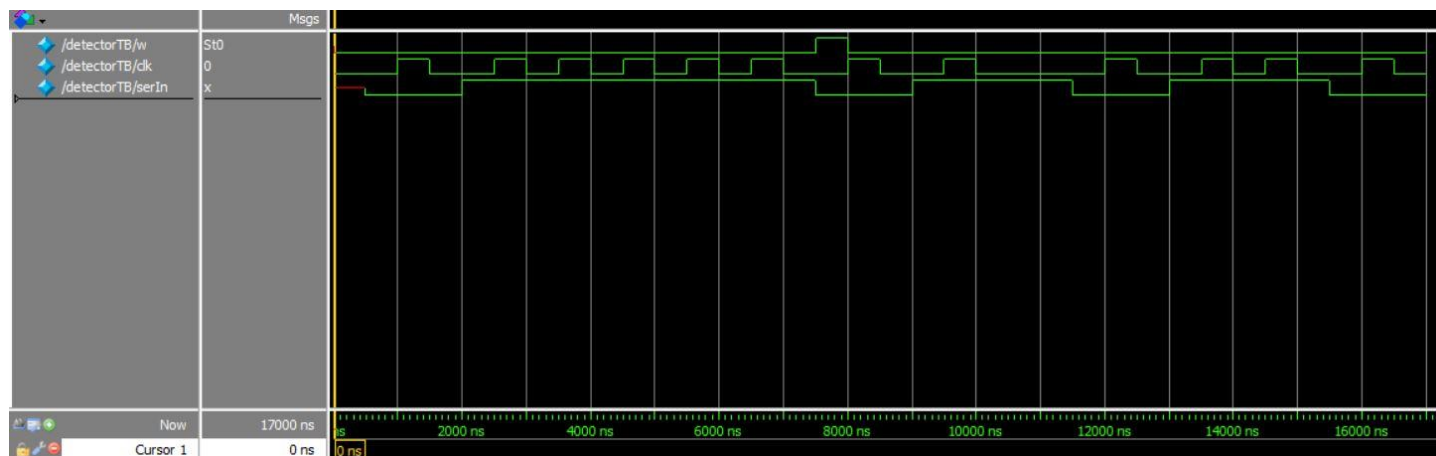
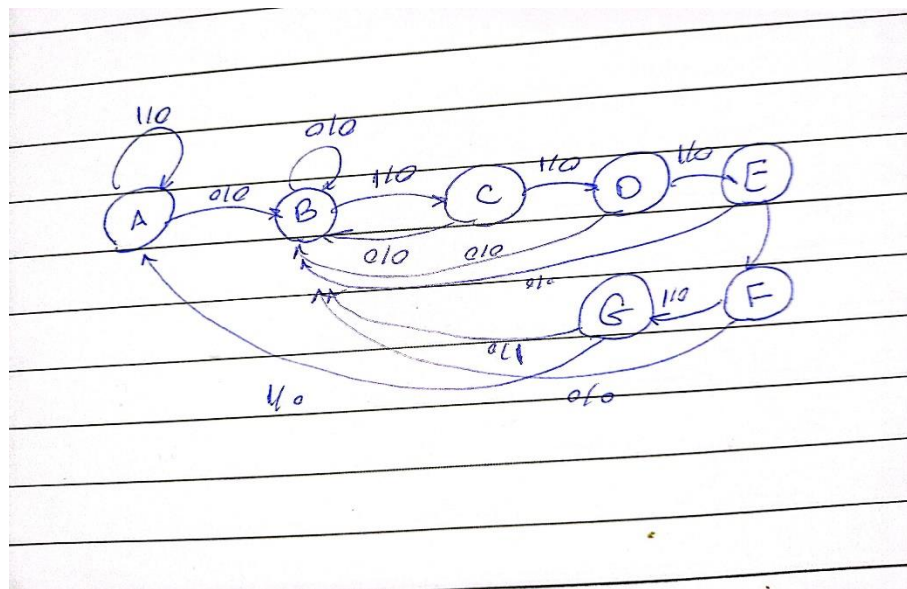
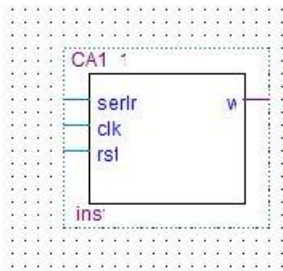


