Appendice D

Flusso di progettazione

D.1 Sommario

In questa appendice viene descritto tutto il processo che consente di passare da una descrizione matematica del sistema che si intende implementare su FPGA sino alla verifica del suo effettivo funzionamento passando per la verifica funzionale del VHDL.

D.2 Modello Matematico

La stesura di un modello matematico è di aiuto sia nella verifica della correttezza di una idea che nella comprensione stessa di determinati aspetti di una tematica che possono inizialmente rimanere poco chiari, soprattutto i numerosi esempi del Toolbox per le comunicazioni di Matlab oltre alle funzioni ad alto livello consentono un approccio semplice a tematiche quali modulazioni e demodulazioni, progettazione di filtri, codifiche e decodifiche. Sostanziale è anche l'utilità di Matlab laddove si desideri spostare tutta la complessità del progetto in migliaia di cifre contenute in una RAM, l'allestimento di script parametrici consente infatti di rispondere con rapidità a modifiche delle specifiche di progetto. Si può passare alla fase successiva del progetto solo dopo aver realizzato delle simulazioni su dei modelli matematici il più possibile vicini alla reale architettura che si intende implementare, in tal modo si possono ad esempio dimensionare opportunamente il numero di bit da assegnare a determinati segnali influenzando così la stesura del modello VHDL.

D.3 Modello VHDL

Anche nella realizzazione del modello VHDL occorre tener conto del passo successivo, l'implementazione, questo in quanto il VHDL è anche un linguaggio descrittivo pertanto alcuni modelli che potrebbero soddisfare

una verifica funzionale possono risultare non sintetizzabili, ciò si verifica ad esempio se all'interno di un processo si richiede che si operi su entrambe i fronti del clock. La verifica funzionale avviene applicando degli stimoli al modello e verificando le risposte o in termini di forme d'onda digitali oppure nel dominio del tempo o della frequenza previa creazione di un file e sua esportazione in Matlab dove si possono utilizzare delle varianti degli script che compongono il modello matematico. Quale software per la verifica e la stesura del modello VHDL si è utilizzato ActiveHDL, esso realizza anche un ambiente integrato verso diversi tipi di sintetizzatori VHDL e software per l'implementazione effettiva su FPGA consentendo di uno stesso progetto la verifica funzionale, RTL[1] e back-annotata.

D.4 Sintesi del progetto

La sintesi consente di passare dalla descrizione VHDL del progetto alla descrizione RTL ossia la sua espressione in termini di registri e logica combinatoria, in questa fase è importante l'utilizzo di un software quale Synplify che illustra graficamente la corrispondenza tra le due descrizioni ed effettua anche una verifica delle prestazioni in risposta a determinati vincoli sul timing di alcuni segnali di rilievo tra cui certamente il clock. La Xilinx distribuisce con i suoi prodotti anche XST un sintetizzatore meno grafico rispetto a Synplify ma con prestazioni abbastanza simili, esso ha il vantaggio di riconoscere automaticamente le stringhe di inizializzazione delle BlockRAM[2] consentendo pertanto un più veloce passaggio dal VHDL per la verifica al VHDL per la sintesi.

In questa fase oltre ai vincoli temporali al progetto è bene inserire anche i vincoli sui pin che si desiderano utilizzare, al fine di ottimizzare il timing ad esempio nel progetto ThinModulator son stati prescelti tutti pin che si trovano dal lato della FPGA dove sono dislocate le BlockRAM utilizzate, con questo semplice accorgimento si risparmia una inutile fatica al software d'implementazione, occorre però tener presente che non tutti i pin della FPGA sono disponibili sugli slot che sulla scheda DINI sono adiacenti ad essa.

D.5 Mappatura

Impostando in Synplify anche il tipo di FPGA sul quale si desidera realizzare il progetto è possibile ottenere la vista tecnologica che mostra come il progetto RTL e quindi il progetto VHDL viene effettivamente implementato nel dispositivo, in sostanza in questa fase vengono stabilite quali tipologie di risorse verranno utilizzate.

