Baugh-Wooley Multiplier

1 Objectives

- Understand the Baugh-Wooley multiplication algorithm for 2's complement data.
- Design a 2's complement multiplier based on the Baugh-Wooley algorithm.
- Employ hierarchical design technique.
- Simulate the multiplier performance.

2 Introduction

We start by explaining the Baugh-Wooley multiplication algorithm.

2.1 Multiplying two 2's compliment numbers

The Baugh-Wooley multiplication algorithm is an efficient way to handle the sign bits. This technique has been developed in order to design regular multipliers, suited for 2's-complement numbers. Let us consider two n-bit numbers, A and B, to be multiplied. A and B can be represented as

$$A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i \quad (1)$$

$$B = -b_{n-1}2^{n-1} + \sum_{i=0}^{n-2} b_i 2^i \quad (2)$$

Where the a_i's and b_i's are the bits in A and B, respectively, and a_{n-1} and b_{n-1} are the sign bits.

The product, $P = A \times B$, is then given by the following equation:

$$\begin{aligned} P &= A \times B \\ &= \left(-a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i\right) \times \left(-b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} b_j 2^j\right) \\ &= a_{n-1}b_{n-1}2^{2n-2} + \sum_{i=0}^{n-2}\sum_{j=0}^{n-2} a_i b_j 2^{i+j} \\ &\quad -2^{n-1}\sum_{i=0}^{n-2} a_i b_{n-1} 2^i - 2^{n-1}\sum_{j=0}^{n-2} a_{n-1} b_j 2^j \quad (3) \end{aligned}$$

Equation (3) indicates that the final product is obtained by subtracting the last two *positive terms* from the first two terms.

2.2 Baugh-Wooley multiplication algorithm

Rather than do a subtraction operation, we can obtain the 2's complement of the last two term and add all terms to get the final product.

The last two terms are $n-1$ bits each that extend in binary weight from position 2^{n-1} up to 2^{2n-3}. On the other hand, the final product is $2n$ bits and extends in binary weight from 2^0 up to 2^{2n-1}.

We pad each of the last two terms in Equation (3) with zeros to obtain a $2n$-bit number to be able to add them to the other terms. The padded terms extend in binary weight from 2^0 up to 2^{2n-1}.

Assuming X is one of the last two terms we can represent it with zero padding as

$$X = -0 \times 2^{2n-1} + 0 \times 2^{2n-2} + 2^{n-1} \sum_{i=0}^{n-2} x_i 2^i + \sum_{j=0}^{n-2} 0 \times 2^j \tag{4}$$

The above equation gives the *value* of X due to the fact that a negative value is associated with the MSB.

When we *store* X in a register, the negative sign at MSB is not used since X is stored as a *binary pattern*. Thus partial product X is, therefore, represented by

bit position	$2n-1$	$2n-2$	$2n-3$	$2n-4$	$\cdots$	n	$n-1$	$n-2$	$n-3$	$\cdots$	0
bit value	0	0	x_{n-2}	x_{n-3}	$\cdots$	x_1	x_0	0	0	$\cdots$	0

The two's complement of X is obtained by complimenting all bits in the above equation and adding '1' at the LSB:

bit position	$2n-1$	$2n-2$	$2n-3$	$2n-4$	$\cdots$	n	$n-1$	$n-2$	$n-3$	$\cdots$	0
bit value	1	1	x_{n-2}	x_{n-3}	$\cdots$	x_1	x_0	1	1	$\cdots$	$1+1$

Adding the '1' at LSB will result in the new pattern for $-X$ as

bit position	$2n-1$	$2n-2$	$2n-3$	$2n-4$	$\cdots$	n	$n-1$	$n-2$	$n-3$	$\cdots$	0
bit value	1	1	x_{n-2}	x_{n-3}	$\cdots$	x_1	x_0+1	0	0	$\cdots$	0