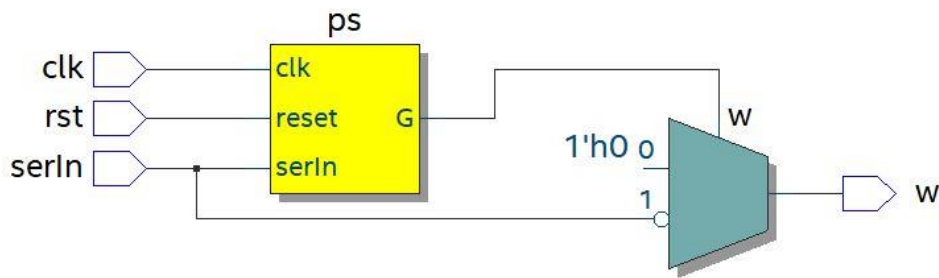
Here we are going to implement a circuit that has 3 main parts. First part is a sequence detector that can be implemented by state machine which mealy is chosen here. Second part is nt collector that can be implemented with a shift register and counter. The last part that is transmitter can be implemented with modulo nt counter.



After compiling with quartus we can create a symbol for this circuit which is:



Also there are some reports that can be used too. Like floor plan, that shows the total structure of the circuit which is:

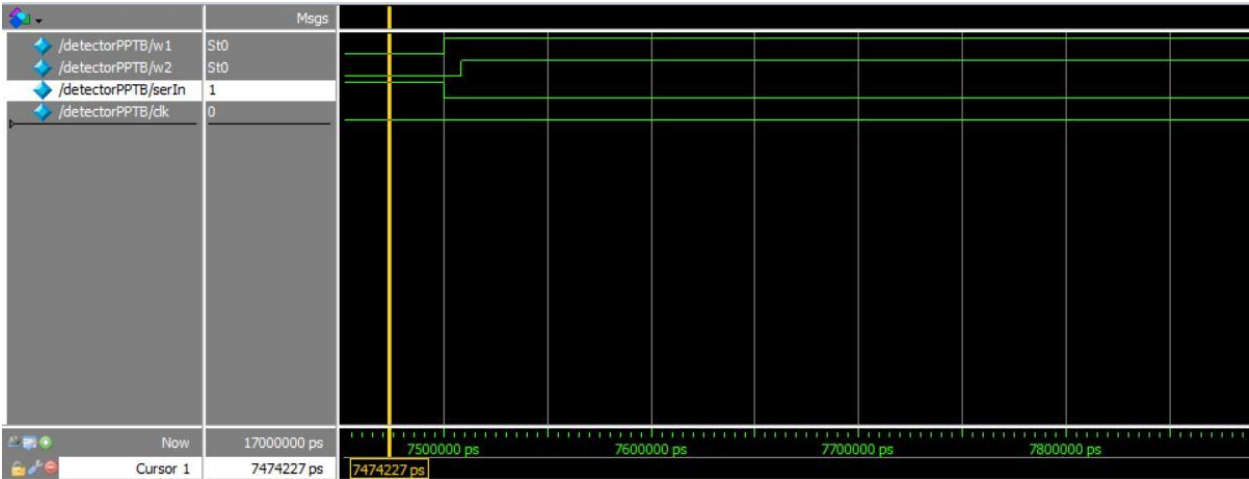


The report of compilation shows how many things are used which is:

