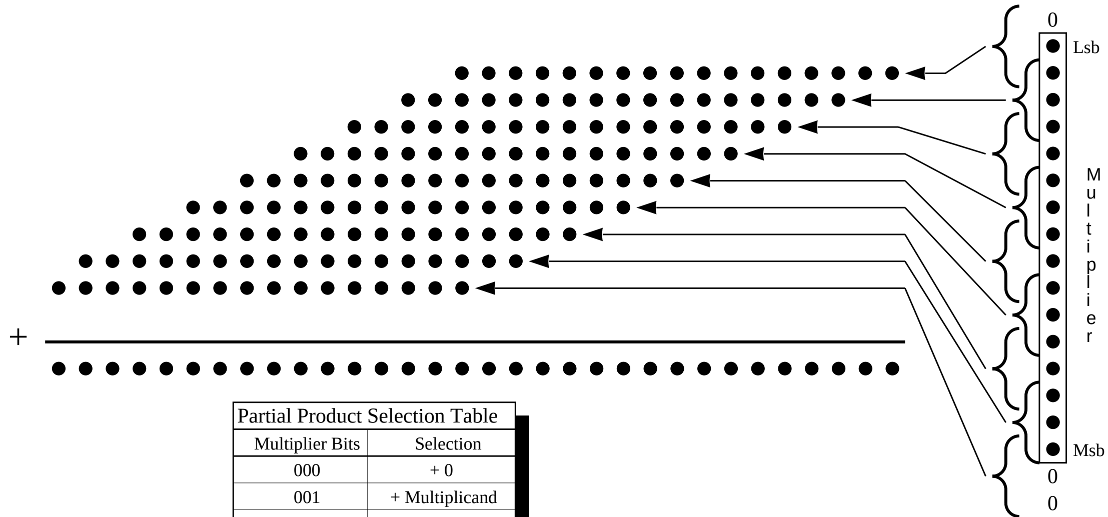
Appendix A

Sign Extension in Booth Multipliers

This appendix shows how to compute the sign extension constants that are needed when using Booth's multiplication algorithm. The method will be illustrated for the 16x16 bit Booth 2 multiplication example given in Chapter 2. Once the basic technique is understood it is easily adapted to the higher Booth algorithms and also to the redundant Booth method of partial product generation. The example will be that of an unsigned multiplication, but the final section of this appendix will discuss the modifications that are required for signed arithmetic.

A.1 Sign Extension for Unsigned Multiplication

The partial products for the 16x16 multiply example, assuming that all partial products are positive, are shown in Figure A.1. Each partial product, except for the bottom one, is 17 bits long, since numbers as large as 2 times the multiplicand must be dealt with. The bottom partial product is 16 bits long, since the multiplier must be padded with 2 zeroes to guarantee a positive result. Figure A.2 shows the partial products if they all happen to be negative. Using 2's complement representation, every bit of the negated partial products is complemented, including any leading zeroes, and 1 is added at the least significant bit. The bottom partial product is never negated, because the 0 padding assures that it is always positive. The triangle of 1's on the left hand side can be summed to produce Figure A.3, which is exactly equivalent to the situation shown in Figure A.2. Now, suppose that a



Partial Product Selection Table	
Multiplier Bits	Selection
000	+ 0
001	+ Multiplicand
010	+ Multiplicand
011	+ 2 x Multiplicand
100	-2 x Multiplicand
101	- Multiplicand
110	- Multiplicand
111	- 0

