VHDL Assignments
Assignment 1

Dr. Shubhajit Roy Chowdhury

Background

In this assignment, you will be designing a 4-bit by 8-bit multiplier. Once completed, you will then have in your "box" of designs this multiplier function, as well as the elementary design blocks.

Before stating the multiplier design problem to be solved, a comment is appropriate regarding the already completed 8-bit adder and 8-bit register designs. If you find that you need a 16-bit adder or a 32-bit adder, you should see that you could use your VHDL code for the 8-bit adder and simply change the entity statement and architecture statement to create the 16-bit or 32-bit adders. In this manner, you could easily add to your "box" of components by simply making changes in the entity/architecture codes from existing components. This also obviously holds for design of 16-bit, 32-bit, or whatever size registers needed. In the case of the multiplier design done here, it would be easy, given the VHDL entity/architecture code for the 4x8-bit multiplier, to adapt that code for an 8x16-bit multiplier. I am keeping the size (4x8) relatively small so that this design is not overly complicated.

Figure 1 illustrates the multiplication of an 8-bit binary number (A7 A6 A5 ...A0) by a 4-bit number (B3 B2 B1 B0) where A0 and B0 are the least significant bits of those numbers.

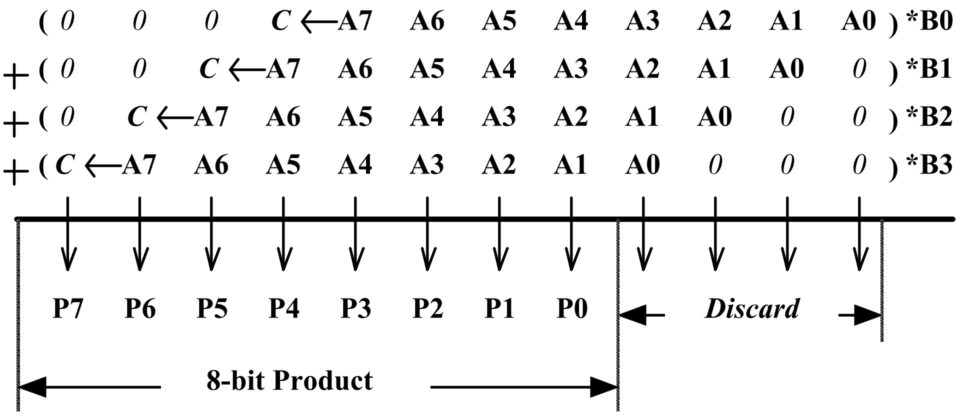
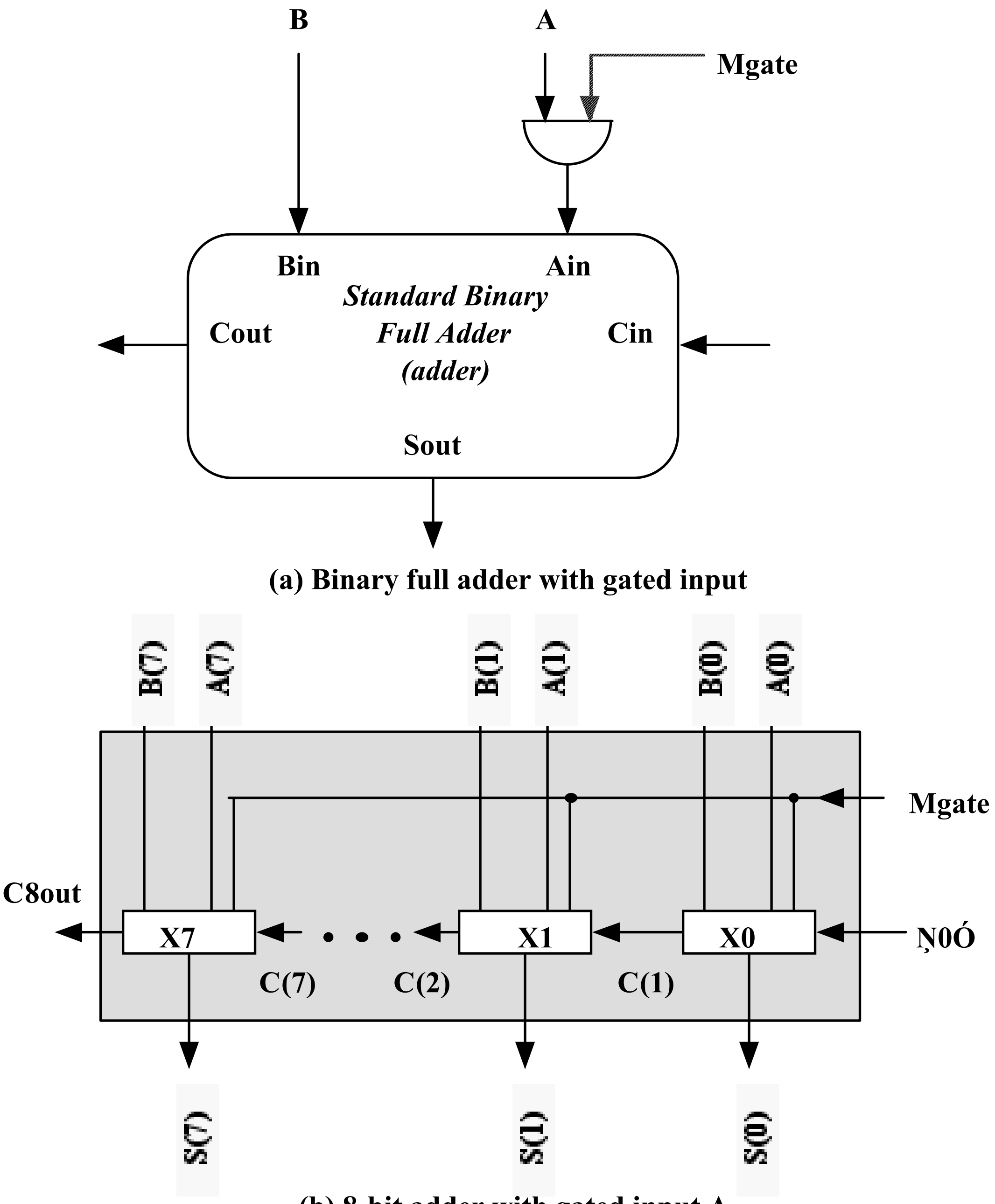