Flow Status	Successful - Sat Dec 30 22:55:19 2023		
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition		
Revision Name	CA1_1		
Top-level Entity Name	CA1_1		
Family	Cyclone IV GX		
Device	EP4CGX15BF14A7		
Timing Models	Final		
Total logic elements	7 / 14,400 ( < 1 % )		
Total registers	6		
Total pins	4 / 81 ( 5 % )		
Total virtual pins	0		
Total memory bits	0 / 552,960 ( 0 % )		
Embedded Multiplier 9-bit elements	0		
Total GXB Receiver Channel PCS	0 / 2 ( 0 % )		
Total GXB Receiver Channel PMA	0 / 2 ( 0 % )		
Total GXB Transmitter Channel PCS	0 / 2 ( 0 % )		
Total GXB Transmitter Channel PMA	0 / 2 ( 0 % )		
Total PLLs	0 / 3 ( 0 % )		

	Statistic	Value
1	Total registers	6
2	Number of registers using Synchronous Clear	0
3	Number of registers using Synchronous Load	0
4	Number of registers using Asynchronous Clear	6
5	Number of registers using Asynchronous Load	0
6	Number of registers using Clock Enable	0
7	Number of registers using Preset	0

Quartus output has delay and timing so we claim that the output of quartus will change with delay in comparison with the pre synthesis which can be understood from wave form:



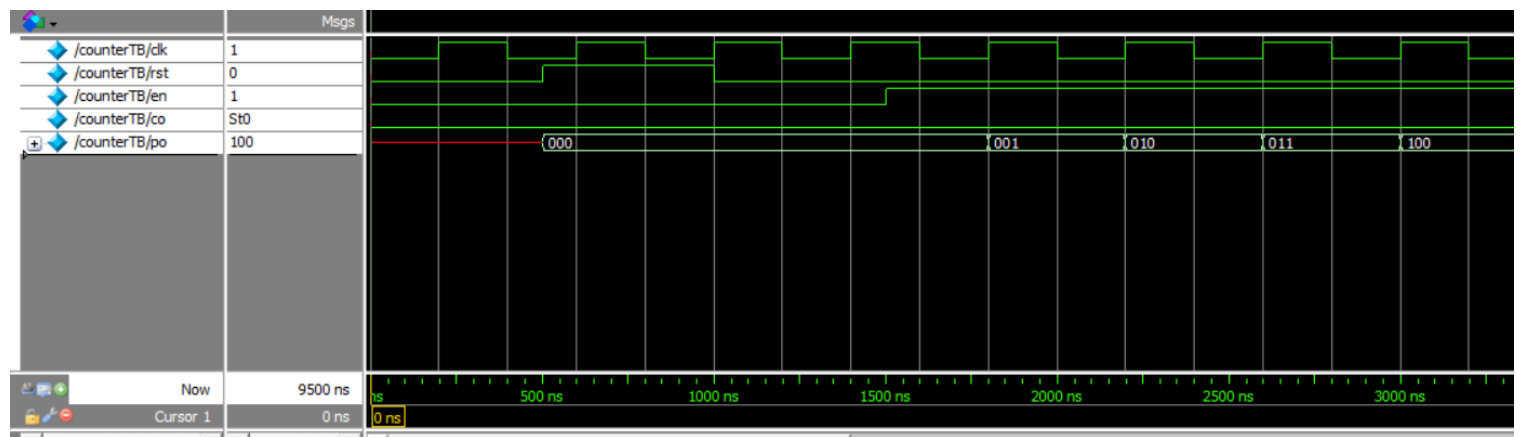
This is a part of waveform that shows the delay of post synthesis part.

