1 Introduction

You will design a custom Application Specific Integrated Circuit (ASIC) implementation of an N x N array multiplier. The ASIC will be fabricated in AMI 0.5μm CMOS technology available through MOSIS.

2 Background and Requirements

You already know how to multiply two binary numbers. Figure 1[1] shows an example of multiplication of two unsigned 6-bit numbers. The first number (12_{10}) is the *multiplicand* and second number (5_{10}) is the *multiplier*. Six partial products (rows) are generated which when summed column wise yields the product (60_{10}).

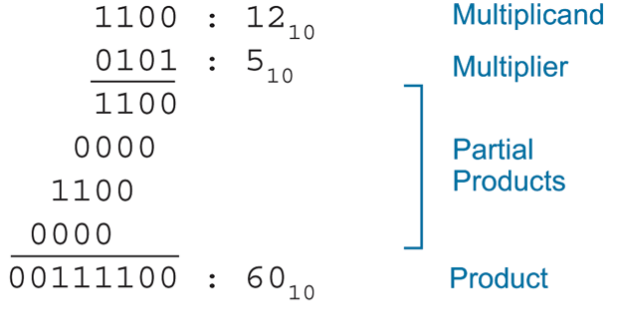

FIG 10.67 Multiplication example

Figure 1: **Binary Multiplication - An Example**

If the multiplicand is represented by Y = { y_{M-1}, y_{M-2}, ... , y_1, y_0 }, and multiplier as X = { $x_{N-1}, x_{N-2}, \ldots, x_1, x_0$ }, then the product, P, is given by Equation 1. The inner summation results in partial products while the outer one sums the partial products. This can be depicted as in Figure 2. Note that $M = N$ in this project.

[1]Note the figures are from Weste and Harris, 3rd edition.

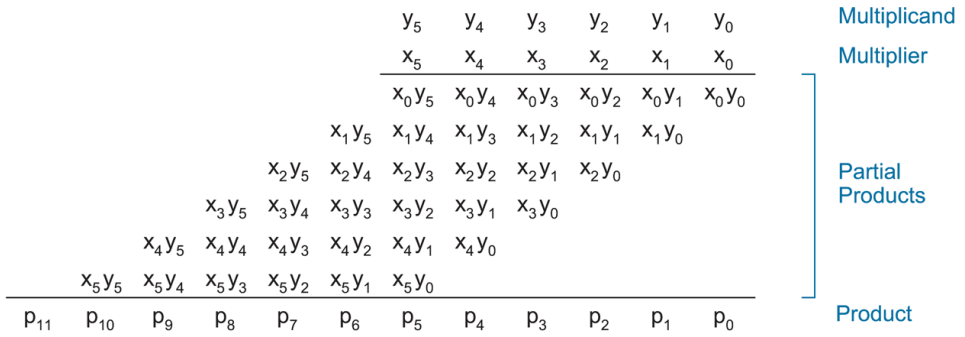

FIG 10.68 Partial products

Figure 2: **Partial Products**

$$P = \sum_{j=0}^{M-1} y_j 2^j \sum_{i=0}^{N-1} x_i 2^i = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} x_i y_j 2^{i+j} \tag{1}$$

Consider a bit-sliced implementation of a 4-bit multiplier as shown in Figure 3. The multiplicand is fed vertically, while the multiplier is fed horizontally. There are two bit slices used in this design; the first one (rectangle with sharp corners) multiplies two bits (i.e., ANDs A and B) and then performs a sum of three bits (AB, S_{in}, C_{in}). The second slice (with rounded corners) performs a three bit sum (A, B, and C_{in}). This implementation uses Carry Save Adders (CSA) along the rows. A CSA, unlike a ripple carry adder, saves the sums and passes them to the CSA in the succeeding row. The critical path for the circuit is shown by a dashed line. The sum of delays of module instances along the critical path will be the worst-case delay and thus will dictate how fast we can multiplier two numbers.

Figure 4 shows a rectangular layout of the array multiplier.

Read Section 11.9 and 11.9.1 (pages 476 – 479) in Weste and Harris text book (4th edition) for a detailed discussion on the array mutliplier.

2.1 Design Requirements

Your team is called upon to layout an N x N array multiplier in a given area (approximately $0.8mm^2$). The following are the design requirements and objectives:

1. Implement a rectangular layout.
2. The inputs X and Y are fed serially to the multiplier.
3. Both X and Y should be registered.
4. You need to maximize N.

