

CAPITOLO 2: PLD

1. ANALISI DEI REQUISITI

Le PLD che si sono inserite nel mercato con la tecnologia OTP sono state, con il passare del tempo, soppiantate da quelle riprogrammabili, sia per la naturale evoluzione tecnologica che per la comodità di utilizzo. Con l'avvento degli FPGA e il loro crescente sviluppo, al fine di evitare di confinare l'applicazione di una PLD ad un numero ristretto di progetti, occorreva introdurre nuovi elementi per aumentarne la competitività

Alle richieste d'utente di I/O gestibili con tensioni miste (3.3 e 5 V), possibilità di avere nello stesso package diverse scelte di velocità e densità (footprint compatibility), possibili cambiamenti del progetto senza alterazione della disposizione dei pin (pin locking), sia gli FPGA che le PLD hanno prima o poi dato una risposta. Le PLD dovevano però colmare il gap che esisteva con gli FPGA da un punto di vista della riconfigurabilità sul circuito stampato. Infatti gli FPGA basati sulla tecnologia static ram hanno insitamente questa caratteristica. E' così nata per le PLD la riprogrammabilità in system. Ma, a differenza degli FPGA static ram che si programmano ad ogni accensione, queste PLD mantengono il contenuto della programmazione fino al momento in cui deliberatamente si decide di modificarlo.

2. ISP (IN SYSTEM PROGRAMMING)

La riprogrammabilità in system configura la logica mediante un download, cioè un trasferimento di dati da un terminale intelligente alla circuiteria inserita nella logica stessa. Questa circuiteria, elemento aggiunto rispetto alla classica architettura di PLD, si fa carico delle operazioni necessarie alla predisposizione degli elementi programmabili come la logica d'utente richiede.

Questo approccio consente di aumentare ulteriormente la velocità di prototipazione, permette specifici test sul circuito stampato in fase produttiva, consente di effettuare la riconfigurazione in campo una volta venduto il prodotto. Inoltre, non estraendo più la PLD dal circuito stampato, questa non subisce l'eventuale danneggiamento dei pin causato dall'estrazione dallo zoccolo che a questo punto è superfluo.

Nella programmazione in system il dispositivo usa una tensione coincidente con quella di alimentazione, ad esempio 3.3 o 5 V. Sul circuito stampato occorre prevedere 4 o 5 connessioni che collegano altrettanti pin del componente ad un sito che può essere una porta di I/O di un microprocessore o una spina per l'inserzione del cavo di download. Nei casi in cui il download mediante cavo non risulta essere conveniente è possibile fare riferimento all'ispREMOTE™. Questa metodologia mette a disposizione diverse alternative al cavo in cui si stabiliscono delle connessioni tra il terminale in cui risiede il software di programmazione e il componente da riprogrammare presente sul circuito stampato. E' possibile avere una connessione mediante raggi infrarossi (ispIR™), mediante radiofrequenze (ispRF™), mediante un modem (ispMODEM^{RF}), mediante microonde (ispμW™).

Le isp PLD, oltre questi non trascurabili vantaggi, aprono nuovi orizzonti il primo dei quali è la User Electronic Signature (UES). Insieme alla logica che viene programmata e a quella che serve per realizzarla, trova posto un registro di dimensione variabile a seconda della densità della logica. Questo registro è destinato ad ospitare un campo informativo di etichetta, un po' come accade nei

floppy disk, solo che a cambiare è il tipo di supporto. In esso si possono inserire dei caratteri, visibili all'interfaccia software preposta al download in modalità ASCII o binaria, e modificabili da utente in maniera interattiva o mediante un file da passare al sistema di sviluppo che sta creando il file di configurazione. In questo registro è possibile inserire brevi commenti come il nome del progetto, la data della modifica, la versione, avendo traccia in ogni momento di quale funzionalità è stata inserita nel componente. I benefici che si traggono in progettazione, collaudo, produzione, nonché upgrade in campo sul prodotto già venduto e funzionante non sono da sottovalutare.

Le potenzialità che offre il download su dispositivi isp sono ben maggiori della riprogrammabilità di un solo componente. Infatti è possibile connettere in cascata più logiche programmabili per formare quella che si chiama Daisy Chain. Si possono connettere, eventualmente utilizzando dei buffer, fino a 60 componenti riprogrammabili in system, gestiti tutti da un unico software e un unico cavo di download. Questi componenti non necessariamente devono essere uguali, anzi possono essere di densità diverse, di famiglie diverse, di tipologie diverse (ad esempio una catena di dispositivi Lattice può essere composta di ispGAL, ispPLD, ispGDS, acronimo quest'ultimo di Generic Digital Switch).

La programmazione di una Daisy Chain può avvenire in maniera sequenziale, in questo caso il tempo di programmazione è la somma dei tempi dei singoli componenti o contemporaneamente su tutti i dispositivi (Turbo Download o Concurrent Programming) in cui il tempo totale coincide con quello del componente che impiega più tempo ad essere programmato. Fattori come grossa densità o vecchia tecnologia influiscono sul tempo di programmazione del singolo componente accrescendolo (si va da pochi secondi a decine di secondi).

3. RIPROGRAMMABILITA' IN SYSTEM E BOUNDARY SCAN

Alcune logiche programmabili prevedono la possibilità di eseguire il Boundary Scan test IEEE 1149.1 noto anche come JTAG (acronimo di Joint Action Test Group). Per realizzare questo test occorre accedere attraverso un'interfaccia a quattro o cinque fili (TDI, TDO, TCK, TMS ed eventualmente TRST) denominata JTAG Test Access Port, al Test Access Port controller (TAP controller), e sollecitare opportunamente la corona di registri predisposta negli I/O. Il TAP controller può essere usato per compiere la riprogrammabilità in system.

Non tutte le logiche in system hanno il TAP controller e non tutte hanno i registri per il Boundary Scan. Ci sono logiche che hanno solo l'interfaccia isp, altre che hanno solo il TAP controller e non i registri per il test Boundary Scan, altre che hanno il TAP controller e i registri, altre che hanno sia l'isp che il TAP controller compatibile Boundary Scan.

Nei casi in cui si ha solo il TAP controller, è possibile emulare il funzionamento Boundary Scan con un opportuno download che differisce e anticipa quello in cui verrà caricata la logica di utente. In questo caso il progettista dovrà farsi carico di riservare un pin di I/O per l'emulazione.

Da quanto detto appare evidente che è possibile realizzare una catena di componenti Boundary Scan tra cui quelli da programmare on board mediante il TAP controller. E' possibile creare catene miste cioè due catene, una isp e l'altra composta da elementi Boundary Scan, entrambe gestite da un unico software.