(0 0 0 C ⟵A7 A6 A5 A4 A3 A2 A1 A0) *B0
+ (0 0 C ⟵A7 A6 A5 A4 A3 A2 A1 A0 0) *B1
+ (0 C ⟵A7 A6 A5 A4 A3 A2 A1 A0 0 0) *B2
+ (C ⟵A7 A6 A5 A4 A3 A2 A1 A0 0 0 0) *B3
P7 P6 P5 P4 P3 P2 P1 P0
Discard
8-bit Product

Figure 1: Illustration of multiplication of 8-bit number by 4-bit number giving 8-bit result

In this figure, the 8-bit number is shifted one place to the left and multiplied by the next bit of the 4-bit number. To create a square set of numbers, zeros have been added both at the right and left hand sides. The C entry corresponds to the carry from its line, that carry being added into the next column. The overall product, including the final output carry, consists of 8+4+1=13 bits. In the example in Figure 1, I have chosen to keep the 8 most significant binary bits. Details of other choices are beyond the needs for this course. Figure 1 shows the need to "gate" the 8-bit number, i.e., multiply the 8-bit number by the corresponding bits of the 4-bit number as the sum of partial products proceeds. This function can easily be added to our earlier adder design, as illustrated in Figure 2.



(a) Binary full adder with gated input

(b) 8-bit adder with gated input A

Figure 2. (a) Addition of gate to binary full adder. (b) Design of gated 8-bit adder

Using 8-bit gated adders such as shown in Figure 2b, an 8x4 bit multiplier can be built as shown in the Figure below, which implements the additions shown in Figure 1.

