Timing Optimization

Professor Adnan Aziz

1. An n-bit adder has a 1-bit carry-in and two n-bit numbers as inputs and produces a 1-bit carry-out and an n-bit sum. When n is large the adder is typically constructed using several smaller sized adders.

   (a) Write a Verilog description of an 8-bit adder that used four 2-bit carry-lookahead adders in cascade. Refer to any textbook on computer arithmetic (such as [1]) to see how this is done.

   (b) Use `vl2mv` to translate your Verilog program to a hierarchical multi-level logic network in *blif-mv* format. Use the *write_blif* command in `VIS` to convert this to the *blif* format.

   (c) Use SIS to minimize the area of the logic of the 8-bit carry-lookahead adder as best you can. Recall that the circuit area is measured by the number of literals in factored form. Report the results, scripts, and CPU times used in your optimization.

   (d) Write a Verilog description of an 8-bit ripple-carry adder. Compare the area of the carry-lookahead adder and the ripple-carry adder after you try to optimize both. Attempt to reduce the delay of the ripple-carry adder by using the *speed_up* command in `SIS`. (Read the help page for this command.) Compare the circuits speeds of the sped-up ripple-carry adder and the original carry-lookahead adder (the *print_delay* command can be used for this). Be careful to ensure that both circuit implementations use the same delay model (suggested model is *unit_fanout*) and that all gates are 2-input NAND gates (the form used by the *speed_up* command; the circuit can be decomposed into 2-input NAND gates using the *tech_decomp* command).

   (e) Obtain an area-delay trade-off curve starting with the ripple-carry adder. (This can be done by setting appropriate delay constraints on the outputs of the adder using the *set_delay* command.)

80 marks

References

[1] N. H. E. Weste and K. Eshragian. *Principles of CMOS VLSI Design.* Addison-Wesley, 1993.