Part b)

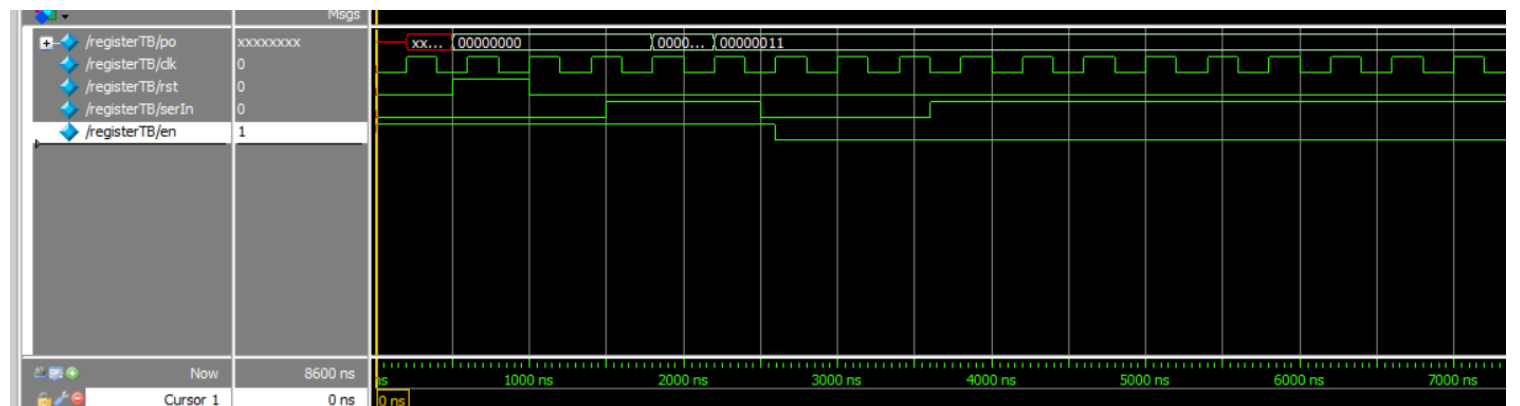
Here we are going to implement nt collector. We need a shift register because serin is just one bit but we want to collect a 8-bit number. Also we need a 3-bit counter that controls shifting. Actually when carry out is one we stop shifting and the outvalid signal will be one as the number is ready.

Here shift register and counter are described in different code files and will be used in another file for combining.

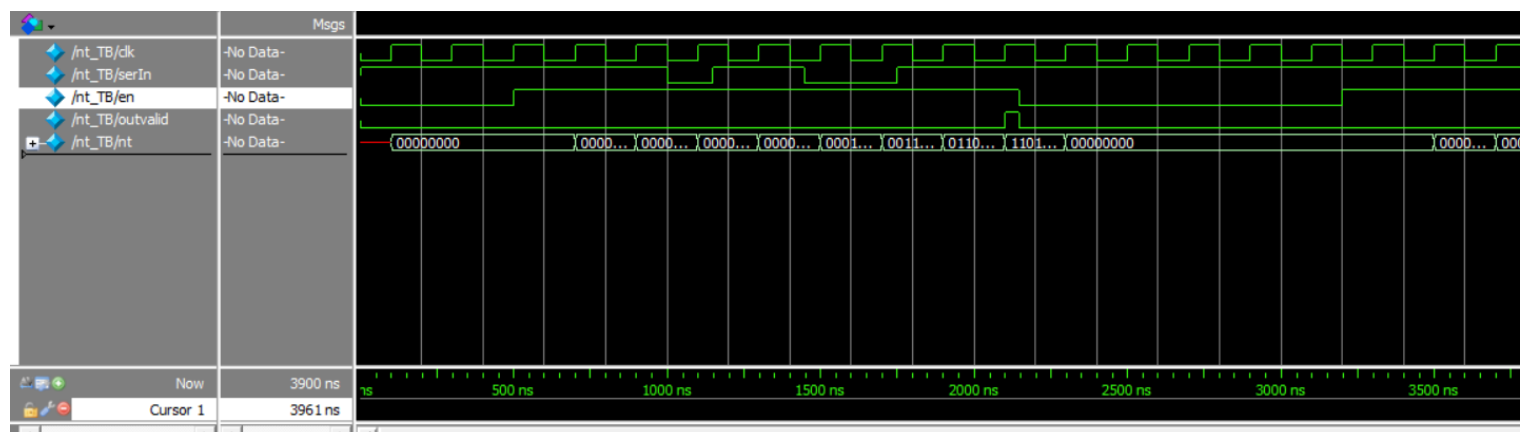
Both parts are tested.