La programmazione di una PLD realizzata con tecnologia EECmos avviene mediante l'interfaccia ispTM o mediante il TAP controller. La programmazione di una Flash PLD o EEPROM PLD avviene mediante il TAP controller. I cavi per il

download, in base al fornitore, prendono il nome di isp Download Cable, BitBlaster o ByteBlaster Cable, Xchecker Cable, JTAG Download Cable.

4. TIPI DI OPERAZIONI POSSIBILI

Utilizzando le logiche programmabili in system mediante il cavo di download e gli opportuni software di programmazione è possibile compiere una serie di operazioni che si affiancano alla programmazione vera e propria. E' possibile verificare il contenuto della PLD programmata confrontandolo con il file di programmazione, in questo caso viene compiuto l'upload dal componente. E' possibile compiere una semplice operazione di cancellazione, calcolare il checksum, leggere o scrivere la User Electronic Signature, leggere l'identificativo del componente (ID), scavalcarlo se inserito in una catena.

Esiste una sequenza di operazioni da eseguire da un punto di vista della tensione di alimentazione al fine di evitare il danneggiamento del componente, del cavo o della porta del computer che si sta usando. Occorre alimentare il circuito in cui risiede la logica e poi inserire nell'apposito sito il cavo di download connesso all'altra estremità all'interfaccia parallela di un computer già funzionante. Al termine delle operazioni prima va rimosso il cavo e poi disalimentato il circuito.

5. PROGRAMMAZIONE MEDIANTE INTERFACCIA ISP

Gli elementi base della programmazione ispTM sono il segnale di controllo (MODE), i dati seriali di ingresso (SDI), quelli di uscita (SDO), il clock dei dati (SCLK), il pin di abilitazione alla programmazione (ispEN). Quest'ultimo segnale determina se il dispositivo è in Edit Mode (cioè modalità di programmazione) o Normal Mode (cioè normale funzionamento). La programmazione si basa sull'evoluzione di una macchina a stati presente all'interno della PLD. Tre sono gli stati che consentono di controllare la programmazione: normalità, caricamento dei comandi, esecuzione dei comandi.

All'attivazione del pin ispEN, la PLD entra in Edit Mode, ponendo i suoi pin di I/O in uno stato di alta impedenza e inizializzando la macchina a stati in quello di idle. In questo stato, è possibile visualizzare sull'interfaccia che controlla il download l'identificativo della PLD. Pilotando opportunamente i pin di MODE e SDI si estrae, con l'ausilio del clock SCLK, il contenuto a otto bit del registro di sola lettura che contiene l'ID.

Sempre con l'uso congiunto di MODE e SDI è possibile passare nello stato di caricamento comandi in cui all'interno di un registro vengono memorizzati i comandi che verranno eseguiti nel terzo stato. Tipicamente le operazioni nella PLD avvengono facendo riferimento ad una matrice di elementi da programmare o verificare. Un registro indirizzerà la riga (address shift register), un altro (data shift register) fornirà colonna per colonna, il valore che deve assumere la cella EECmos nella successiva fase di esecuzione.

6. PROGRAMMAZIONE MEDIANTE TEST ACCESS PORT (TAP)

Anche il TAP controller basa il suo principio di funzionamento su una macchina a stati, in questo caso sedici. I segnali interessati sono di nome diverso ma funzionalità analoga al caso precedente (TDI, TDO, TMS, TCK, eventualmente TRST). Le operazioni sono compiute sulla base del caricamento e aggiornamento di due

registri denominati Data Register (DR) e Instruction Register (IR). La decodifica del contenuto del secondo registro determinerà il tipo di operazione da eseguire all'interno della logica, considerando o meno il contenuto del primo registro.

7. ANALISI DELLE PRINCIPALI ARCHITETTURE

Le PLD, riprogrammabili o meno in system, hanno subito diversificazioni nelle architetture ispirate soprattutto da leggi di competitività sul mercato. Ciascun vendor ha presentato una tipologia a suo modo vincente creando un mercato in cui la logica del secondo fornitore era considerata perdente. Riportare tutte le architetture che si sono susseguite può risultare noioso e non rientra negli obiettivi di una analisi critica. Quindi, nel seguito, verranno descritte solo le principali architetture delle logiche riprogrammabili in system.

Una PLD è composta da una serie di elementi: celle di logica, interconnessioni, porte di I/O, circuiteria per la programmazione. Il modo con cui vengono realizzate le interconnessioni e la composizione delle celle di logica forniscono gli elementi necessari alla differenziazione. Esistono PLD con canali di interconnessione o con interconnessioni globali; celle di logica con uno, quattro o otto elementi sequenziali.

La struttura di interconnessione (globale o canalizzata) su cui si basa una PLD consente una accurata predicibilità delle prestazioni ancor prima di realizzare la fase di fitting. Questa predicibilità non è ottenibile in un FPGA.

8. ARCHITETTURA XC9500 (XILINX)

L'architettura XC9500 comprende una serie di function block in cui risiede la logica, connessi mediante una struttura di interconnessione globale denominata FastCONNECT Switch Matrix. I segnali entranti nella PLD confluiscono nella matrice prima di raggiungere i function block, quelli in uscita dai function block possono andare sia direttamente agli I/O sia alla matrice ove possono essere instradati verso altri function block. Il function block è costituito da un programmable AND-array, un product term allocator, 18 macrocelle.

Il programmable AND-array consente la creazione di 90 operazioni di and logico, comunemente denominate product term, costruite su una base di 36 segnali provenienti dalla switch matrix, considerati con il loro valore o il loro complementato.

I product term vengono smistati verso le macrocelle a gruppi di cinque dal product term allocator. Questi può anche riassegnare altri product term del function block alla macrocella in esame sfruttando una or di espansione localizzata al suo interno. Si raggiungono in questo modo 15 product term per macrocella introducendo un piccolo ritardo incrementale che viene definito t_{PTA} . E' possibile, con questa filosofia, rendere disponibili tutti i 90 product term ad una unica macrocella introducendo un ritardo di $8 t_{PTA}$. In quest'ultimo caso, in base al tipo di progetto e alla capacità del sistema di sviluppo, vengono utilizzate delle or di product term intermedie per non sacrificare completamente le restanti 17 macrocelle. In caso contrario queste verranno messe automaticamente in modalità powerdown al fine di ridurre la potenza assorbita.