[1] Register Transfer Logic
[2] Appendice(C.3.4.2)

D.6 Piazzamento delle risorse

Una volta individuate la tipologia di risorse da utilizzare rimane da effettuare il piazzamento, ossia la selezione nominale delle risorse da utilizzare, questa fase è in genere abbastanza veloce ed il progettista la può influenzare mediante il FloorPlanner con il quale si specificano quali siti all'interno della FPGA debbono ospitare una determinata funzione, in sostanza si impongono dei vincoli fisici che possono essere memorizzati in un file UCF[3] o PCF[4]. Il Floorplanner può risultare utile ad esempio per impostare le BlockRAM da utilizzare per un dato segnale in modo da rendere il più breve possibile il percorso dalla sua uscita sino al registro ed al pin d'uscita.

D.7 Connessione delle risorse

Questa fase è sicuramente la più lunga e complessa, vengono selezionate le risorse di connessione[5] che consentono di soddisfare al meglio i vincoli temporali impostati nel file UCF garantendo comunque i vincoli fisici impostati nel file UCF o PCF. Anche per questa fase Xilinx mette a disposizione del progettista un Tool quale FPGA Editor che gli consente di imporre determinate connessioni o modificare le connessioni effettuate dal tool automatico.

D.8 Simulazione back-annotata

Dopo aver verificato mediante lo STA[6] che i vincoli sul timing sono soddisfatti si può generare il modello di simulazione del progetto ottenuto dopo il PAR[7], esso è costituito da due file uno con estensione .sdk e l'altro .vhd, quest'ultimo realizza l'architettura *structure* del modello VHDL, creando per essa un TestBench si debbono ottenere a parità di stimoli le stesse forme d'onda che si ottengono dal modello funzionale.

D.9 Programmazione della FPGA

Se anche la simulazione back-annotata ha avuto esito positivo si passa alla creazione del file .bit e successivamente del file .hex da inserire nella PROM che ad ogni riaccensione si occupa di riprogrammare le FPGA presenti sulla scheda DINI. E' necessario impostare per il file il formato HEX e non selezionare l'opzione swap bits. Il Data Stream deve contenere nell'ordine il

[3] User Constrain File
[4] Physical Constrain File
[5] Appendice(C.3.3)
[6] Static Timing Analyzer
[7] Place And Route

file fpga_f.bit per la gestione del bus PCI, il file .bit relativo al progetto che andrà ad occupare la FPGA A e 4 file blank_1.bit che riempiranno le altre FPGA. Salvando viene automaticamente generato il file .hex, per trasferirlo nella Prom occorre seguire la seguente procedura sul computer nel quale è stata installata la scheda DINI:

1. inserire il Jumper J2 sulla scheda e premere il tasto giallo.
2. riavviare il computer.
3. spostare il file .hex generato dal prom formatter nella directory `F:/SchedaDini/AETEST/aetest_nt` .
4. far partire il programma AETEST.EXE ed eseguire i diversi passi di programma:
   - selezionare Open Device.
   - selezionare 3) Flash menu.
   - selezionare 6) Download Hex File.
   - scrivere il nome del file .hex e premere invio.
   - premere due volte invio quindi uscire.
5. togliere il Jumper J2 e premere il tasto giallo.
6. controllare che si spengano i led rossi superiori.

D.10 Verifica sperimentale

Per la verifica sperimentale vengono utilizzati un *pattern-generator* che viene caricato con dei vettori di test da applicare alla FPGA in maniera ripetitiva o singola, lo script Matlab ***CreaSequenzaPatternGenerator.m*** *(Listato E.2.2)* consente di creare in maniera automatica il file di testo con il quale caricare il *pattern-generator*, in esso si può anche impostare se il clock di emissione dei dati debba esser prodotto autonomamente dal *pattern-generator* oppure sincronizzato ad un clock esterno come è avvenuto per le prove del modulatore. Le uscite del modulatore vengono invece inviate all'*analizzatore di stati logici*, si effettua una singola acquisizione di 64mila campioni in accordo ai 6mila applicati ed il file ottenuto viene esportato verso Matlab dove un'analisi dello spettro consente di confrontare il risultato teorico col risultato sperimentale.