Figure A.1: 16 bit Booth 2 multiplication with positive partial products

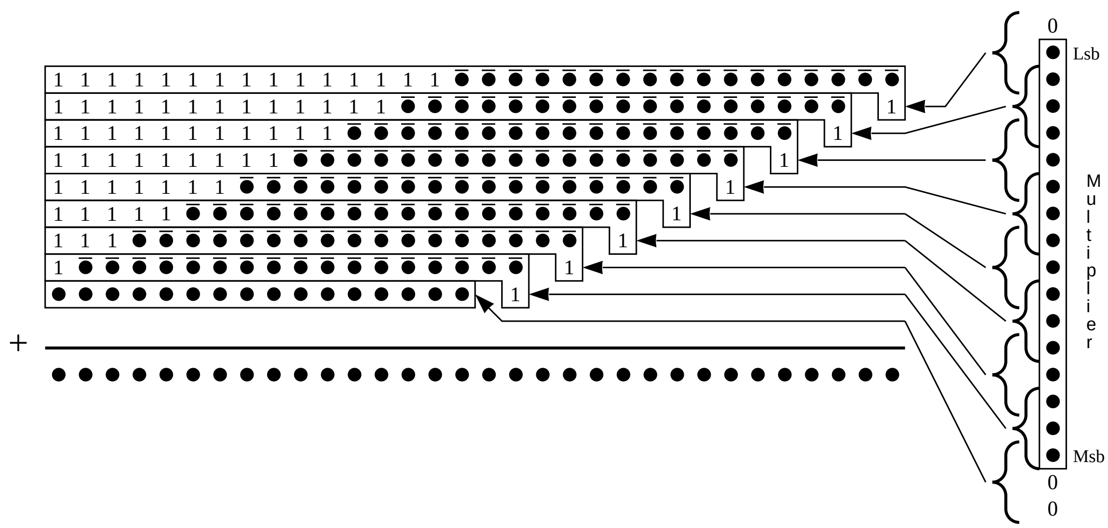


Figure A.2: 16 bit Booth 2 multiplication with negative partial products

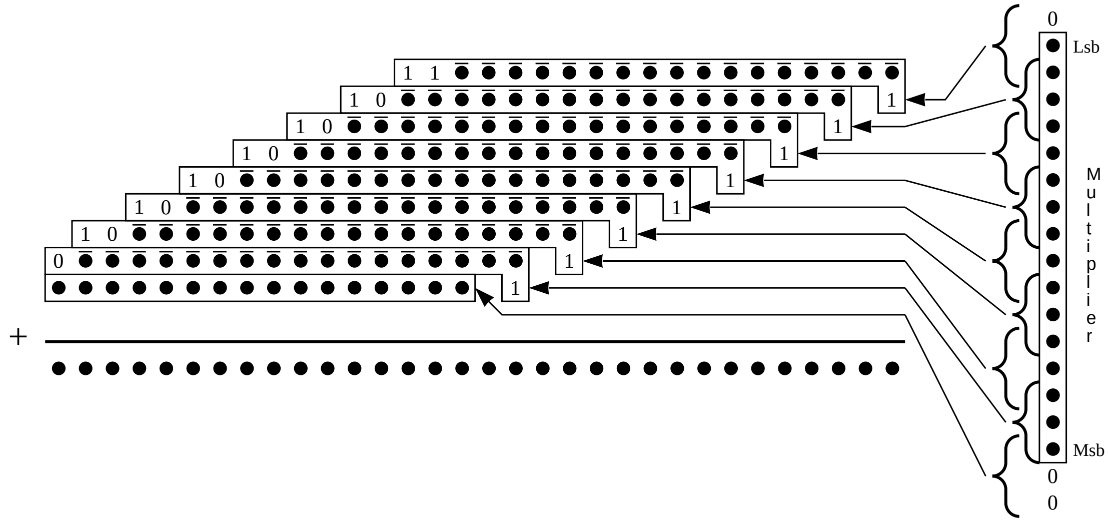
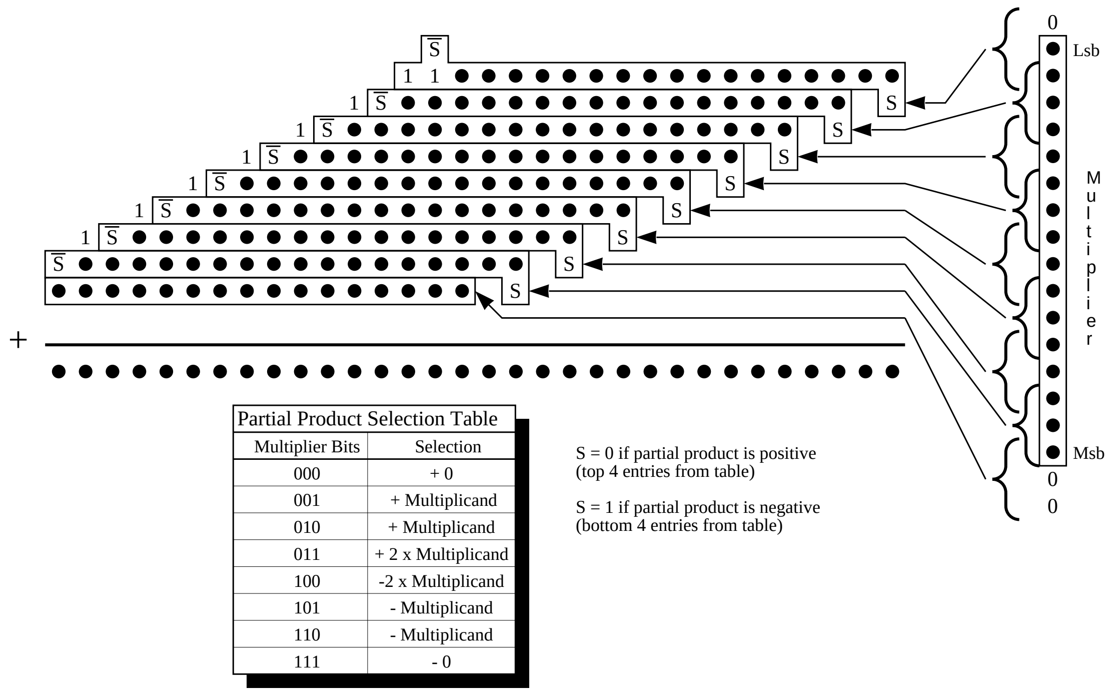