Le macrocelle sono composte da una or che raccoglie i cinque segnali provenienti dal product term allocator ed eventualmente un sesto per l'espansione; una xor hardware, eventualmente non utilizzata, che opera tra un product term e la or stessa, oppure inverte semplicemente il valore di quest'ultima; un flip flop (configurabile di tipo D o Toggle) che a sua volta può essere scavalcato e in

quest'ultimo caso non più riutilizzato. I flip flop hanno sia il segnale di reset che di set asincrono.

L'intera PLD è poi supportata da una linea di reset o set globale e tre linee precostituite di global clock.

I blocchi di I/O non hanno elementi sequenziali. Nel caso in cui non sono usati possono essere programmati come pin di massa aggiunti per garantire maggiore immunità ai disturbi presenti sul circuito stampato. In linea con questo è prevista la possibilità di pilotare lo slew rate, pin per pin, rallentando la rapidità delle transizioni dei segnali ove l'applicazione in quel punto non richieda particolari velocità

Appartengono alla famiglia XC9500 componenti con un numero di macrocelle, e quindi possibili registri, che va da 36 a 288 per frequenze di funzionamento che vanno dai 100 ai 125 MHz.

9. ARCHITETTURA ISPLSI 1000,2000,3000,5000,6000 (LATTICE)

Anche in questa categoria di componenti l'interconnessione è di tipo globale e la matrice in questo caso si chiama Global Routing Pool (GRP). Questa fornisce le necessarie interconnessioni tra gli I/O e la logica di utente, in questo caso denominata Generic Logic Block (GLB), e quelle tra i vari GLB. Il GLB può essere visto come un AND-array, un Product Term Sharing Array (PTSA) e un output logic macrocell (OLMC). L'AND-array produce 20 product term (40 nella famiglia 3000, 160 nella 5000), a partire da un certo numero di segnali provenienti dal GRP (considerati con il proprio valore o il complementato) e da pin dedicati al solo ingresso. Nella famiglia 5000 questi segnali che alimentano l'AND-array sono 68, nella famiglia 3000 sono 24, nelle restanti famiglie 1000 e 2000 sono 16 più due provenienti direttamente dai pin d'ingresso. Dopo aver compiuto una or logica sui product term, il Product Term Sharing Array li redistribuisce per formare, in un unico livello di logica e senza espansioni, somme di prodotti da un minimo di due ad un massimo di 35. Le somme di prodotti nella macrocella possono alimentare una xor hardware e possono raggiungere, scavalcandola o meno, 4 flip flop (8 nella famiglia 3000, 32 nella famiglia 5000). Questi, a loro volta, possono essere scavalcati se la logica è di tipo combinatorio. Tranne nel caso della famiglia 5000, i flip flop non sono riutilizzabili ed ognuno all'interno di un GLB lavora con lo stesso segnale di clock.

Le uscite dei GLB possono essere riconvogliate nel GRP e contemporaneamente nell' Output Routing Pool (ORP), che risulta essere un insieme di risorse di routing preposto alle connessioni tra i pin di I/O e i GLB. Nelle famiglie 1000 e 3000 è presente un flip flop in ogni I/O, nella famiglia 2000 e 5000 no. In quest'ultima non è presente nemmeno l'ORP.

Appartengono a queste famiglie componenti che hanno da 32 a 672 registri (in questo caso 448 dedicati alle macrocelle 224 agli I/O), per frequenze di funzionamento dai 60 ai 200 MHz.

Una misura di densità in termini di macrocella è assimilabile a quella fatta per la famiglia XC9500 con le dovute attenzioni in quanto le macrocelle sono diverse. Una buona misura potrebbe essere il numero dei product term. In un caso (XC9500) in 36 macrocelle ci sono 180 product term, in un altro (ispLSI2000) in 32 macrocelle ci sono 160 product term. Il numero di product term per flip flop è lo stesso, cioè 5. Allora assimilando la quarta parte di un GLB alla macrocella, con un apporto medio di 5 product term, si ottiene una misura confrontabile in termini di densità. Quando invece si confronta il numero di product term usabili senza espansione si crea una

differenza di 20 a 5, allora in questo caso una macrocella della famiglia ispLSI2000 sembra composta di 20 product term.

A questo gruppo di componenti appartengono le famiglie 1000, 1000E, 2000, 2000E, 2000V, 2000VE, 3000, 5000V. Le sigle finali forniscono una differenziazione in termini di tensione di alimentazione (V che sta ad indicare 3.3 V) o accresciute risorse di routing (E), non di architettura.

Una struttura di interconnessione globale la si ritrova nella famiglia ibrida 6000 in cui è stata creata una PLD con tre moduli: uno di memoria, uno di registri o contatori precostituiti e configurabili, uno di logica programmabile che ricalca l'architettura della famiglia 3000. I moduli di memoria, in base al tipo di componente che si sceglie tra i tre possibili, possono essere una FIFO (ispLSI6192FF), una single port RAM (ispLSI6192SM), una dual port RAM (ispLSI6192DM), da 256x18 o 512x9 bit.

10. ARCHITETTURA AD INTERCONNESSIONE BIPLANARE

Lattice, nella famiglia 8000, realizza un sistema di interconnessione globale gerarchico. Crea una serie di blocchi all'interno dei quali prevede GLB e matrici ad interconnessione globale e connette questi blocchi con un unico piano di routing che chiama Global Routing Plane.

I blocchi denominati Big Fast Megablock contengono 6 GLB (in ciascuno dei quali si trovano 20 macrocelle) e il sistema di interconnessione che prende il nome di Big Fast Megablock Routing Pool. Le uscite dei GLB possono raggiungere sia il piano di routing globale che quello locale.

Le 20 macrocelle si ripartiscono 82 product term calcolati su una base di 44 ingressi e sono organizzate nella consueta struttura composta da un PTSA, un xor hardware, un registro. Il PTSA raccoglie i product term a gruppi di quattro e questa volta può creare funzioni senza ricorrere ad espansioni condividendone fino a 28.

Il numero dei registri totali è di 1152, di cui 840 inseriti nelle macrocelle. Se la logica di utente nella parte combinatoria è scomponibile, nel peggiore dei casi, in somme di quattro product term di 44 variabili, è possibile avere un sistema che evolve con 840 registri a 110 MHz.

11. ALTERA MIGRATION

Altera è partita da un'architettura ad interconnessione globale (famiglia Classic) per giungere, passando attraverso il Programmable Interconnect Array (famiglia MAX5000) e l'Enhanced Programmable Interconnect Array (famiglia MAX7000), alla struttura di routing canalizzata caratteristica della famiglia MAX9000 (FastTrack Interconnect).

