiglie di FPGA avere un miglioramento nella gestione delle risorse di interconnessione che comunque restano una variabile non nota a priori, e un maggiore apporto di elementi sequenziali nelle PLD. Ma la decisione ultima che porterà ad una scelta dipende sempre dal tipo di progetto che si realizza e a quale architettura questo si addice. E tra i criteri di scelta quello basato sull'esperienza personale sicuramente ha un peso non trascurabile.

C'è comunque da dire che a parità di gate, una PLD oggi risulta più veloce di un FPGA che dalla sua parte presenta un maggiore numero di elementi sequenziali.

6. L'ARCHITETTURA IMPATTA SUI TOOL DI SVILUPPO

Nelle logiche programmabili non solo l'architettura predispone il dispositivo al tipo di applicazione ma anche il tool di sviluppo. La struttura di interconnessione è probabilmente l'elemento principale che contribuisce a complicare il software del tool di sviluppo.

Le interconnessioni segmentate rendono il place and route del progetto più difficoltoso e comunque in generale si richiede un processo di fitting in due tempi. Il primo step è il piazzamento, nel quale il software decide dove piazzare le strutture logiche tra quelle disponibili. Il successivo step è il routing che valuta differenti percorsi tra i disponibili per connettere due elementi logici.

Aspetti come timing e congestioni devono essere tenuti in conto dal software di place and route. Per soddisfare le esigenze del progettista questi due step richiedono sofisticati algoritmi e significativi processi di calcolo e utilizzo di memoria. Il tempo di compilazione può anche dilatarsi da minuti o ore a giorni.

Quindi la segmentazione delle risorse di routing impatta sul tempo di compilazione che risulta essere impredicibile. Più gli elementi logici sono separati all'interno del dispositivo, più variabili ci sono che il software deve considerare per scegliere il miglior percorso. Inoltre la cosa più importante è che solo alla fine del place and route il progettista conosce le reali prestazioni del progetto.

Nel campo delle PLD i ritardi sono predicibili a priori e quindi è facile fornire una previsione sulle prestazioni. Inoltre il piazzamento, essendoci interconnessioni continue, determina il routing. Questo singolo step di fitting comporta un minore tempo di CPU e memoria richiesta.

7. CRONOLOGIA

Si riporta nel seguito la cronologia storica degli eventi più significativi che hanno segnato l'evoluzione delle logiche programmabili in questi ultimi anni legandola alla sorte delle tre compagnie che il mercato indica come le più importanti.
Questa carrellata parte dal momento in cui si affermano le logiche programmabili come le conosciamo oggi omettendo l'evoluzione di quelle che sono i capostipiti dei dispositivi programmabili quali PAL e PLA che trovano radici nei primi anni 60.
Con questo non si vuole trascurare, anche per brevità di esposizione, le altre compagnie presenti sul mercato quali Actel, Vantis (AMD), Lucent (AT&T), Cypress, Atmel, Motorola, Philips, GateField, QuickLogic, le quali chi più chi meno giocano un ruolo importante nelle vendite e nella ricerca di nuove tecnologie e architetture.

1983: Viene fondata la Lattice con sede ad Hillsboro (Oregon - USA).

Viene fondata l'Altera con sede a San Jose (California - USA).

1984: Viene fondata la Xilinx con sede a San Jose (California - USA).

Altera introduce la prima PLD al mondo la EP300 e l'A-Plus che è il primo software per sistemi di sviluppo basati su PC.

1985: Lattice introduce la tecnologia EECMOS che useràper le sue PLD e inventa l'architettura GAL.

Xilinx introduce la famiglia di FPGA di prima generazione la XC2000 e il primo sistema di sviluppo che realizza il place and route: lo XACT.

1987: Lattice introduce la prima PLD CMOS con frequenze di funzionamento di sistema di 60 Mhz.

Xilinx introduce la famiglia di FPGA di seconda generazione la XC3000.

1988: Altera introduce la famiglia di PLD MAX5000.

La Xilinx crea nel suo catalogo una pin-compatibility all'interno della sua famiglia per rendere agevoli gli upgrade dei progetti.

1989: Viene introdotto l'AHDL (Altera High Description Language) da parte di Altera.

1990: Altera rende disponibile il primo sistema di sviluppo basato su Workstation.

1991: Altera introduce l'architettura PLD MAX7000 e il sistema di sviluppo MAX+PLUS II.

Xilinx introduce la famiglia di FPGA di terza generazione la XC4000 ed è la prima a rendere disponibile la RAM in FPGA.

1992: Lattice presenta le prime PLD riprogrammabili in system: le ispLSI1000.

Altera introduce l'architettura FPGA FLEX8000.

La Xilinx evolve la sua famiglia di XC3000 creando la performante XC3100 destinata alle applicazioni in FPGA high-speed.

1993: Altera introduce la prima PLD a 3.3 volt ed evolve la famiglia MAX7000 in MAX7000E.

Le PLD CMOS Lattice sono giunte a 5 ns nei tempi di propagazione.

La Xilinx introduce la sua famiglia di PLD XC7300.

1994: Altera introduce l'architettura PLD MAX9000.

Lattice presenta il primo kit (software più cavo di download) basato su PC per la programmazione in system. Inoltre introduce le famiglie ispLSI 2000 e ispLSI 3000 quest'ultima dotata di Boudary Scan IEEE1149.1.

Xilinx rompe la barriera dei 25000 gate introducendo il componente XC4025.