Figure A.3: Negative partial products with summed sign extension

particular partial product turns out to not be negative. The leading string of ones in that particular partial product can be converted back to a leading of zeroes, by adding a single 1 at the least significant bit of the string. Referring back to the selection table shown in Figure A.1, a partial product is positive only if the most significant bit of the select bits for that partial product is 0. Additionally, a 1 is added into the least significant bit of a partial product only if it is negative. Figure A.4 illustrates this configuration. The $\overline{S}$ bits represent the 1's that are needed to clear the sign extension bits for positive partial products, and the S bits represent the 1's that are added at the least significant bit of each partial product if it is negative.

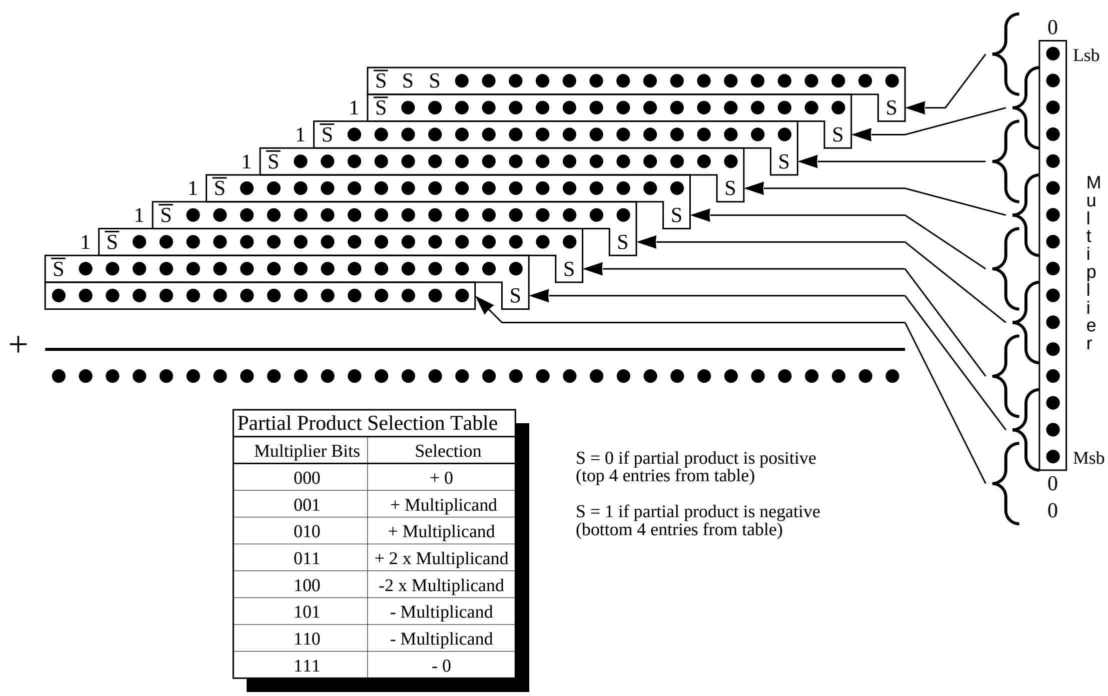
A.1.1 Reducing the Height

Finally, the height (maximum number of items to be added in any one column) of the dot diagram in Figure A.4 can be reduced by one by combining the $\overline{S}$ term of the top partial product with the two leading ones of the same top partial product, which gives the final result, shown in Figure A.5 (this is the same as Figure 2.4).



