

Sintesi di un blocco DFT

1. Introduzione

Questa esercitazione si colloca nell'ambito della realizzazione di una piattaforma di simulazione per sistemi embedded multiprocessore (MPSIM). Tale piattaforma ha come obiettivo quello di permettere l'investigazione di metodi per ottimizzare la performance, in velocità e dissipazione, di System-On-Chip (SoC). A questo scopo, è utile considerare la possibilità di affiancare, nella piattaforma, unità general-purpose come dei processori ed unità dedicate allo svolgimento di funzioni specifiche. Una di queste funzioni, molto utilizzata in ambiti audio/video, è ad esempio la Discrete Fourier Transform (DFT)/Fast Fourier Transform (FFT).

2. Scopo del Progetto

Obiettivo finale dello sviluppo sarà l'ottenimento di una netlist che descriva un circuito capace di implementare la funzionalità di DFT/FFT. Parallelamente, dovrà essere possibile simulare l'unità all'interno della piattaforma MPSIM per ottenere la validazione funzionale e prestazionale. Nel corso del progetto, sarà possibile familiarizzare con l'ambiente di sviluppo di Synopsys, utilizzandone vari tool, e sarà richiesta una breve analisi delle conseguenze che vari approcci di scrittura del codice e di metodologie di sintesi possano avere sulla netlist risultante.

3. Passi implementativi

Lo sviluppo del progetto potrà essere portato avanti in più fasi.

3.1 Prima fase

Obiettivo: scrittura di codice che implementi la DFT/FFT a livello behavioral.

Durante questa fase, sfruttando le metodologie di progetto e simulazione fornite da Synopsys CoCentric System Studio (CCSS), dovrà essere scritto del codice che implementi la DFT/FFT a livello behavioral. In particolare, non sarà richiesta una corrispondenza specifica tra la computazione eseguita e le unità di calcolo effettivamente istanziate per eseguirla, né una stima delle latenze di calcolo.

In questa fase, inoltre, si dovranno stendere le specifiche dell'interfaccia di comunicazione tra il modulo e l'esterno (segnali, temporizzazioni e spazi di scambio dati). L'interfaccia potrà essere minimale, basata su un handshake semplice ed un insieme di registri.

3.2 Seconda fase

Obiettivo: scrittura di codice che implementi la DFT/FFT a livello RTL.

Basandosi sul codice precedente, la descrizione del modulo dovrà essere affinata e portata al livello RTL. Questo comprenderà una prima analisi delle risorse di calcolo da impiegare e delle temporizzazioni del modulo.

3.3 Terza fase

Obiettivo: validazione del modulo.

In questo passaggio, il codice SystemC scritto ai punti precedenti dovrà essere testato all'interno di MPSIM, interfacciandolo con un device che permetta di connetterlo su bus AMBA AHB.

3.4 Quarta fase

Obiettivo: sintesi del modulo.

Durante questa fase, si procederà alla sintesi del blocco DFT/FFT. La sintesi procederà secondo più vie parallele, per confrontare i risultati ottenuti: sintesi diretta da SystemC behavioral; sintesi da SystemC RTL; conversione da SystemC ad HDL (VHDL o Verilog) e sintesi

da HDL. Nel corso di questa operazione, verranno confrontati diversi approcci alla scrittura del codice ed all'impostazione della fase di sintesi per studiare gli effetti sulla netlist in uscita. Si cercheranno di analizzare i possibili metodi per ottimizzare la netlist: riuso di unità funzionali e di memoria (risparmio di area) oppure loro duplicazione (minimizzazione delle latenze).

3.5 Quinta fase

Obiettivo: cosimulazione finale del modulo.

Il codice HDL ricavato nel passo precedente dovrà essere risimulato in ambiente SystemC, sfruttando la funzione "HDL wrapper" di CoCentric System Studio.

4. Documentazione

- www.systemc.org
 - <http://astronomy.swin.edu.au/~pbourke/analysis/dft/>
 - Manualistica Synopsys (non disponibile pubblicamente)
- VHDL:
- <http://www.cs.ucr.edu/content/esd/labs/tutorial/>
 - <http://www.gmvhdl.com/VHDL.html>