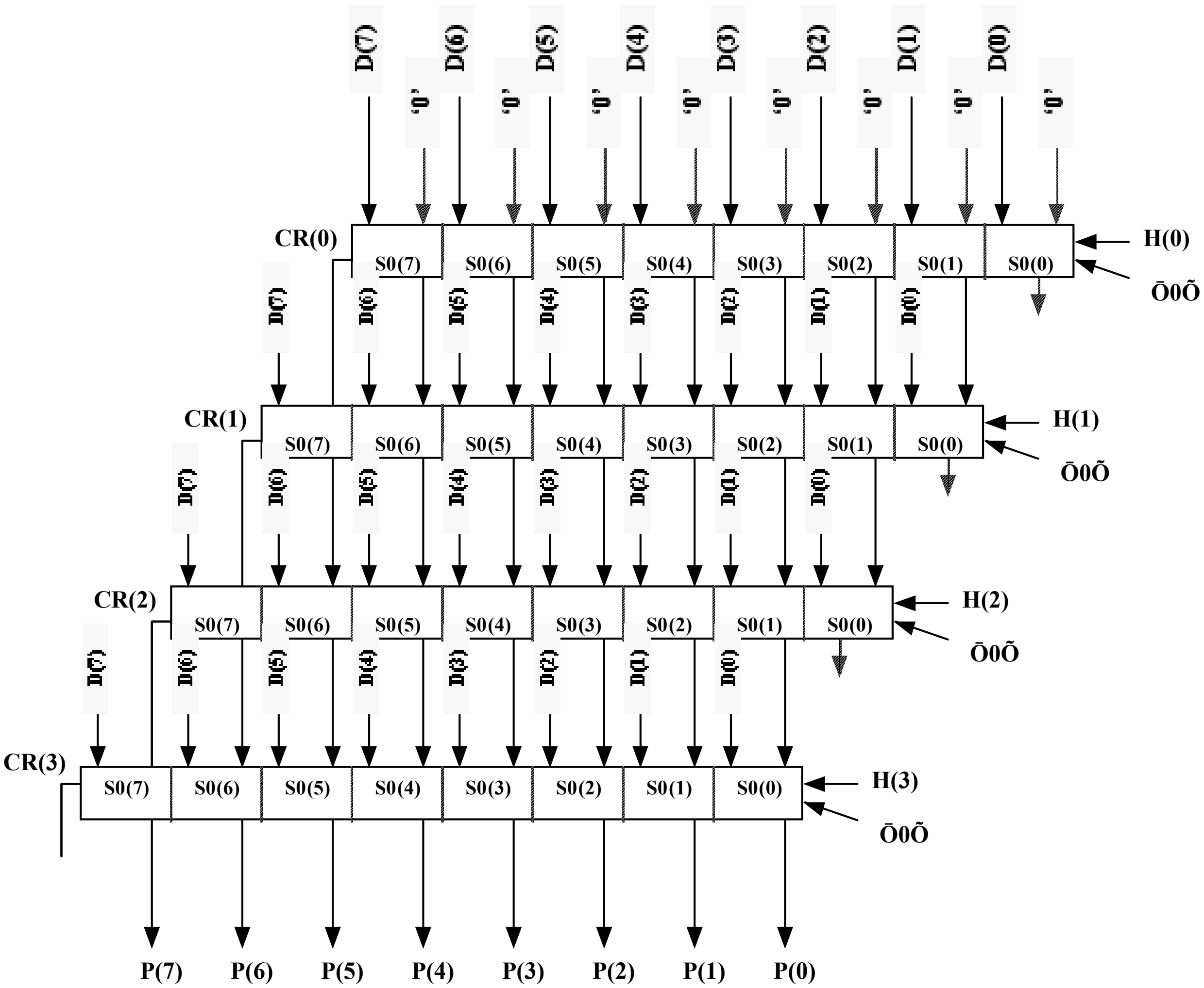


Figure 3: 4x8 multiplier design

The inputs A(i) of the adder in Figure 2b are the bits of the 8-bit input data word D in Figure 3. The inputs B(i) of the adder in Figure 2b become the outputs of the previous adder stage in Figure 3 (with zeros being entered for the first adder). The "Mgate" input signal to the adder in Figure 2b are the binary bits H(i) of the 4-bit multiplier in Figure 3.

Assignment Statement

Design of the multiplier requires that we complete the design of the 8-bit gated adder in Figure 2b. Complete the following steps.

1. Design the gated binary full adder in Figure 2a. When doing this, use your earlier design of the basic binary full adder as one component and simply add the AND gate to create the structural model of the

gated binary full adder. The two components of the gated binary full adder will be your earlier simple binary full adder and the AND gate. Do the following.

a. Design the AND gate by writing the entity definition and using a behavioral model of the AND gate. Call the entity AND2.

b. Give the entity description for the gated binary full adder in Figure 2a. Call this GBFA

c. Write the structural architectural model for the gated binary full adder using its two components (the earlier binary full adder and the AND2 gate).

2. Design the gated 8-bit binary adder in Figure 2b. Do the following.

   a. Give the entity description for the gated 8-bit binary adder. Call this G8BFA.

   b. Write the structural architectural model of the G8BFA, using the GBFA gated binary full adders as components.

Having completed the VHDL description for the G8BFA, you next design the 4x8 multiplier shown in Figure 3 above. In particular,

3. Design the 4x8 multiplier. Do the following.

   a. Give the entity description for the 4x8 multiplier. Call this Mult4x8.

   b. Write the structural architectural model of the Mult4x8 component using the G8BFA components. This will require that you use some care when wiring one G8BFA to the next. Use bit-vectors to allow use of indices. As you work through this, you will see the pattern of indexing that emerges.

   c. Simulate the multiplier for a couple of non-trivial pairs of inputs to verify that it is generally correct.