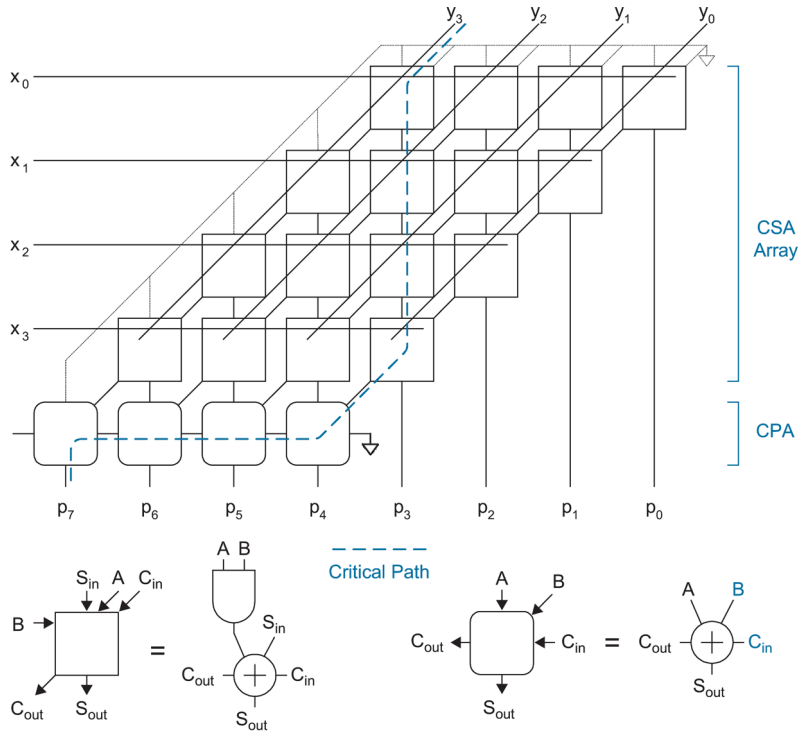

FIG 10.70 Array multiplier

Figure 3: **A 4-bit Array Mutliplier**

5. Product P shouald be registered and clocked out serially.
6. The three registers are clocked by an external clock signal.

2.2 Design Goals and Constraints:

- **Functionality:** Your design should work for atleast 8 (eight) input cases.
- **Area:** You are limited by the chip area of approximately $900\mu m$ x $900\mu m$. You must incorporate your layout in the TinyChip Padframe provided by MOSIS and made available to you on Canvas.
- **Performance:** As such there is no minimum clock constraint, however, you should determine what is the fastest clock speed your design will work at.
- **I/O Pin Constraint:** TinyChip has 40 pads. You will be provided with a pin-mapping diagram that you should adhere to strictly.
- **Testability:** The ASIC can be put in test mode (by setting input TEST=1). We should be able to test the X, Y, and P registers in a scan-chain mode. A scan chain is formed with the three registers cascaded to form one long serial register.

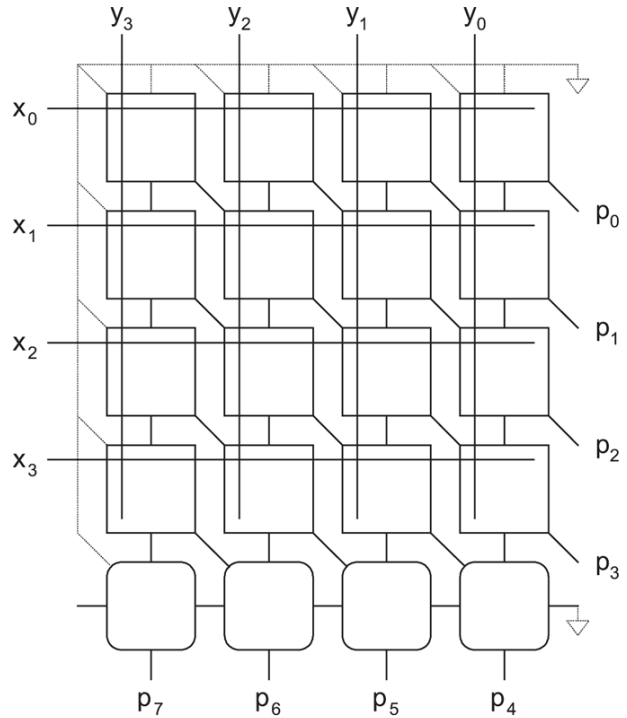

FIG 10.71 Rectangular array multiplier

Figure 4: **Rectangular Array Mutliplier**

2.3 Design Validation

You must validate your design by simulation using Spectre. You need to show the simulation results for two modes:

- **Normal mode:** You will be provided with 8 test cases for which your design should work correctly.
- **Test Mode:** You will be provided with 2 test cases for which your design should work correctly.