La famiglia 7000 riprogrammabile in system (MAX7000S a 5V, MAX7000A a 3.3V), alloca intorno al Programmable Interconnect Array (PIA), architettura di interconnessione di tipo globale, dei logic array blocks (LAB) composti ciascuno da 16 macrocelle. Ogni LAB riceve dal PIA 36 segnali (con i loro complementati) e produce per ogni macrocella una or di 5 product term. Questo numero di product term che affluisce nella or può aumentare, se necessario, ricorrendo ad un'espansione gestita dal Product Term Select Matrix a scapito di un ritardo incrementale. Questa espansione può essere di due tipi: condivisa o parallela. Nella condivisa si sfruttano dei product term eventualmente non utilizzati rendendoli disponibili come ulteriori ingressi da affiancare a quelli provenienti dal PIA (uno per ogni macrocella fino ad un massimo di 16 all'interno del LAB). Nella parallela si

propagano le somme parziali da una macrocella a quella adiacente sempre all'interno del LAB. Anche in questo caso la macrocella è composta da una xor hardware e un flip flop non riutilizzabile. Nei casi in cui la logica di utente non richieda elevate frequenze di funzionamento è possibile disattivare, macrocella per macrocella, l'opzione di Turbo Bit, la quale rallenta il funzionamento della sezione in esame salvaguardando la potenza assorbita. Gli I/O di questa famiglia non hanno registri.

Appartengono a queste famiglie componenti che hanno da 32 a 1024 registri, per frequenze di funzionamento dai 60 ai 180 MHz. Se la logica di utente nella parte combinatoria è scomponibile, nel peggiore dei casi, in somme di cinque product term di 36 variabili, è possibile avere un sistema che evolve con 1024 registri a 125 MHz.

La famiglia 9000 è basata su di una canalizzazione della risorsa di routing denominata FastTrack Interconnect. Questa struttura è composta da una serie continua di linee orizzontali e verticali che attraversano l'intero dispositivo fornendo l'interconnettività necessaria ai vari LAB le cui macrocelle hanno la medesima struttura vista per la famiglia 7000S. La differenza si ha invece nell'insieme di segnali che affluisce all'AND-array che è composto da 33 (e non più 36) segnali provenienti dalle FastTrack Interconnect, 16 per le condivisioni dei product term, 16 per le retroazioni locali delle uscite delle macrocelle che appartengono al LAB. La completa predicibilità dei ritardi caratteristica di una struttura a routing globale viene in questo caso persa ottenendo una predicibilità basata su due livelli di ritardo. Uno riguarda le connessioni all'interno del LAB, l'altro quelle tra diversi LAB.

Il numero delle macrocelle per questa famiglia varia da 320 a 560 per frequenze di sistema dai 110 ai 180 MHz.

12. JAM

Allo scopo di uniformare il formato dei file per la programmazione delle PLD, Altera ha introdotto il linguaggio di programmazione e test denominato JAM™. Questo software si compone di una sezione (JAM Composer) che crea il file di programmazione e include in esso l'algoritmo di programmazione, e una sezione (JAM player) che mediante una catena JTAG la effettua. Il JAM player può risiedere sia nel computer che effettua il download, sia nel sistema operativo di un microprocessore presente insieme alla PLD sul circuito stampato.

Nei casi in cui si utilizzi il download, il JAM è in grado di programmare le PLD alla massima velocità tecnologicamente consentita. Infatti viene eseguita una prima scrittura di test in una cella per valutare, in base alla tolleranza di costruzione del silicio, che dimensione dell'impulso di programmazione applicare, e poi sulla base della misura fatta, viene compiuta la programmazione con impulsi di durata calibrata. In questo modo, evitando che la durata dell'impulso sia basata sul peggiore processo costruttivo ottenibile, si ottiene sempre un'operazione efficace.

Il formato del file JAM, proposto da Altera, vuole proporsi come alternativa al jedec file utilizzato dagli altri costruttori di PLD. Poiché ciascuno sostiene la validità del proprio standard sarà interessante vederne gli sviluppi.

13. ARCHITETTURA MACH (VANTIS)

La prima e la seconda generazione di PLD della Vantis, riprogrammabili in system attraverso l'interfaccia JTAG, sono le famiglie MACH 1 e MACH 2. Questi componenti basano la loro architettura su una serie di blocchi del tipo PAL interconnessi da una Switch Matrix. La funzione della Switch Matrix è quella di garantire una completa connessione tra gli ingressi e i segnali di feedback provenienti dai blocchi di logica che, a loro volta, sono costituiti da un product term array, un logic allocator, un gruppo di otto o sedici macrocelle e celle di I/O.

Il product term array genera la logica base gestendo un numero di ingressi dipendente dalla famiglia scelta. Il logic allocator distribuisce i product term alle macrocelle in base ai requisiti del progetto e a gruppi al massimo di quattro (product term cluster). Le macrocelle configurano i segnali e le celle di I/O producono i segnali destinati al mondo esterno. Nella famiglia MACH 2 esistono due tipi di macrocelle che possono generare sia uscite combinatorie che sequenziali: le Output macrocell e le Buried macrocell. Le Output macrocell inviano i loro segnali sia alle celle di I/O che alla Switch Matrix. Le Buried forniscono solo un collegamento interno verso la Switch Matrix. Oltre ad essere alimentate dal Logic Allocator, le Buried macrocell possono essere utilizzate per la sincronizzazione dei segnali provenienti dagli input. Questa ridondanza di macrocelle, consente il raddoppio della capacità della PLD, senza aumentarne il numero dei pin.

La famiglia MACH 4 ha un'architettura simile a quella vista per la famiglia MACH 2, con la presenza della Buried macrocell; presenta una capacità maggiore, l'inserimento nel Logic Allocator di un XOR hardware, un blocco di gestione dei clock in ogni blocco PAL.

La famiglia MACH 5 ha un'architettura di interconnessione dei blocchi PAL basata su due livelli. Quattro blocchi PAL, contenenti ciascuno 16 macrocelle, formano un gruppo interconnesso da una risorsa di routing locale chiamata Block Interconnect. La combinazione di questi blocchi e interconnessioni ad essi associate è chiamata Segmento. I Segmenti sono connessi attraverso il secondo livello gerarchico di routing chiamato Segment Interconnect. In questa famiglia non sono presenti le Buried macrocell, ma il Logic Allocator è in grado di assegnare fino a otto cluster di quattro product term l'uno a una macrocella.