This wave form is related to testing the 3-bit counter.

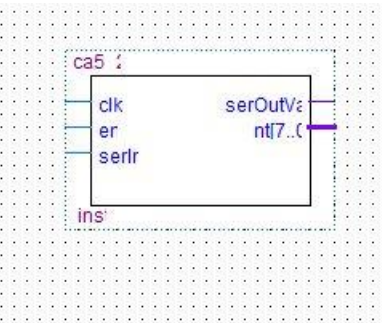


This is wave form of testing shift register that shown on posedge of clock serin will be in this register and also shift to left.

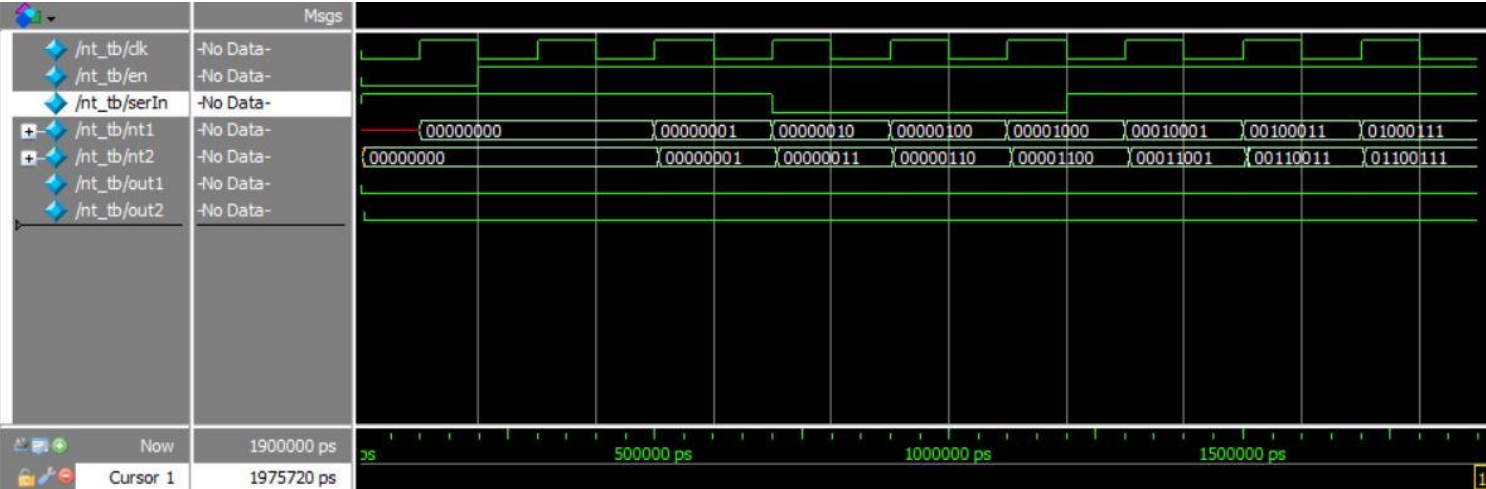


This wave form is for testing nt collector. As it's shown here after 8 clock cycle outvalid is ready and becomes one after that because the circuit is not enable outputs don't change.

