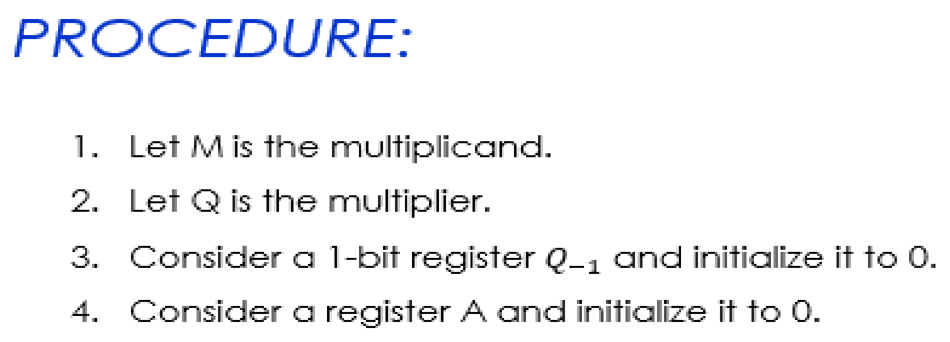
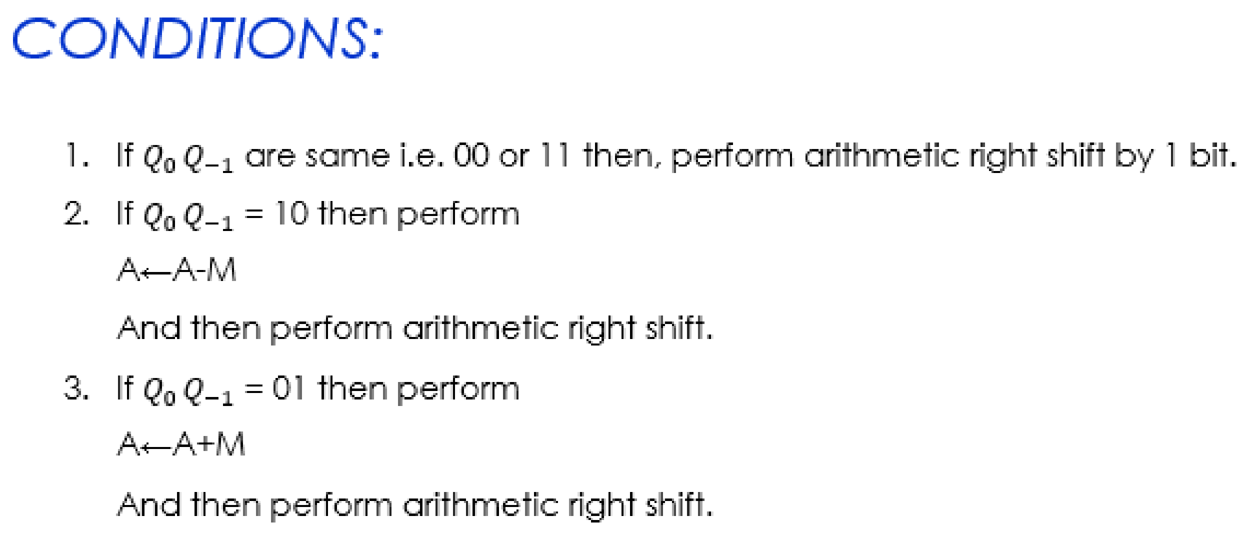
Booth's Multiplication Algorithm

Booth's algorithm is a multiplication algorithm that multiplies two signed binary numbers in 2's compliment notation.



PROCEDURE:

1. Let M is the multiplicand.
2. Let Q is the multiplier.
3. Consider a 1-bit register Q_{-1} and initialize it to 0.
4. Consider a register A and initialize it to 0.



CONDITIONS:

1. If $Q_0\,Q_{-1}$ are same i.e. 00 or 11 then, perform arithmetic right shift by 1 bit.
2. If $Q_0\,Q_{-1}$ = 10 then perform

   A←A-M

   And then perform arithmetic right shift.
3. If $Q_0\,Q_{-1}$ = 01 then perform

   A←A+M

   And then perform arithmetic right shift.

