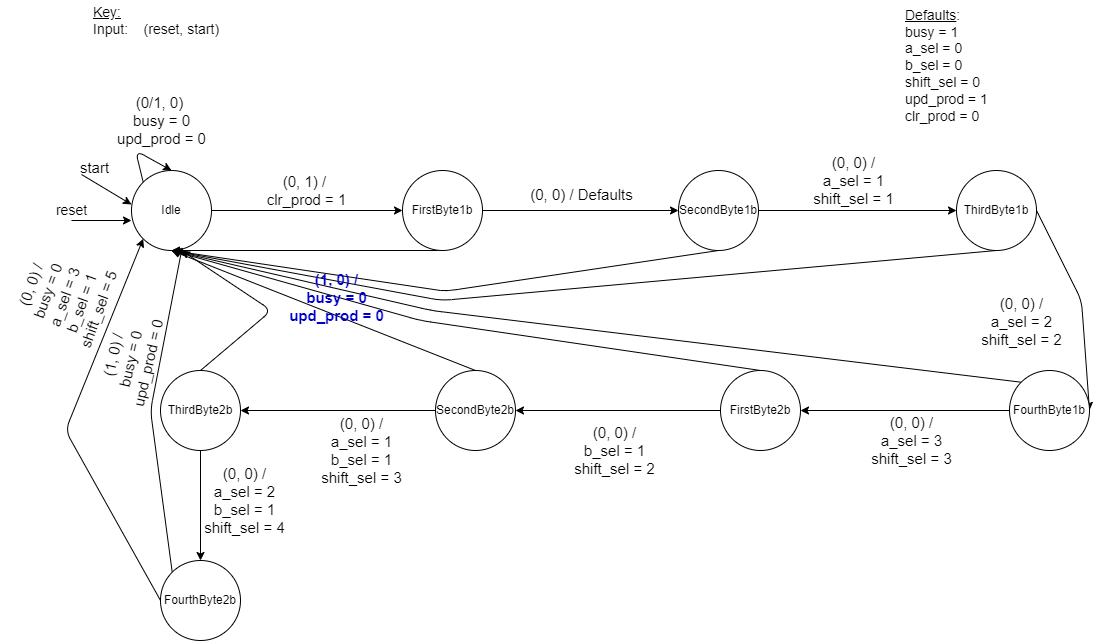
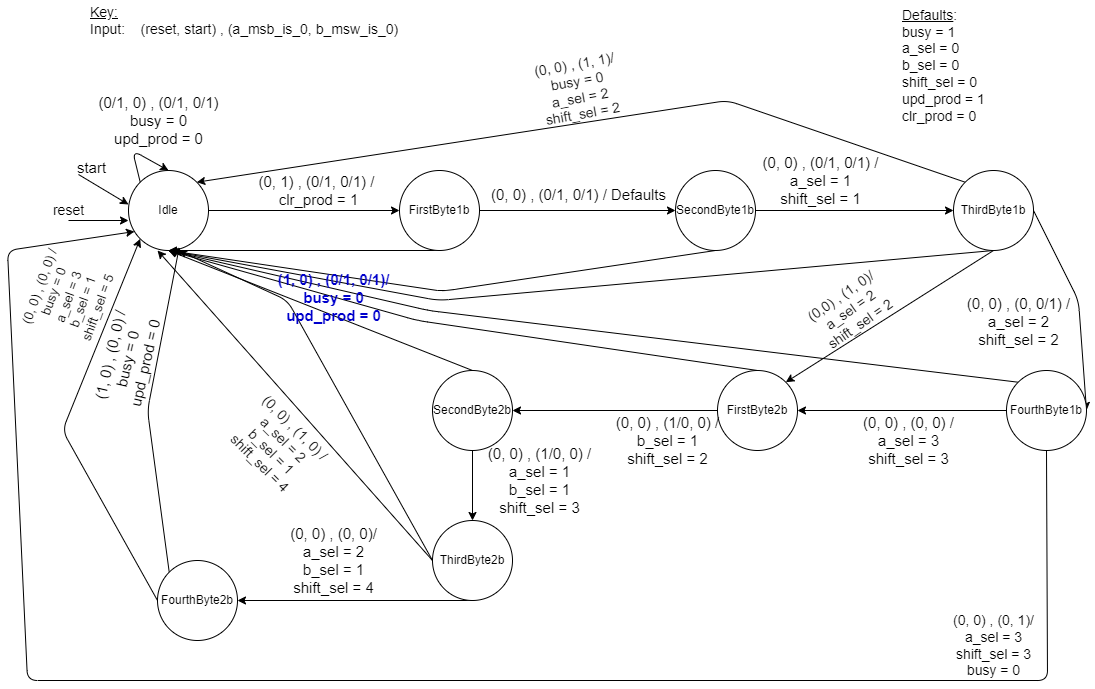
|  |  |
| --- | --- |
| 341312304 | שרה גריפית |
| 207223066 | טל שמיר |

2.1: Diag

The following is a diagram of the FSM we will implement:

Each multiplication takes 9 clock cycles.

2.2: 

The fastest calculation of the multiple will be given when both a\_msb\_is\_0 and b\_msw\_is\_0 are 1. In this case, the calculations after the state ThirdByte1b are irrelevant, and so the calculations will be completed after 4 clock cycles.

If a\_msb\_is\_0 is 1 while b\_msw\_is\_0 is 0, the calculation will continue to the state FirstByte2b after ThirdByte1b, skipping the FourthByte1b. Likewise, the calculation will skip over the state FourthByte2b and instead return to Idle after ThirdByte2b. The calculation in this case will take 7 clock cycles.

If a\_msb\_is\_0 is 0 while b\_msw\_is\_0 is 1, the calculation will continue to Idle after the state FourthByte1b. The calculation in this case will take 5 clock cycles.

In every other instance (when both a\_msb\_is\_0 and b\_msw\_is\_0 are 0), the calculation time will remain the same as it was in the previous question, 9 clock cycles.

3.4: Simulation of Multiplier 32x32

The following are the wave lines of the simulation and test bench created:

A picture containing chart

Description automatically generated

A picture containing chart

Description automatically generated

The clock starts at the value of 0 and changes its value every 10 units of time. Once the clock changes to 1 for the first time, the values of product, start, and reset are initialized: product and start are equal to 0, and reset is equal to 1 as requested. This lasts for 4 cycles of the clock.

The values of a and b are initialized after these 4 cycles, and thereafter, start is changed to 1 for one cycle, starting the multiplication process. One cycle after start changes, the value of busy is changed to 1, in accordance with the requirements of the exercise. As seen in the diagrams, with each clock cycle, the next\_state and current\_state change in the order described in the diagram from question 2.1. Finally, after 9 clock cycles, busy goes down to the value of 0, the current and next states return to Idle, and the product from the multiplication of our ID numbers is received. As shown, the result is: 341312304 x 207223066 = 70727782098404064