Assuming the last two terms are expressed as X and Y, then adding $-X$ to $-Y$ amounts to adding the following two bit patterns:

bit position	$2n-1$	$2n-2$	$2n-3$	$2n-4$	$\cdots$	n	$n-1$	$n-2$	$n-3$	$\cdots$	0
$-X$	1	1	x_{n-2}	x_{n-3}	$\cdots$	x_1	x_0+1	0	0	$\cdots$	0
$+(-Y)$	1	1	y_{n-2}	y_{n-3}	$\cdots$	y_1	y_0+1	0	0	$\cdots$	0

The '1' pattern at most significant bits transforms into

bit position	$2n-1$	$2n-2$
	1	1
+	1	1
	1	0

Similarly, the '1' pattern at position $n-1$ becomes

bit position	n	$n-1$
		1
+		1
	1	0

The final product $P = A \times B$ in Equation (3) becomes:

$$\begin{aligned} P &= a_{n-1}b_{n-1}2^{2n-2} + \sum_{i=0}^{n-2}\sum_{j=0}^{n-2} a_i b_j 2^{i+j} + \\ & 2^{n-1}\sum_{i=0}^{n-2}\overline{b_{n-1}a_i}2^i + 2^{n-1}\sum_{j=0}^{n-2}\overline{a_{n-1}b_j}2^j + \\ & -2^{2n-1} + 2^n \end{aligned} \tag{5}$$

Let us assume that A and B are 4-bit binary numbers, then the product $P = A \times B$ is 8-bits long and is given by

$$\begin{aligned} P &= a_3b_3 2^6 + \sum_{i=0}^{2}\sum_{j=0}^{2} a_i b_j 2^{i+j} + \\ & 2^3\sum_{i=0}^{2}\overline{b_3 a_i}2^i + 2^3\sum_{j=0}^{2}\overline{a_3 b_j}2^j + \\ & -2^7 + 2^4 \end{aligned} \tag{6}$$

Figure 1 shows the implementation of the Baugh-Wooley multiplier.

The basic Baugh-Wooley cells are shown in Figure 2.

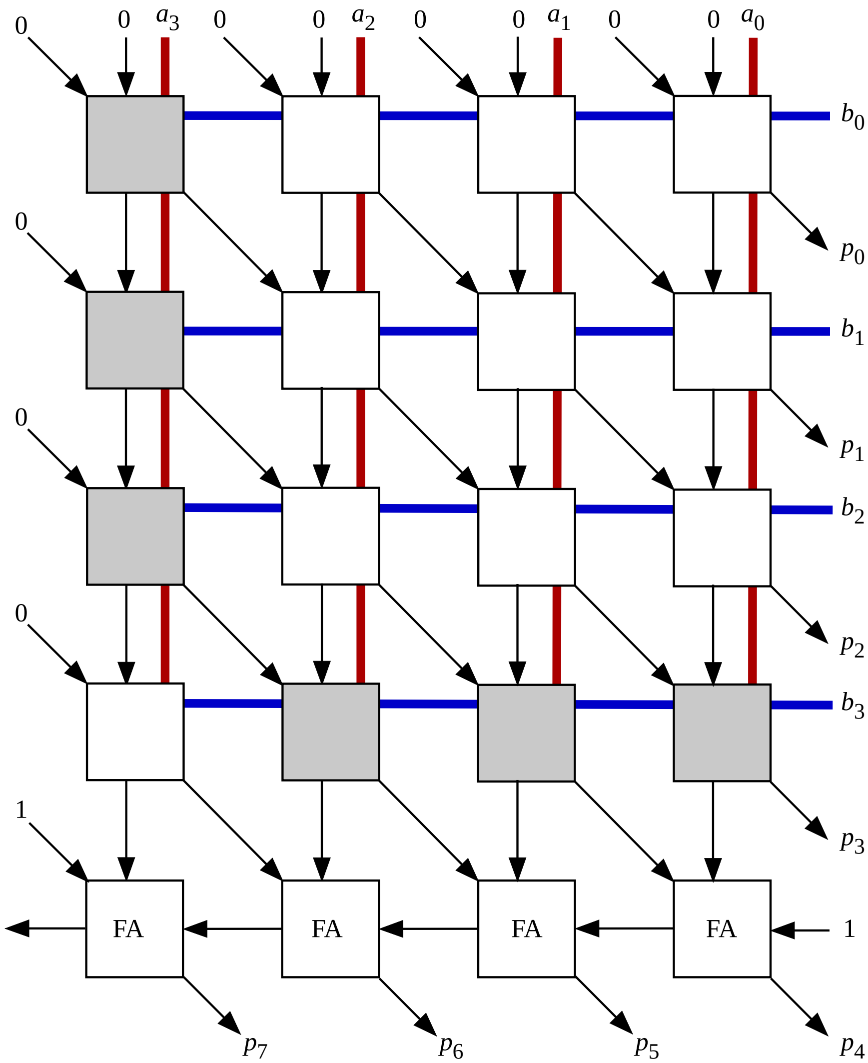
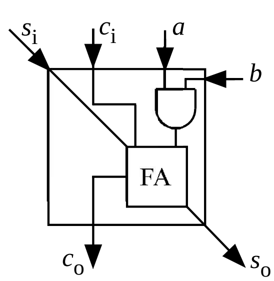
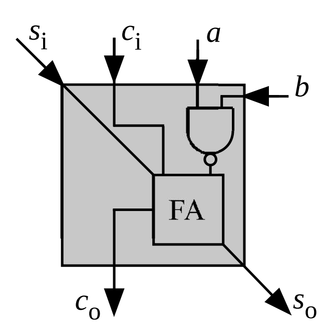