For example:

Consider two numbers 6 and 2 and we have to perform their multiplication by using Booth's algorithm.

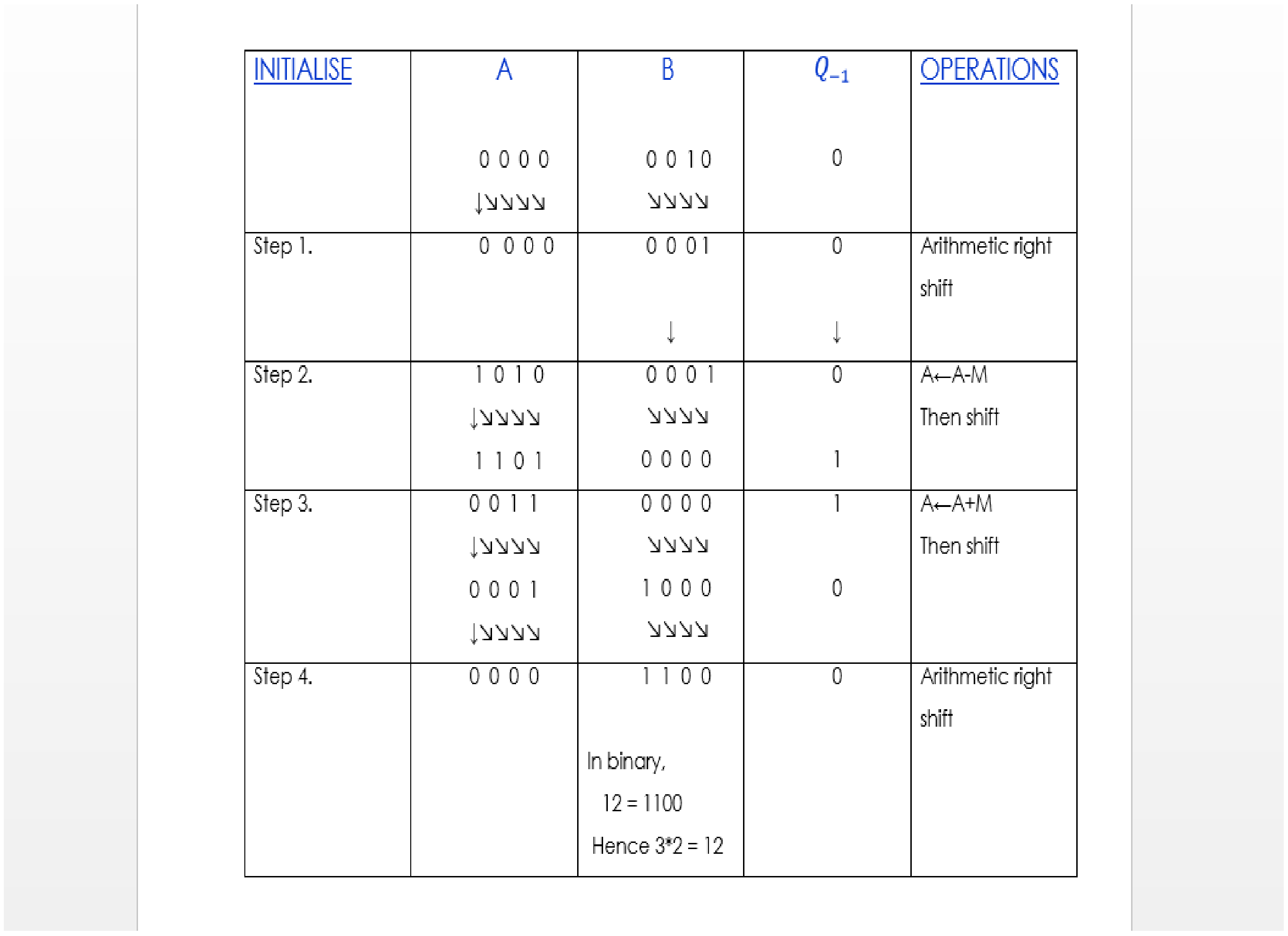
Here 6 is multiplicand (M) and 2 is multiplier (Q).

Now write 6 and 2 in binary form.