1995: Altera presenta la sua prima PLD riprogrammabile in system la MAX7000S e la nuova architettura di FPGA la FLEX10K che ha blocchi di RAM embedded.

La Xilinx introduce la famiglia di FPGA XC5200.

Entrambi forniscono la possibilitàdi utilizzare Core di terze parti.

1996: Altera fa evolvere la FLEX10K in FLEX10KA.

Lattice con la sua famiglia 6000 è la prima a creare PLD con memoria embedded.

La Xilinx introduce la sua prima PLD riprogrammabile in system la XC9500, fa evolvere la famiglia XC4000 in EX e XL e introduce il suo nuovo sistema di sviluppo l'M1.

1997: Altera introduce la famiglia FLEX6000 e la MAX7000A.

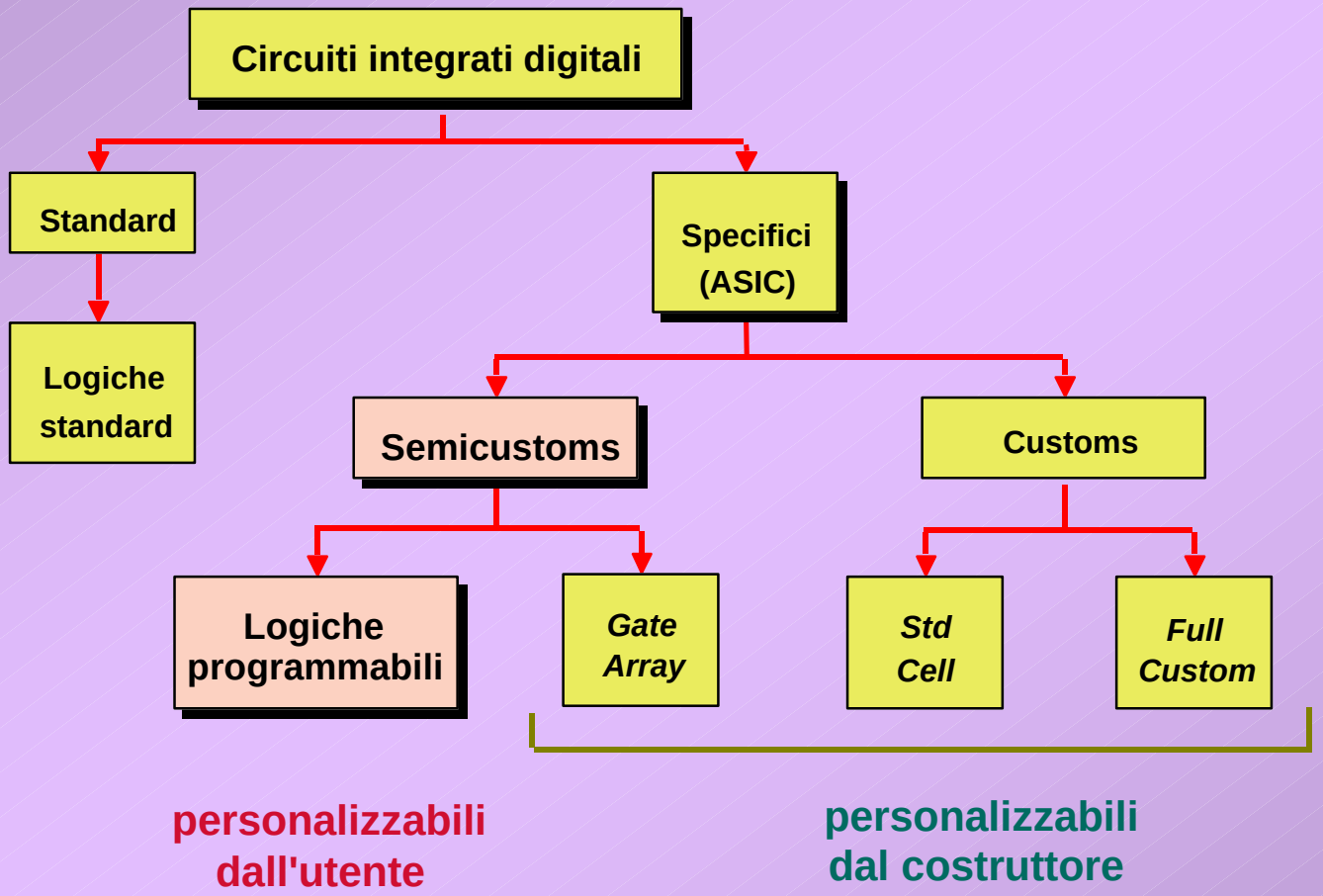
Lattice presenta i generic crosspoint digitali riprogrammabili in system.

Xilinx introduce la famiglia Spartan.

1998: Altera propone l'evoluzione della FLEX10K in FLEX10KE e presenta la nuova famiglia APEX 20K e il nuovo software Quartus.

Lattice presenta le nuove famiglie 5000 e 8000.

Xilinx presenta la VIRTEX e introduce nell'analisi statica dei ritardi il tempo minimo garantito di percorrenza di un segnale in una interconnessione.


Circuiti integrati digitali
Standard
Logiche standard
Specifici (ASIC)
Semicustoms
Customs
Logiche programmabili
Gate Array
Std Cell
Full Custom
personalizzabili dall'utente
personalizzabili dal costruttore

Albero distintivo