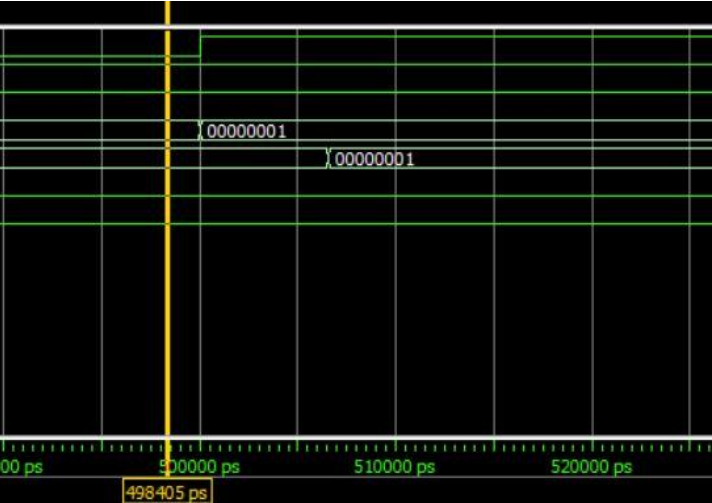
After using quartus the symbol will be:



As we know quartus output has delay so based on the wave form it's true:



This is the total test bench but in parts of changing:

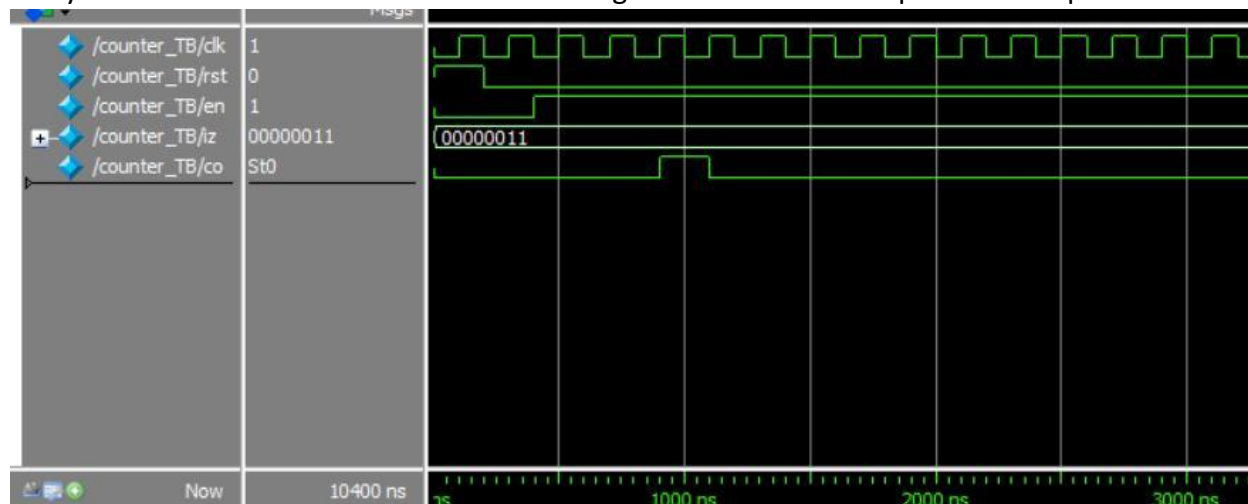


There will be delay.

### Part c)

Now we are going to implement the transmitter that has a counter. This counter will work differently based on the nt that we collected from last part. So, initializing this circuit won't be constant so we use a subtractor for calculating the iz amount.

Carry out of this counter will be signal for transfer part that puts serin on the output.

