

FPGAではじめるVerilog-HDLの学習

CQ出版社刊行物を活用した 効率的なVerilog HDL言語の学習法

Embedded Technology 2006

In JAPAN

来栖川電工有限公司

中野R&Dカンパニ 井倉 将実

- ASIC開発現場ではVerilog-HDLが標準的
 - FPGA/CPLDによる機能をASICに落とすときにファブベンダが要求するハードウェア設計言語として一般的
 - C言語を拡張したような言語形態
- シミュレーション環境が他HDL言語に比べて優秀
 - もともとASIC開発支援用の検証ツールから派生
 - C言語ライクな検証環境の提供がされている
 - Verilog-HDL
 - PLI (Programmable Language Interface)による拡張
 - VHDLのFLIによる機能拡張よりも豊富である
 - 一般的にVHDLにくらべてシミュレーション速度が速い

- 他のHardware設計言語を知らない人が言うこと
 - xxxxよりも記述量が少なくて済む
 - ┆ “>”や“++”などをつかえば、確かに小さくなるが…
 - 記述量の少なさよりも、正しく論理合成させることが大切
 - ┆ 連結演算子などを多用することはいいが…
 - `&(signal_A[xxx:xxx]), ~|(signal_B[xxx:xxx])`等, このような記述をあとから読んで「???’なことになるように.
 - VHDLの” for i in 0 to xx loop ~ end loop”よりも便利には違いない記述法があるが, 可読性も十分に考慮すべき.
 - 細かなゲート設計が可能と「聞いている」
 - ┆ VHDLなどでも同様に可能
 - むしろ, 厳格にビット幅などを管理する必要のあるVHDLのほうが記述でミスを起こすことは少ない.

■ すでにVHDL言語を覚えている場合には・・・

■ 細かな言語仕様の違いで違和感を覚える

┆ ビット幅指定無しでも連結が可能

- VHDL : `Q[31 downto 0] <= A[15 downto 0] ;`
 - 上記の場合にはエラー

- Verilog-HDL : `Q[31:0] <= A[15:0] ;`

- 上記の場合には, 上位16ビットに”0”が入る(Unsigned型の時)

┆ 論理演算や条件判断の記述に悩む

- 論理演算

- VHDL : AND , OR , NOT , XOR

- Verilog-HDL : & , && , | , || , ~ , ^

- ビット演算子と論理演算が厳密に別れている

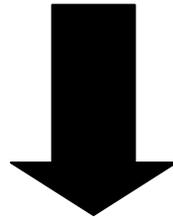
- 条件判断

- VHDL : = , /= , >= , <=

- Verilog-HDL : == , != , >= , <=

あいまいな記述でも通ってしまう
言語仕様のアバウトさが目に付く

なにをやっても回路図ができそう



どんな回路が出来上がるんだろう...