CSE 331 FALL 2020

HOMEWORK 3

ABDULLAH ÇELİK

171044002

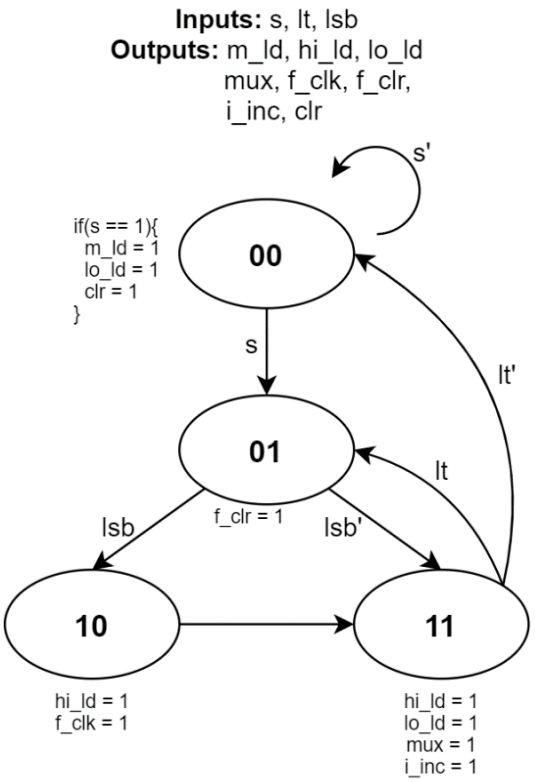
Improvements

1. I designed 32 bit adder and 1 bit right shifter.
2. In order to reduce the total number of states to 4 in the state diagram, I connected the outputs of the control unit in the 0th state to the condition that the s signal is 1. The reason for this is that when the s signal is 0 at the 0th state, the control unit gives outputs, which causes the result to be reset. I designed like this to avoid adding extra state to solve this problem. In this way, the control unit became 4 states and 2 bit registers worked. Otherwise I would have to use 3 bit registers. This would require me to consider one more bit in the outputs of the control unit. Some of the 2 input gates would go up to 3 inputs. Control unit register would be 3 bits.
3. A comparison had to be made for the loop to work 32 times. Instead of doing this with the comparator, I used the counter's overflow bit. In this way, I made the loop to rotate 32 times.

State Diagram

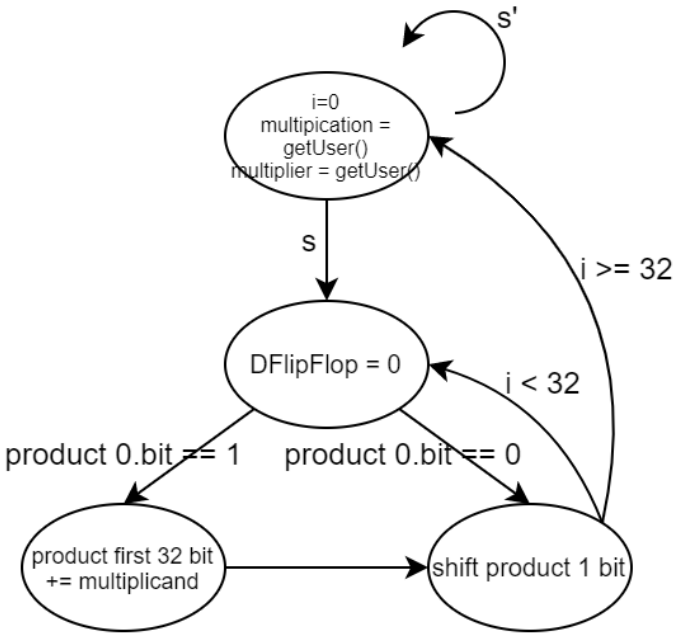
To simplify the diagram below, outputs that it's 0 are not shown in the diagram. And also, s input is expected to be 1 in order for the inputs to be 1 in the 00 state(The reason is mentioned in the optimization section).