Le famiglie MACH 1 e 2 hanno una densità che varia da 32 a 128 macrocelle, la MACH 4 da 32 a 256, la MACH 5 da 128 a 512.

14. ARCHITETTURA ULTRALOGIC E FLASH370I (CYPRESS)

La famiglia riprogrammabile in system UltraLogic 37000 è basata su un numero di blocchi logici (Logic Block) composti da un product term array, un product term allocator, 16 macrocelle, tutti interconnessi da una Programmable Interconnect Matrix (PIM). E' un'architettura che ricorda quella della famiglia MAX 7000S e XC9500 e consente di gestire da 32 a 512 macrocelle nelle tensioni 3.3 e 5 V.

La famiglia FLASH370i, che gestisce da 32 a 128 macrocelle, ha una architettura simile a quella della famiglia UltraLogic.

Queste due famiglie che, mediante il TAP Controller, effettuano la riprogrammazione, si presentano in due configurazioni base: I/O intensive e register intensive. La differenza tra queste risiede nella presenza o meno delle Buried macrocell nei Logic Block. Nella configurazione I/O intensive sono presenti, in entrambe le famiglie, 16 macrocelle e 16 celle di I/O; nella register intensive, nella

famiglia FLASH370i 16 macrocelle di cui 8 Buried e 8 celle di I/O; nella famiglia UltraLogic 16 macrocelle di cui 4 Buried e 12 celle di I/O.

15. ARCHITETTURA COOLRUNNER (PHILIPS)

L'architettura della famiglia CoolRunner™ si basa su un certo numero di Logic Block connessi da un array di interconnessione denominato zero-power (ZIA). Quest'array è ottenuto da un processo tecnologico denominato Fast Zero Power, che consente di ridurre il consumo di potenza statica a valori estremamente contenuti rispetto agli altri tipi di PLD (ad esempio 100 µA per 128 macrocelle).

Ogni Logic Block riceve 36 segnali dallo ZIA ed è composto da un PAL AND-array che gestisce 80 product term e un PLA AND-array che a sua volta ne gestisce 32. L'array PAL consiste di un AND-array programmabile con un OR-array fissato. L'array PLA consiste di un AND-array programmabile con un OR-array programmabile. Ciascuna delle 16 macrocelle che compongono il Logic Block è alimentata da una or che raccoglie al massimo 5 product term provenienti dal PAL AND-array e un product term addizionale proveniente dal PLA AND-array.

Questa famiglia gestisce, nelle due possibili tensioni di alimentazione (3.3. e 5 V), da 32 a 128 macrocelle.

FIGURA 1: Programmazione in system. Si monta il componente non programmato sulla piastra e poi si effettua la programmazione mediante download.

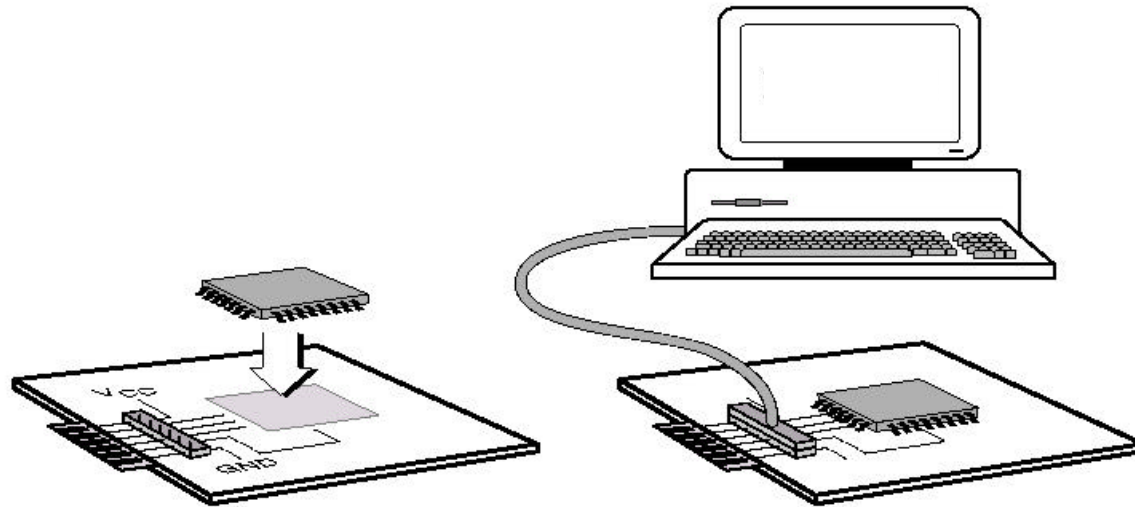


FIGURA 2: Daisy Chain composta da dispositivi riprogrammabili di famiglie diverse (Lattice Data Book)

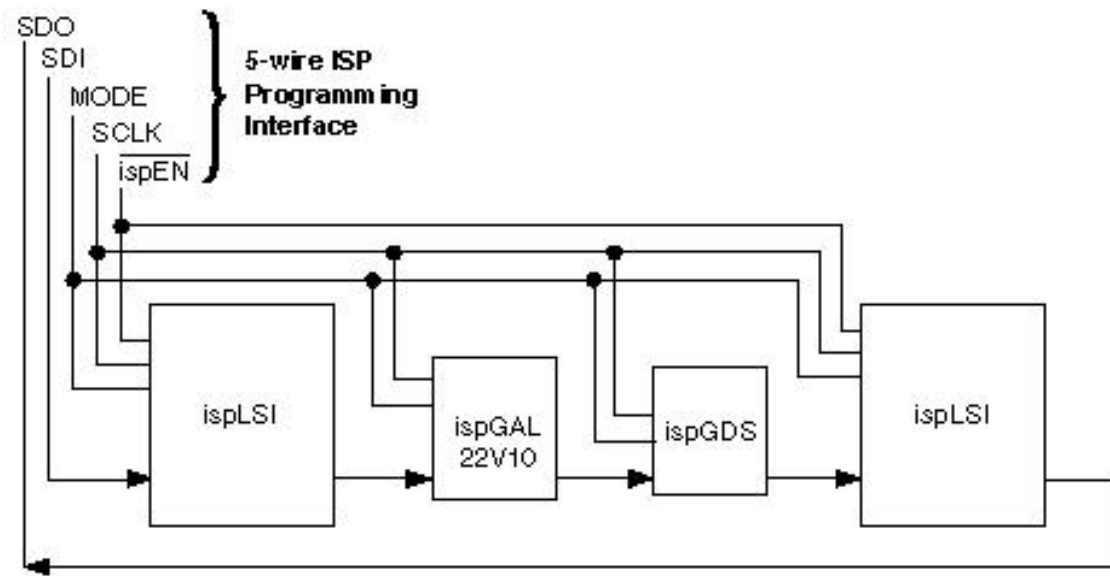


Figura 3: Macchina a stati che descrive la sequenza delle operazioni necessarie alla programmazione di una PLD EECmos (Lattice Data Book)