M = 6 = 0 1 1 0

Q = 2 = 0 0 1 0 (Q_3, Q_2, Q_1, Q_0)

Booth's algorithm calculates the product in n steps where n is the number of bits used to represent the numbers.

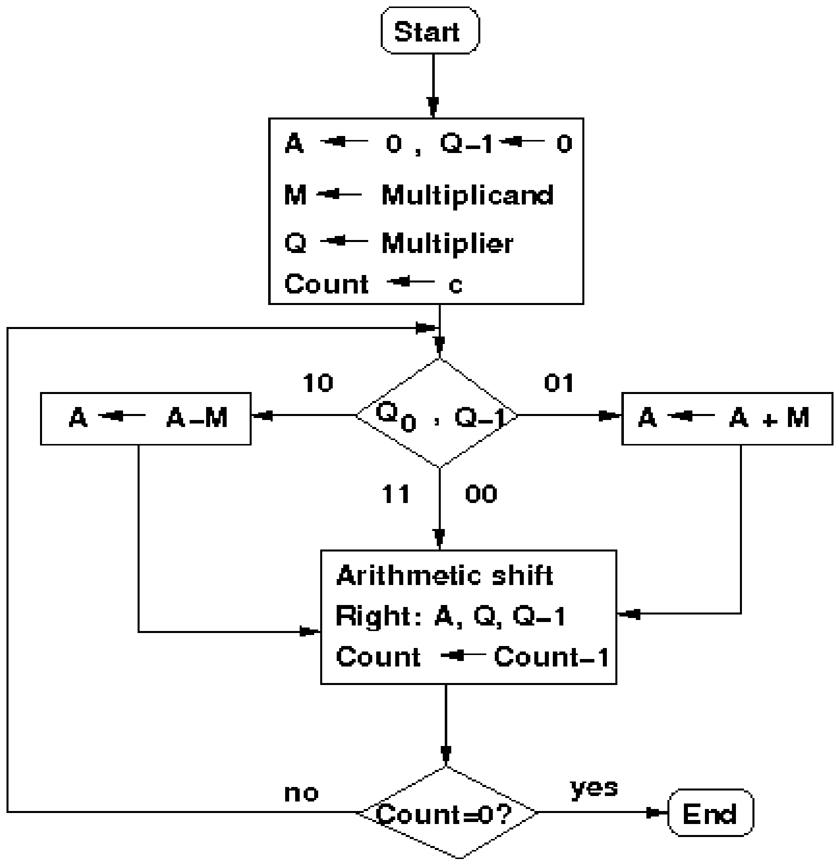


INITIALISE	A	B	Q_{-1}	OPERATIONS
	0 0 0 0 ↓↘↘↘↘	0 0 1 0 ↘↘↘↘	0	
Step 1.	0 0 0 0	0 0 0 1 ↓	0 ↓	Arithmetic right shift
Step 2.	1 0 1 0 ↓↘↘↘↘ 1 1 0 1	0 0 0 1 ↘↘↘↘ 0 0 0 0	0 1	A←A-M Then shift
Step 3.	0 0 1 1 ↓↘↘↘↘ 0 0 0 1 ↓↘↘↘↘	0 0 0 0 ↘↘↘↘ 1 0 0 0 ↘↘↘↘	1 0	A←A+M Then shift
Step 4.	0 0 0 0	1 1 0 0 In binary, 12 = 1100 Hence 3*2 = 12	0	Arithmetic right shift

Booth's Multiplier:

