ECE 5720 Modeling and Synthesis of Digital Systems Using Verilog

Worcester Polytechnic Institute

C-Term 2023s

Homework Report 3: Sequential Synthesis

Submitted by

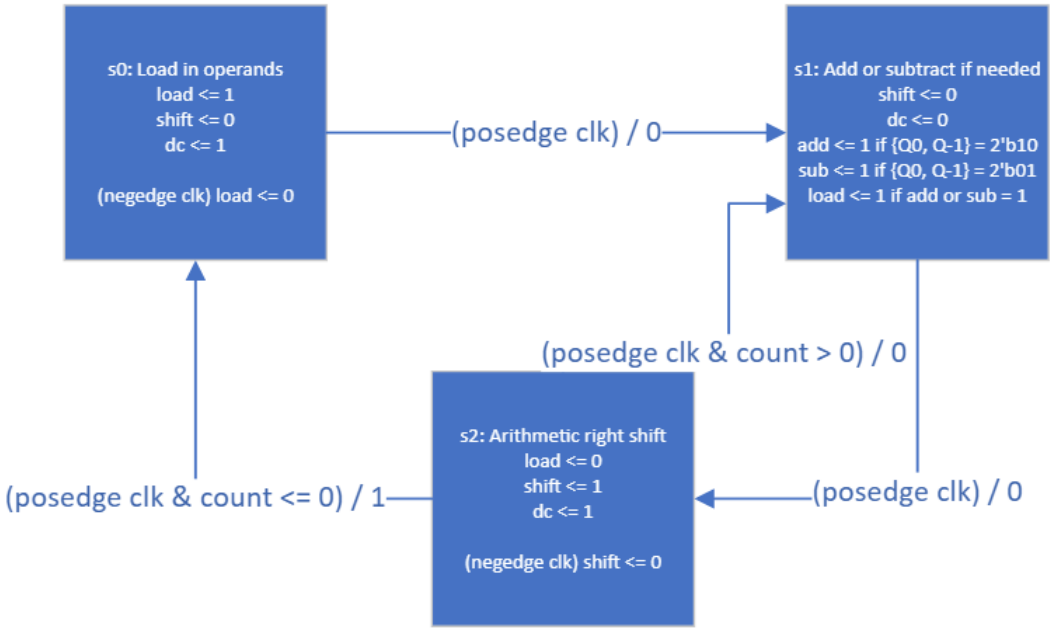
_____

Drew Solomon

_____

Professor Navabi

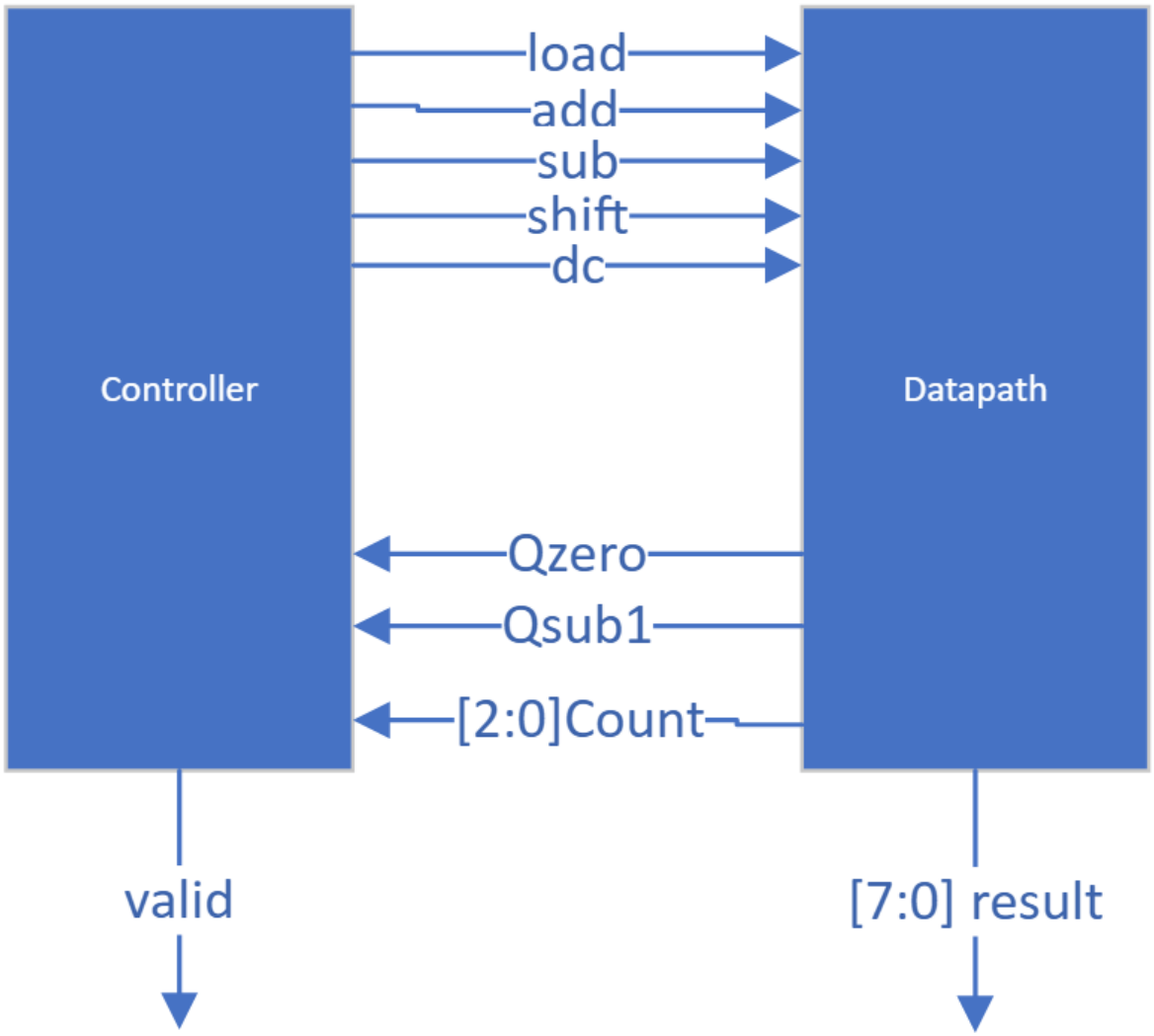
Controller States



The controller operates on 3 states, each representing one of the major steps in the Booth algorithm. The first state is solely about loading in the operands M and Q, and resetting registers to = 0 if necessary to prepare for the next multiplication operation. Upon the next clock tick, the next state is for adding or subtracting depending on the present state of Q[0] and register Q_{-1}. Addition flags occur when Q[0] = 1 and Q_{-1} = 0, and subtraction flags occur when Q[0] = 0 and Q_{-1} = 1. A load flag is also raised to load the output of the add/sub unit to the A register when either of these operations occur, but logic ensures the operand registers won't be changed during this state. State 1 will always lead into state 2 on next clock tick.

State 2 is about the arithmetic shift steps of the algorithm, and has two different potential next states depending on the current state of the counter's count. After shifting the result register right by 1 bit, upon the next clock tick the next state will be state 1 again if count > 0, or state 0 if count <= 0. If state 0 is entered, the algorithm has produced the final outcome of the multiplication and a valid flag is raised signaling the current output of the module is not an intermediate value but the final product. If state 1 is entered again, it means the algorithm is not finished yet and will loop between state 1 and 2 until the count is 0.

Datapath and Controller



The controller and datapath share quite a few control signals to keep synchronized with each other. The controller outputs flags load (for loading registers), add (signals math unit to add operands), sub (signals math unit to subtract operands), shift (shift output registers right 1 bit) and dc (decrement the counter by 1). These flags are sent to the datapath in response to the current state it is in, which is decided partially by the signals the datapath sends to the controller. These signals include the Qzero (Q[0]) register and Qsub1 (Q_{-1}) register, as well as the down counter's current count. The registers are used to determine whether to send an add or sub signal to the datapath, and the count is used to determine which state to enter next while in state 2.

Valid is the flag indicating multiplication is finished and the module output is usable, and result is the module output. Notably, the positive edge is the indicator, not just the signal itself being a 1.