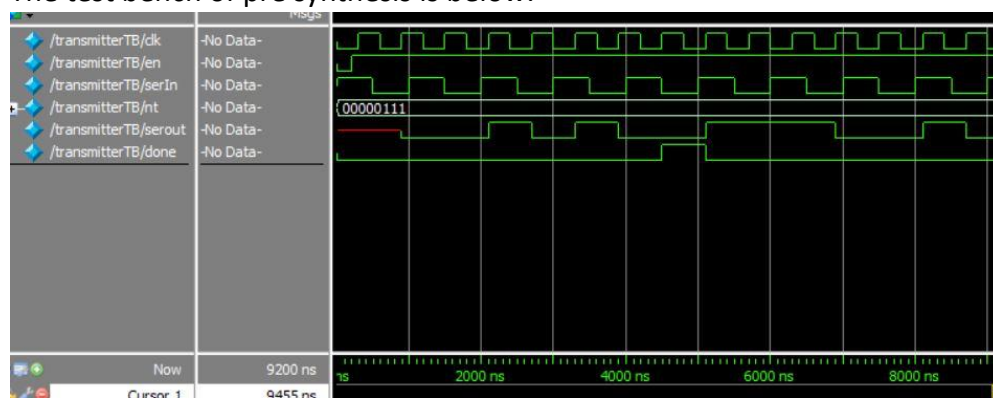


This wave form is related to counter with nt = 00000011 that makes the carry out 1 after 4 clock cycle.

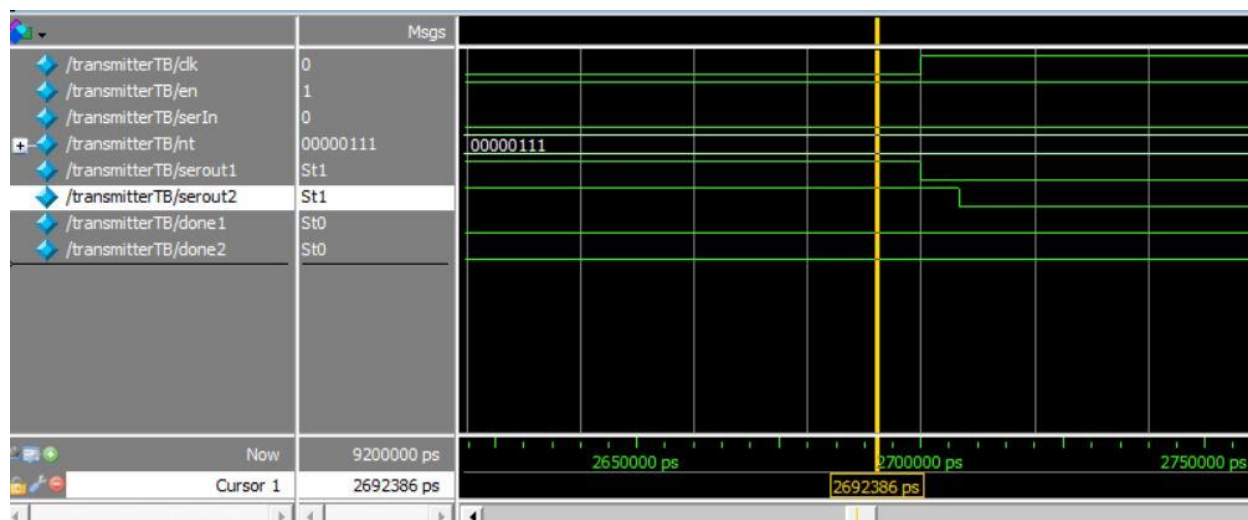
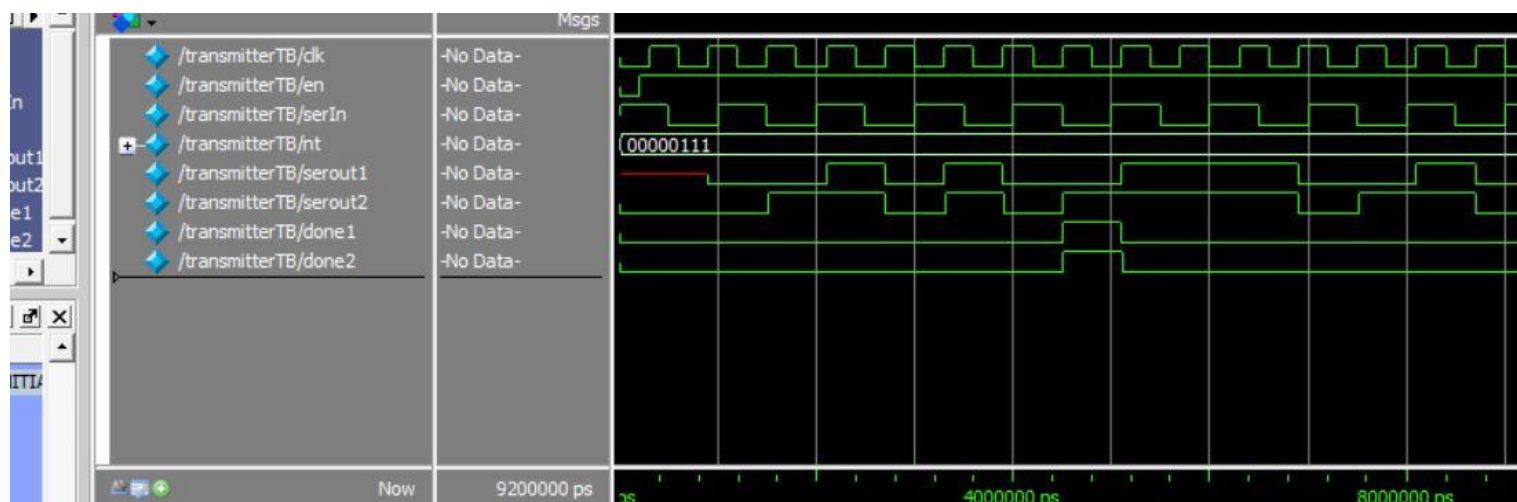