- s (start)
- lt (i less than 32)
- lsb (less significant bit of product)
- m_ld (load multiplicand)
- hi_ld (load product first 32 bit)
- lo_ld (load product last 32 bit)
- mux (multiplexor select bit)
- f_clk (enable d flip flop)
- f_clr (clear d flip flop)
- i_inc (increment i)
- clr (clear)



Draw Datapath

Inputs: s, i, multiplicand, multiplier, product 0.bit
Outputs: product(64 bit)



Draw Truth Table

Inputs					Outputs								
s1	s0	s	lsb	lt	n1	n0	m_ld	hi_ld	lo_ld	f_clk	f_clr	i_inc	clr
0	0	0	X	X	0	0	0	0	0	0	0	0	0
0	0	1	X	X	0	1	1	0	1	0	0	0	1
0	1	X	0	X	1	1	0	0	0	0	1	0	0
0	1	X	1	X	1	0	0	0	0	0	1	0	0
1	0	X	X	X	1	1	0	1	0	1	0	0	0
1	1	X	X	0	0	0	0	1	1	0	0	1	0
1	1	X	X	1	0	1	0	1	1	0	0	1	0

Derive Boolean Expressions From the Truth Table

- n1 = s1’s0lsb’ + s1’s0lsb + s1s0’
  = s1 XOR s0
- n0 = s1’s0’s + s1’s0lsb’ + s1s0’ + s1s0lt
  = s0’s + s1’s0lsb’ + s1s0’ + s1lt
- m_ld = s1’s0’s
- hi_ld = s1s0’ + s1s0lt’ + s1s0lt
  = s1s0’ + s1s0
  = s1
- lo_ld = s1’s0’s + s1s0lt’ + s1s0lt
  = s1’s0’s + s1s0
- f_clk = s1s0’
- f_clr = s1’s0lsb’ + s1’s0lsb
  = s1’s0

- i_inc = s1s0lt' + s1s0lt
  = s1s0
- clr = s1's0's

Test Cases

Test ID	Multiplicand	Multiplier	Expected Product	Actual Product	Pass/Fail
1	1111 1111 1111 1111 1111 1111 1111 1111	1111 1111 1111 1111 1111 1111 1111 1111	1111 1111 1111 1111 1111 1111 1111 1110 0000 0000 0000 0000 0000 0000 0000 0001	1111 1111 1111 1111 1111 1111 1111 1110 0000 0000 0000 0000 0000 0000 0000 0001	Pass
2	0111 0000 1011 0000 1100 0010 0000 0000	0011 1110 1010 0010 1001 1111 1111 1011	0001 1011 1001 0010 0110 0101 0100 0011 0000 1001 1100 1100 0011 0110 0000 0000	0001 1011 1001 0010 0110 0101 0100 0011 0000 1001 1100 1100 0011 0110 0000 0000	Pass
3	0000 1101 0101 1111 1110 0010 1010 1010	1111 0101 1101 1101 1100 0101 1011 1000	0000 1100 1101 1000 0101 1010 0000 0111 1100 0101 0001 1001 1011 1100 0011 0000	0000 1100 1101 1000 0101 1010 0000 0111 1100 0101 0001 1001 1011 1100 0011 0000	Pass
4	1111 1110 1101 1100 1011 1010 1001 1000	1101 1010 1011 1100 0000 0011 0000 0010	1101 1001 1100 0011 0010 0011 1111 1101 0000 1011 1000 1001 0011 1101 0011 0000	1101 1001 1100 0011 0010 0011 1111 1101 0000 1011 1000 1001 0011 1101 0011 0000	Pass
5	1111 1110 1101 1100 1011 1010 1001 1000	0000 0000 0000 0000 0000 0000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	Pass
6	0000 0000 0000 0000 0000 0000 0000 0000	1111 1110 1101 1100 1011 1010 1001 1000	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	Pass