Department of Electrical and Computer Engineering
State University of New York at Stony brook

ESE 555 Advanced VLSI Systems Design (Fall 2009)

CAD Assignment 5: The Shifter

Assignment

Assignment To design a 16-bit shifter for your microprocessor.

Description

The shifter is an essential element for many microprocessor operations. It may be used to align or scale data, manipulate bits and bytes, or in an automatic or program-controlled shift-and-add multiply function. In the baseline machine, you are required to implement only a Logical Shift, which shifts data in a register (dest) as specified by a signed count operand (twos complement). A positive count specifies a left shift; a negative count specifies a right shift. You must support a shift amount contained in a register or the immediate field of the instruction. All bits shifted out of the destination register are lost. All destination bits not mapped from the original operand are filled with zeros.

Other common shift functions, which you may want to add (optional) to your processor are arithmetic shifts, arithmetic shift including the carry bit, byte swap, and rotate. In right arithmetic shifts, the high-order bits are filled with the original sign bit. To shift the contents of a register one bit per clock cycle, an ordinary shift register, which can be assembled from cascaded D-latches, could be used. However if there is a need to shift data by an arbitrary number of bit positions within one clock cycle, a dedicated, programmable shifter is required.
Two frequently used approaches are (i) Barrel Shifters, and (ii) Logarithmic Shifters. While the barrel shifter implements the whole shifter as an array of pass transistors or multiplexers, the logarithmic shifter uses a staged approach. In general, barrel shifters are appropriate for small shift width and logarithmic shifters are more suitable for shifters of large width. Both types of shifter can be implemented in either pass gates or gate-based multiplexors. Shifters should be disabled in low-power designs when they are not in use.

Barrel Shifter

A simple 4x4, transmission-gate barrel shifter is shown in the textbook. It consists of an array of transmission gates, with the number of rows equal to the bit width of the data words, and there is one column for each shift possibility. In this case, both are set equal to four. The input to the shifter is the value to be shifted, a literal <6:0> and the shift amount <3:0>. The control wires are routed diagonally through the array. A major advantage of this shifter is that a signal has to pass through at most one transmission gate, regardless of the size of the shifter. The shifter delay does, nevertheless, grow with the size of the shifter, as the capacitance at the buffer inputs rises with additional columns. The layout size of this shifter is dominated by the number and pitch of wires, rather than by transistor area. The size of the shifter grows linearly with the number of shift positions allowed. A decoder is required to select one of the shift lines.

Logarithmic Shifters

In logarithmic shifters, the total shift value is decomposed into shifts over powers of two. A shifter with a maximum shift width of M consists of a log2(M) stages, where the i-th stage either shifts over 2 i or passes the data unchanged. The textbook shows a multiplexor-based logarithmic shifter. One could also use tristate buffer multiplexors or logic gate multiplexors instead of transmission gates. A circuit like this with many transmission gates in series would benefit from having buffers inserted along the path. Minor changes will convert it to a logical shift, or a small amount of logic can replace the hard wiring to allow several kinds of shifts. Logarithmic shifters have intrinsic decode, as the shift bits are used directly to control the muxes or transmission gates. Small barrel shifters, however, may be more compact than logarithmic shifters. Logarithmic shifters are always better when the word size is large and there are many shift possibilities.

Procedure

° Schematic Design
You may implement any shifter, which achieves the required functionality-both left and right logical shifts from 0 to 15 bits. If you are designing the pass transistor-based shifter, keep in mind that you will also have to design the decoder, but it is sufficient in this CAD assignment to design just the shifter array itself. (The decoder can be done as part of the control.)
° Layout
You have to make sure the shifter is pitch-matched or bitslice width-matched with rest of your datapath structures. You may find the layout of the shifter to be metal intensive, especially if you are implementing the pass gate implementation, so think first of how to run metal lines.
° Design Verification
Run DRC, LVS on the entire shifter.
° Analog Simulation
Find the delays through your shifter using SpectreS.

Requirements

° Schematic of the entire shifter, including the drivers/buffers (but not necessarily the decoder).
° SpectreS traces showing how you calculated delays.
° Layout of the shifter. This should include hierarchical layout of the drivers and buffers.
° DRC report.
° LVS report.
This should discuss the considerations that went into the choice of your shifter design and the floorplanning. You should also discuss the rise and fall delays and the critical path. Any other comments that you feel are relevant should be included.

Deadline

You need to turn in CAD5 by Thursday, November 10, 2009, 12:30 pm.