Partial Product Selection Table	
Multiplier Bits	Selection
000	+ 0
001	+ Multiplicand
010	+ Multiplicand
011	+ 2 x Multiplicand
100	-2 x Multiplicand
101	- Multiplicand
110	- Multiplicand
111	- 0

Figure A.4: Complete 16 bit Booth 2 multiplication



Partial Product Selection Table	
Multiplier Bits	Selection
000	+ 0
001	+ Multiplicand
010	+ Multiplicand
011	+ 2 x Multiplicand
100	-2 x Multiplicand
101	- Multiplicand
110	- Multiplicand
111	- 0

Figure A.5: Complete 16 bit Booth 2 multiplication with height reduction

A.2 Signed Multiplication

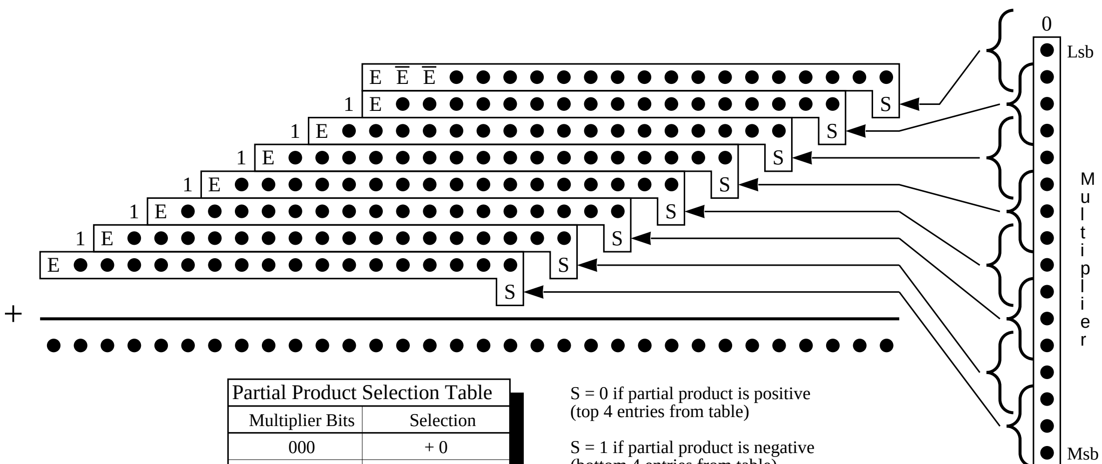
The following modifications are necessary for 2's complement, signed multiplication.

- The most significant partial product (shown at the bottom in all of the preceeding figures), which is necessary to guarantee a positive result, is not needed for signed multiplication. All that is required is to sign extend the multiplier to fill out the bits used in selecting the most significant partial product. For the sample 16x16 multiplier, this means that one partial product can be eliminated.

- When $\pm$Multiplicand (entries 1,2,5 and 6 from the partial product selection table) is selected, the 17 bit section of the effected partial product is filled with a sign extended copy of the multiplicand. This sign extension occurs before any complementing that is necessary to obtain $-$Multiplicand.

- The leading 1 strings, created by assuming that all partial products were negative, are cleared in each partial product under a slightly different condition. The leading 1's for a particular partial product are cleared when that partial product is positive. For signed multiplication this occurs when the multiplicand is positive and the multiplier select bits chooses a positive multiple, and also when the multiplicand is negative and the multiplier select bits choose a negative multiple. A simple EXCLUSIVE-NOR between the sign bit of the multiplicand and the high order bit of the partial product selection bits in the multiplier generates the one to be added to clear the leading 1's correctly.

The complete 16x16 signed multiplier dot diagram is shown in Figure A.6



Partial Product Selection Table	
Multiplier Bits	Selection
000	+ 0
001	+ Multiplicand
010	+ Multiplicand
011	+ 2 x Multiplicand
100	-2 x Multiplicand
101	- Multiplicand
110	- Multiplicand
111	- 0

S = 0 if partial product is positive (top 4 entries from table)

S = 1 if partial product is negative (bottom 4 entries from table)

E = 1 if multiplicand is positive and partial product is positive, or if multiplicand is negative and partial product is negative.

E = 0 if multiplicand is positive and partial product is negative, or if multiplicand is negative and partial product is positive.

Figure A.6: Complete signed 16 bit Booth 2 multiplication