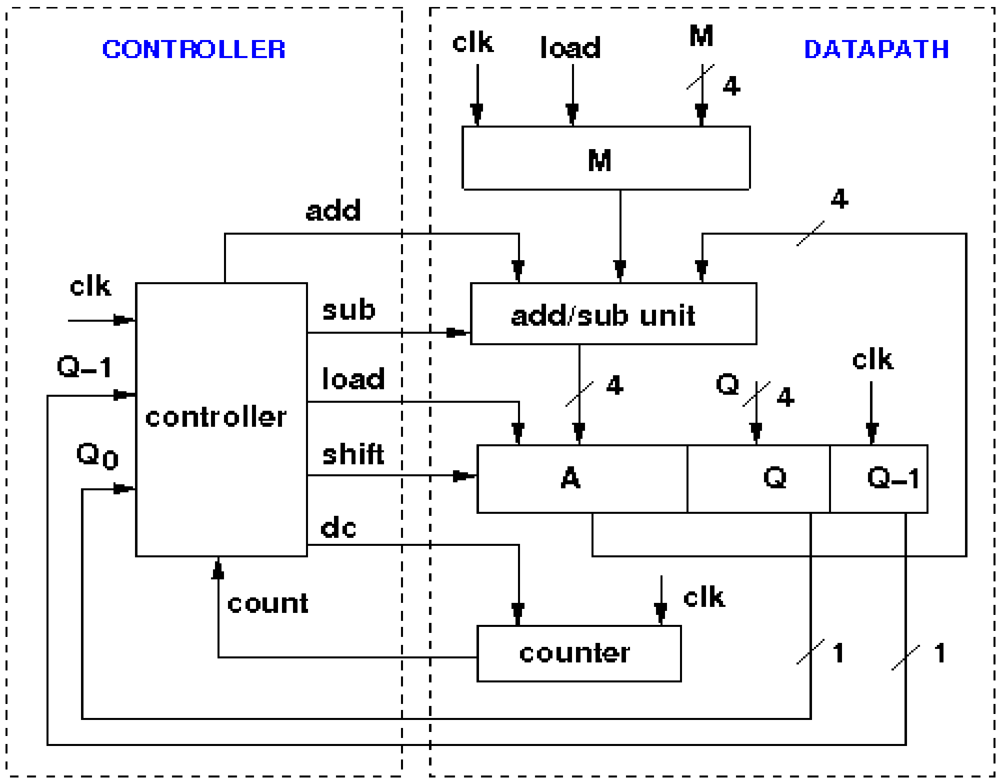
Booth's multiplication algorithm is an algorithm which multiplies 2 signed integers in 2's complement. The algorithm is depicted in the following figure with a brief description. This approach uses fewer additions and subtractions than more straightforward algorithms.

The multiplicand and multiplier are placed in the m and Q registers respectively. A 1-bit register is placed logically to the right of the LSB (least significant bit) Q_0 of Q register. This is denoted by Q_{-1}. A and Q_{-1} are initially set to 0. Control logic checks the two bits Q_0 and Q_{-1}. If the two bits are same (00 or 11) then all of the bits of A, Q, Q_{-1} are shifted 1 bit to the right. If they are not the same and if the combination is 10 then the multiplicand is subtracted from A and if the combination is 01 then the multiplicand is added with A. In both the cases results are stored in A, and after the addition or subtraction operation, A, Q, Q_{-1} are right shifted. The shifting is the arithmetic right shift operation where the left most bit namely, A_{n-1} is not only shifted into A_{n-2} but also remains in A_{n-1}. This is to preserve the sign of the number in A and Q. The result of the multiplication will appear in the A and Q.



Design Issues:

Booth's algorithm can be implemented in many ways. This experiment is designed using a controller and a datapath. The operations on the data in the datapath is controlled by the control signal received from the controller. The datapath contains registers to hold multiplier, multiplicand, intermediate results, data processing units like ALU, adder/subtractor etc., counter and other combinational units. Following is the schematic diagram of the Booth's multiplier which multiplies two 4-bit numbers in 2's complement of this experiment. Here the adder/subtractor unit is used as data processing unit. M, Q, A are 4-bit and Q-1 is a 1-bit register. M holds the multiplicand, Q holds the multiplier, A holds the results of adder/subtractor unit. The counter is a down counter which counts the number of operations needed for the multiplication. The data flow in the data path is controlled by the five control signals generated from the controller. these signals are load (to load data in registers), add (to initiate addition operation), sub (to initiate subtraction operation), shift (to initiate arithmetic right shift operation), dc (this is to decrement counter). The controller generates the control signals according to the input received from the datapath. Here the inputs are the least significant Q0 bit of Q register, Q-1 bit and count bit from the down counter.


CONTROLLER
DATAPATH
clk
load
M
4
M
add
4
add/sub unit
clk
sub
Q-1
load
4
Q
4
clk
controller
shift
Q0
A
Q
Q-1
dc
count
clk
counter
1
1