Figure 1: Block diagram of 4×4 Baugh-Wooley multiplier



(a) Baugh-Wooley multiplier white-cell



(b) Baugh-Wooley multiplier gray-cell

Figure 2: Baugh-Wooley multiplier basic cell construction

3 Pre-Lab Report

1. Estimate the delay of the 4×4 Baugh-Wolley multiplier. The gate delays are listed in Table 1.

2. Draw a block diagram of an 8×8 Baugh-Wooley multiplier.

Table 1: CMOS gate delays and areas normalized relative to an inverter.

Gate	**Delay**	**Area**	**Comment**
Inverter	1	1	Minimum delay
2-input NOR	1	3	More area to produce delay equal to that of an inverter
2-input NAND	1	3	More area to produce delay equal to that of an inverter
2-input AND	2	4	Composed of NAND followed by inverter
2-input OR	2	4	Composed of NOR followed by inverter
2-input XOR	3	11	Built using inverters and NAND gates
n-input OR	2	$n/3 + 2$	Uses saturated load
n-input AND	3	$4n/3 + 2$	Uses n-input OR preceded by inverters

4 Project Requirements

In this project you are required to design, model, and simulate a 4×4 multiplier based on the Baugh-Wooley algorithm.

1. The 4×4 Baugh-Wooley multiplier is to be hierarchically designed using 1-bit full adders and the two types of cells shown in Figure 1.

2. The delays of the components should be assigned with the help of Table 1 and assuming the delay of an inverter is 1 ns.

3. The multiplier has two inputs a and b of type `signed` representing the multiplier and multiplicand; and one output p of type `signed` representing the product.

4. Write a testbench to verify the operation of the multiplier. The testbench should try different number values and signs. Simulate the behavior of the multiplier using the testbench you developed.

5 Lab Report

1. Refer to the lab report grading scheme for items that must be present in your report.
2. Estimate the area of a 4×4 Baugh-Wolley multipliers. The gate areas are listed in Table 1.
3. Find the delay of the 4×4 multiplier using the waveform you got from simulation and comment on your results.