Design Project: 8-bit X 8-bit Serial Multiplier

Preliminary Instructions

A. J. Davis

University of Rhode Island, Kingston, RI 02881-0805, U.S.A.

Abstract:

The course project is to complete the design, layout and verification of a 8-bitx8-bit serial multiplier using the architecture described in section 2. The serial multiplier will be connected in a pad frame and made ready for design submission in the MOSIS 1.2μm double-poly CMOS process.

Primary grading will be based upon the team's performance in the following areas:

§1. Timing and Control (due 25 Nov).

§2. Pin description including test signals (25 Nov for preliminary pin description).

§3. Structure of the design/design approach.

§4. Layout; The tar zipped directory containing ALL magic & irsim files.

§5. Verification Test Plan/Verification.

§6. IC test plan.

§7. Team presentation to the class (4 Dec).

§8. Project write up (due 18 Dec).

1 Introduction

Each design team will be responsible the complete implementation of a 8-bitx8-bit multiplier. Your team will develop all of the control/timing necessary for the actual operation and 1^{st} Silicon die testing. Thus, you will need additional control circuitry in order to test the die in the lab when fabrication has been completed.

You will have flexibility in deciding how to implement the multiplier architecture (described in the next section) and in the timing/control required

for pre-fab verification and post-fab testing. You are expected to use of the ELE447 library. ALL of your primitive gates and single-bit logic cells must be loaded from this library as subcells.

2 Serial Multiplier Description

The multiplier architecture will be implemented using the bit-serial implementation described in figure 1. The multiplication is implemented by performing m-additions. The clock must run m times faster than the multiplier output word rate.

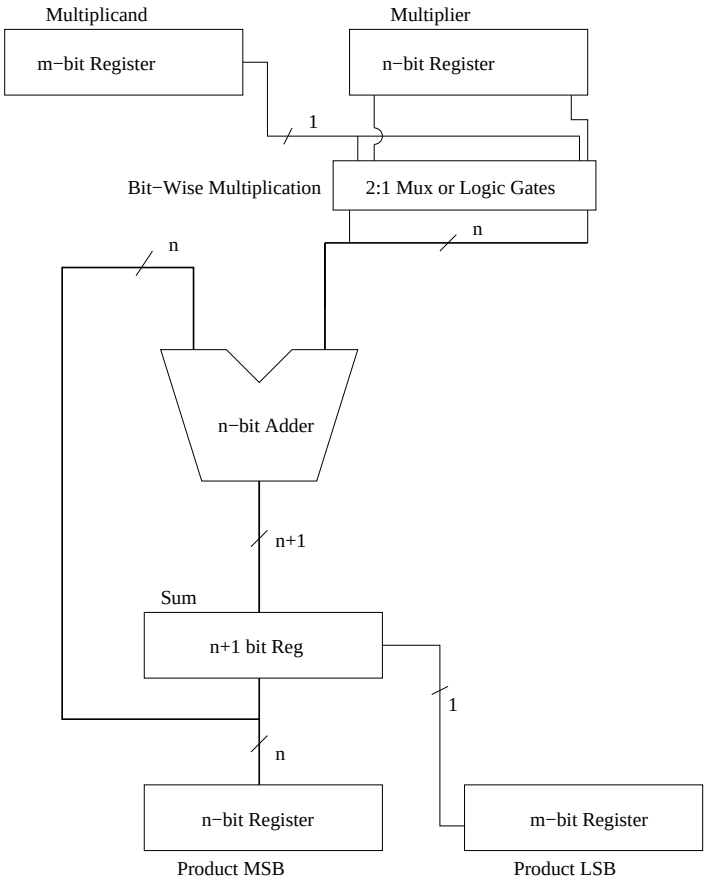

Figure 1. m-bitxn-bit Serial Multiplier.

On each clock step, one-bit from the m-bit multiplicand is shifted and multiplied by the multiplier word. The result is added to the existing partial sum. The least significant bit is shifted out of the sum register and into the LSB output register. This provides the shift and add. This operation is repeated m times. Thus, m-clock cycles are required to produce the m+n bit result.

3 Ground Rules

1) The multiplier will use the architecture in figure 1 with m=n=8.
2) The design and layout MUST be hierarchical. Flat designs will NOT be accepted. This project should have several hierarchical levels including:
   a) Design the data path using a 1-bit architecture which can be arrayed (see magic :array cmd).
   b) The 1-bit data path will be verified/tested.
3) You are expected to use synthesis, layout and verification techniques practiced during the lab sequence.
4) The primitive single-bit cells, e.g. logic gates, multiplexers, decoders, adders, etc. MUST be from the ELE447 cell library (no modification to these cells will be permitted).
5) It is up to your team to decide on the implementation of the blocks in figure 1, appropriate control/timing and the pin out.
6) Although you are asked to turn in a preliminary pin out on 1 December. You must understand that you are certainly allowed to some of the details as your project matures.

4 Project Instructions

1) Develop the timing for the serial multiplier. You will need to have a sketch of important timing signals by 1 December.
2) Prepare the top level design. Have the preliminary pin-out for the 40 DIP completed on 1 December.
3) Other information will be requested as time progresses.
4) Try to be creative and, despite the time limit, please attempt to have a good time.

5 Reporting

1) A report format will be provided at a later time.
2) Tar zip all magic files, *.ext files and irsim files into a file named: $ProjectYourNames.tar.gz$.