Test Bench Results

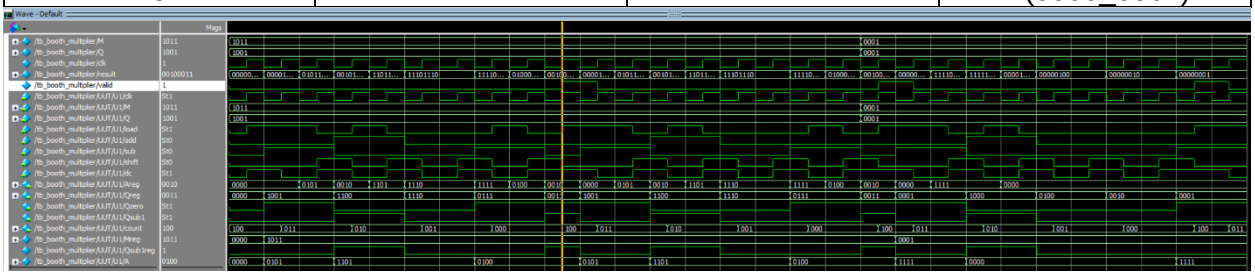
Table 1: multiplication of -5 and -7 using Booth algorithm

Operation	A	Q	Q_{-1}	step
	0000	1001	0	0
A-M	0101	1001	0	1
Arithmetic shift right	0010	1100	1	
A+M	1101	1100	1	2
Arithmetic shift right	1110	1110	0	
Arithmetic shift right	1111	0111	0	3
A-M	0100	0111	0	4
Arithmetic shift right	0010	0011	1	

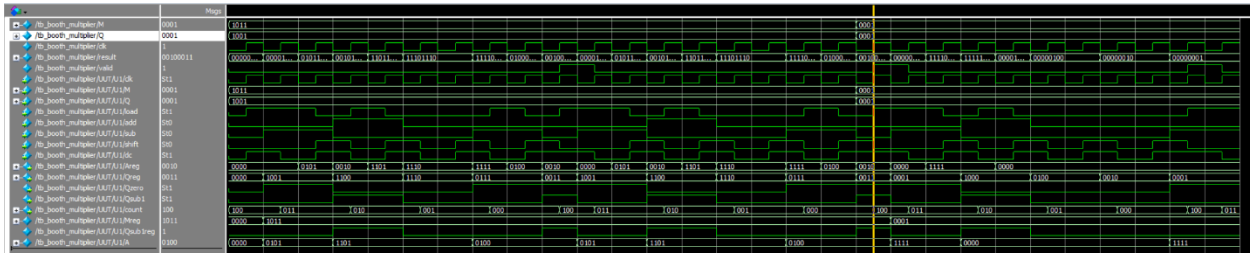
M = -5

For the test bench, the example of -5 * -7 from the assignment instructions was used, with an additional test of 1 * 1 for extra confidence. To ensure that subsequent multiplication operations would work, the -5*-7 operation occurs twice in a row, and the 1*1 operation occurs after. Below is a table of the inputs for each trial and expected output:

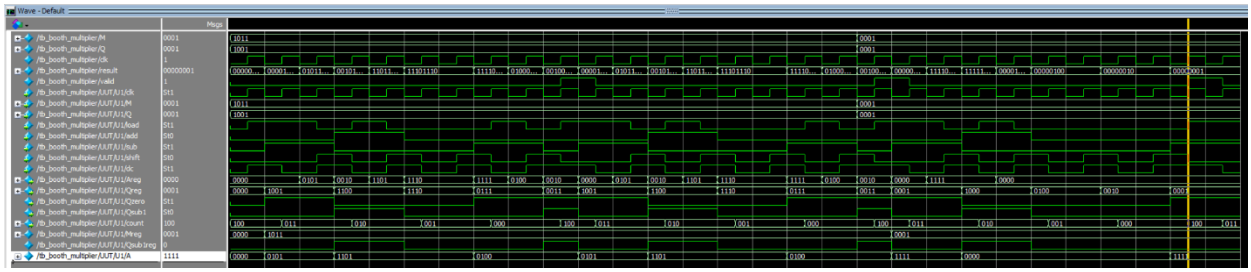
Trial	M	Q	Result
1	-5 (1011)	-7 (1001)	35 (0010_0011)
2	-5 (1011)	-7 (1001)	35 (0010_0011)
3	1	1	1 (0000_0001)



The first trial of -5*-7 produced an outcome of 35, as expected.



The second trial of -5*-7 produced an outcome of 35, as expected.



The third trial using 1*1 produced an outcome of 1, as expected.

Quartus Synthesis Report

Compilation Report - booth_multiplier

Flow Summary

<<Filter>>

Flow Status	Successful - Tue Feb 28 16:00:18 2023
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Lite Edition
Revision Name	booth_multiplier
Top-level Entity Name	booth_multiplier
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	14 / 56,480 (< 1 %)
Total registers	33
Total pins	18 / 268 (7 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)