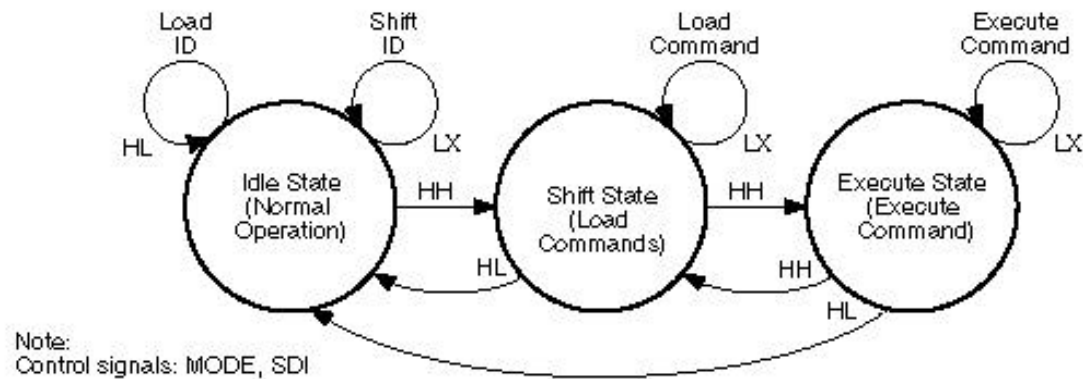


FIGURA 4: Evoluzione degli stati del TAP controller (Lattice Data Book)

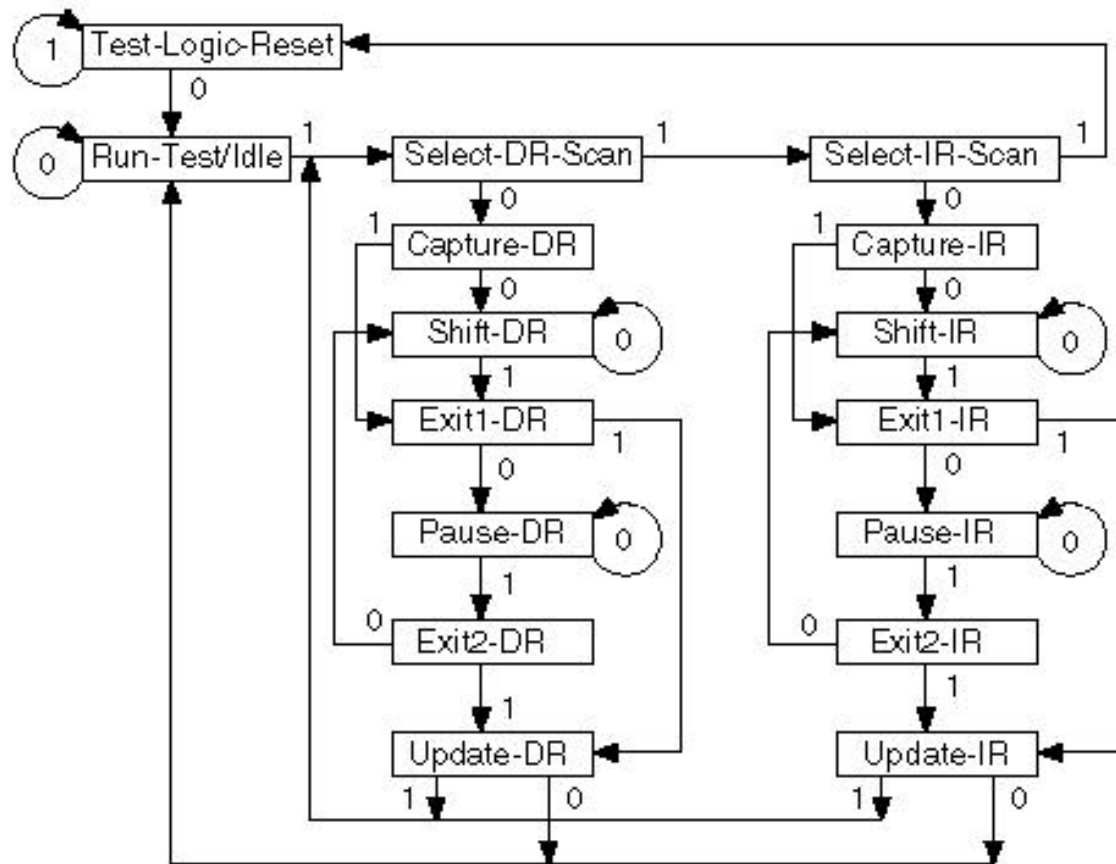


FIGURA 5: Architettura della famiglia MAX7000S. Si evidenziano i LAB in cui sono contenute le macrocelle e la struttura di routing formata dal Programmable Interconnect Array (Altera Data Book)

