VLSI Systems Design (ESE 355)
Spring 2003
Home Work - 3
Due on March 31, 2003 (Monday)

Design management methodologies such as modular design, hierarchical design, and bit-sliced design come quite handy while designing VLSI systems. Of particular interest is the *bit-sliced design technique* where one basic cell can be designed and repeated many times to comprise a larger system. Structural regularity of connections among the cells is the characteristic of bit-sliced designs. For example, register cells can be designed such that n of which, when properly composed, form n-bit registers. A full adder cell can be repeated in both x and y directions to generate a multiplier. Bit-sliced designs often result in dense layouts, minimize the design time and show high performance provided each cell is designed carefully.

Pipelining is a design technique often used to improve the throughput rate of VLSI systems. Pipelined designs facilitate spatial overlapping of computations belonging to different input data sets, without much duplication of hardware. Whereas you will study pipelining techniques in detail the Computer Architecture course, this assignment aims to give you some introductory experience in designing simple, hazard-free structural pipelines. Pipelining, when combined with the other powerful VLSI design technique viz. bit-slicing, can be extremely effective in designing high-performance systems in a relatively short time.

Standard cell libraries are collections of predesigned layouts that facilitate reduction in design time, while not compromising performance too much. Standard cells thus let you concentrate more on solving the problem rather than worrying about layout entry. Standard cells also facilitate bottom-up construction of a hierarchical design. A collection of standard cells will be provided to you by the TA.

In this assignment you will *not* design any basic cells of your own. Instead, you will exclusively use the standard cell library for all your basic cell needs.

Bit-Sliced Absolute Value Network: It is desired to design a combinational logic network $AVN(n)$, to compute the absolute value of the difference between two n-bit positive binary numbers A and B. Design a AVN(1) slice such that n AVN(1) slices can be connected together, without needing any additional glue logic, to form AVN(n). *Show the logic gate diagram of AVN(1). Show how three AVN(1) cells can be composed to form AVN(3).*

Design, using Magic and IRSIM, a AVN(1) slice in 1.2 micron CMOS such that the worst case propagation delay is no more than 100 ns.

Subit the magic layout and IRSIM simulation results for various combinations of inputs.

Connect four AVN slices together to from AVN(4). *Submit magic layout and the IRSIM simulation results for AVN(4).*

Develop a pipelined design of AVN. Your pipeline should have four stages, ie. four AVN(1) cells. It can be viewed as modification of the AVN(4) where sufficient number of flip-flops are introduced in every path from input to output. In steady state the pipeline should accept a pair of 4-bit numbers in every clock cycle and produce one 4-bit result in every clock cyle. *Submit the magic layout and IRSIM simulation results. Report the expected maximum clock speed of your design, the expected latency and throughput rate.*

In all cases, your simulation waveforms must be annotated with comments so what you are trying to show is evident.