This wave form is related to transfer part that with condition of enable, on the posedge of clock serin will be on output.

The test bench of pre synthesis is below:



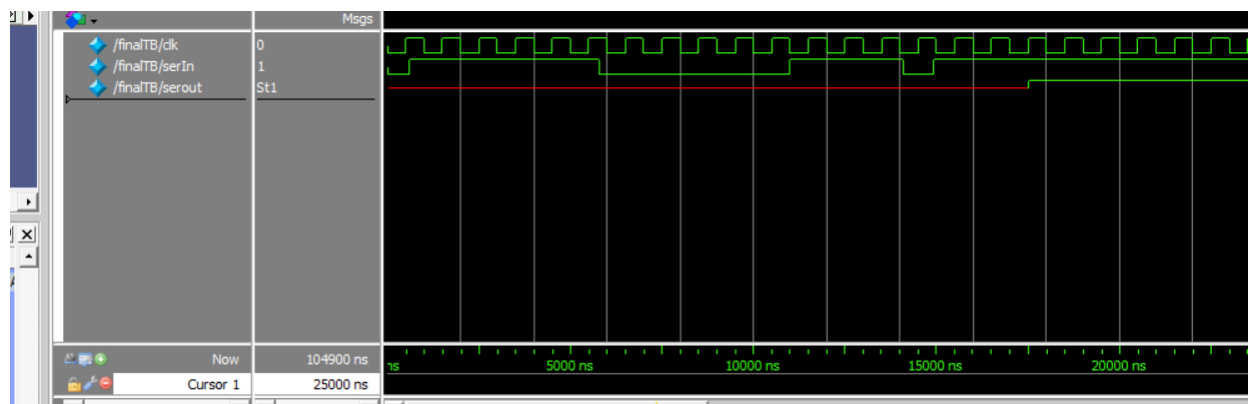
Post and pre synthesis:



Part d)

Now we are going to combine all parts together.

Output of sequence detector will be signal for nt collector and nt collector outputs will be either signal or input data for transmitter.



Here there is a pre synthesis that shows after finding sequence and collecting nt serin will be on serout.