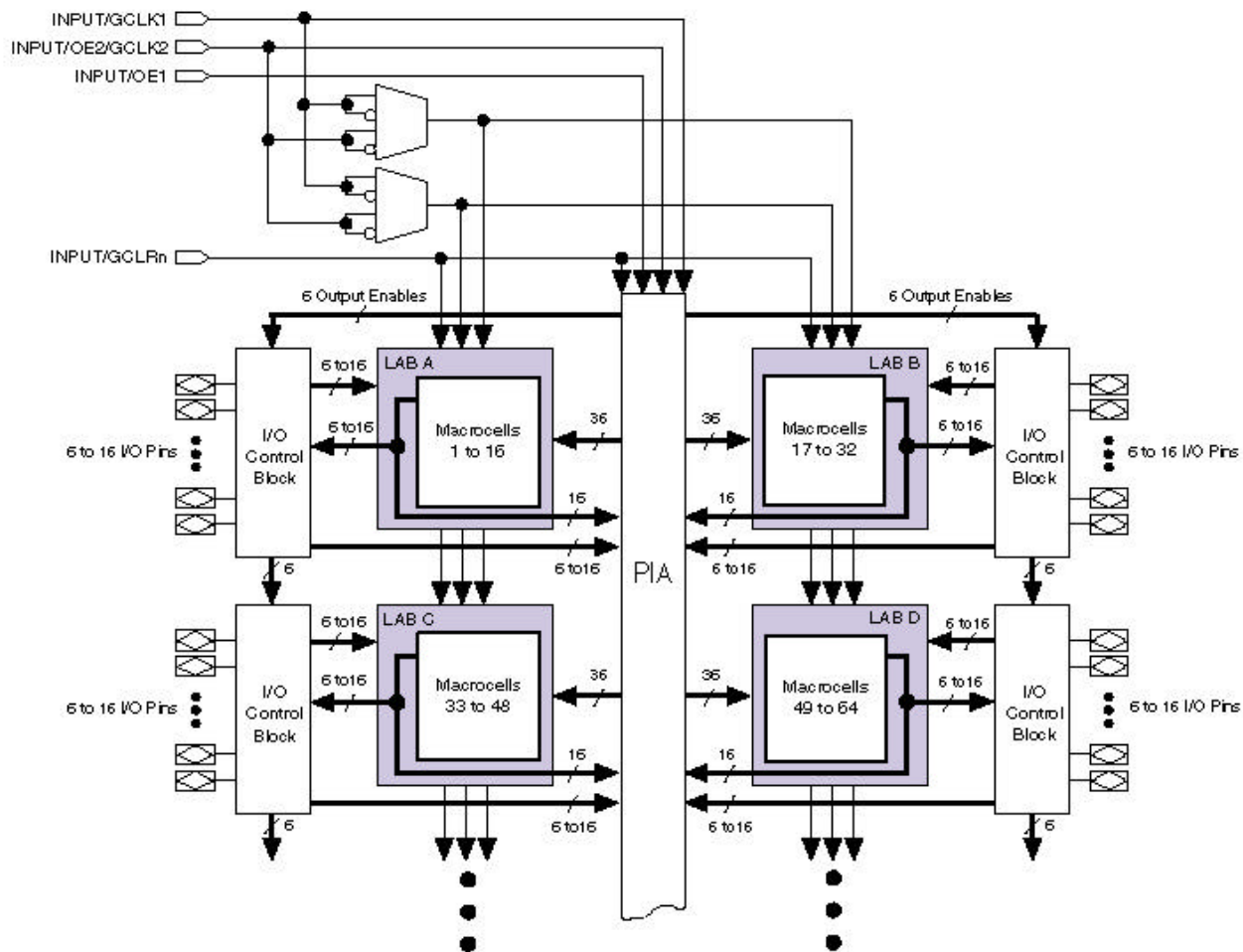


FIGURA 6: Elementi di un function block (Xilinx Data Book)

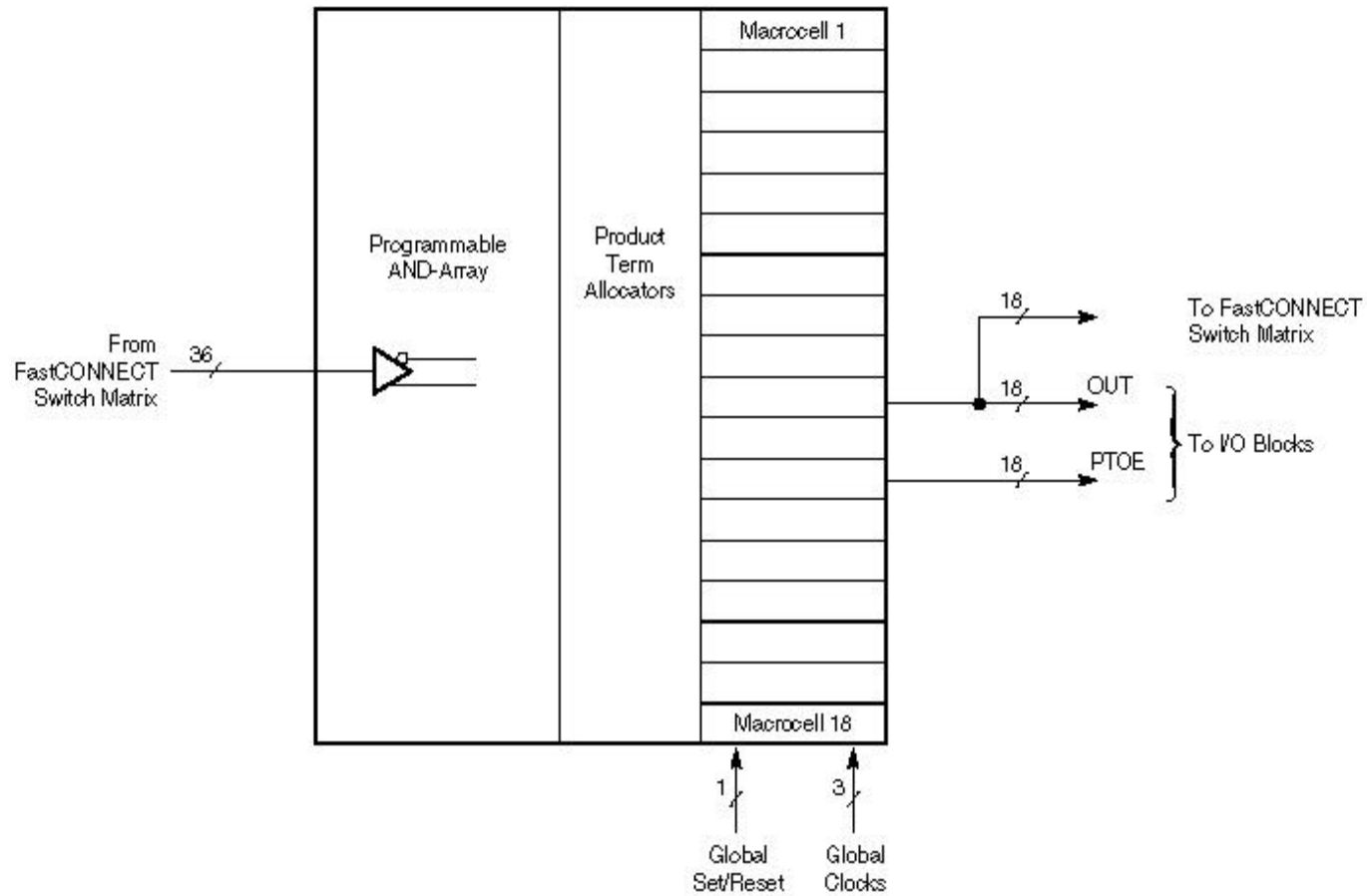


FIGURA 7: Macrocella della famiglia MAX7000S (Altera Data Book)

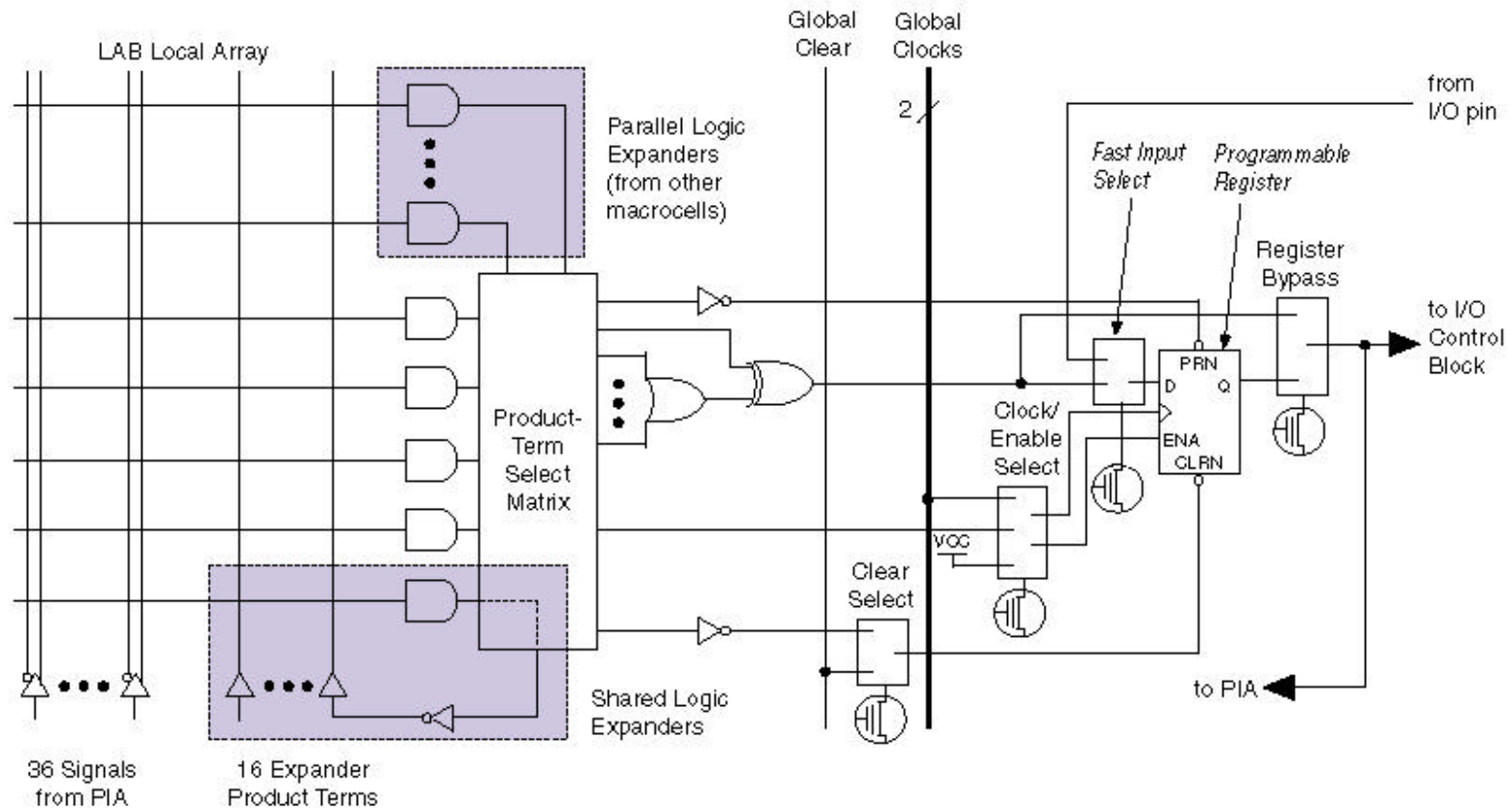


Figura 8: Architettura della famiglia ispLSI8000 con un piano di routing globale (GRP) e isole di routing globale locali (BRP) (Lattice Data Book)

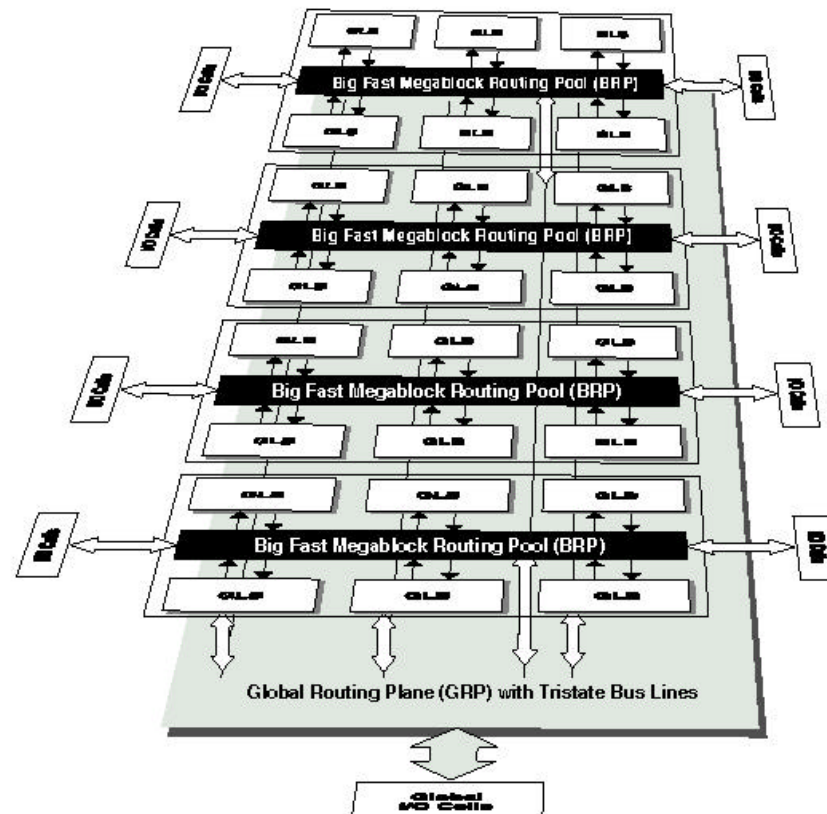


Figura 9: Architettura della famiglia MAX9000 basata sulle FastTrack Interconnect (Altera)

