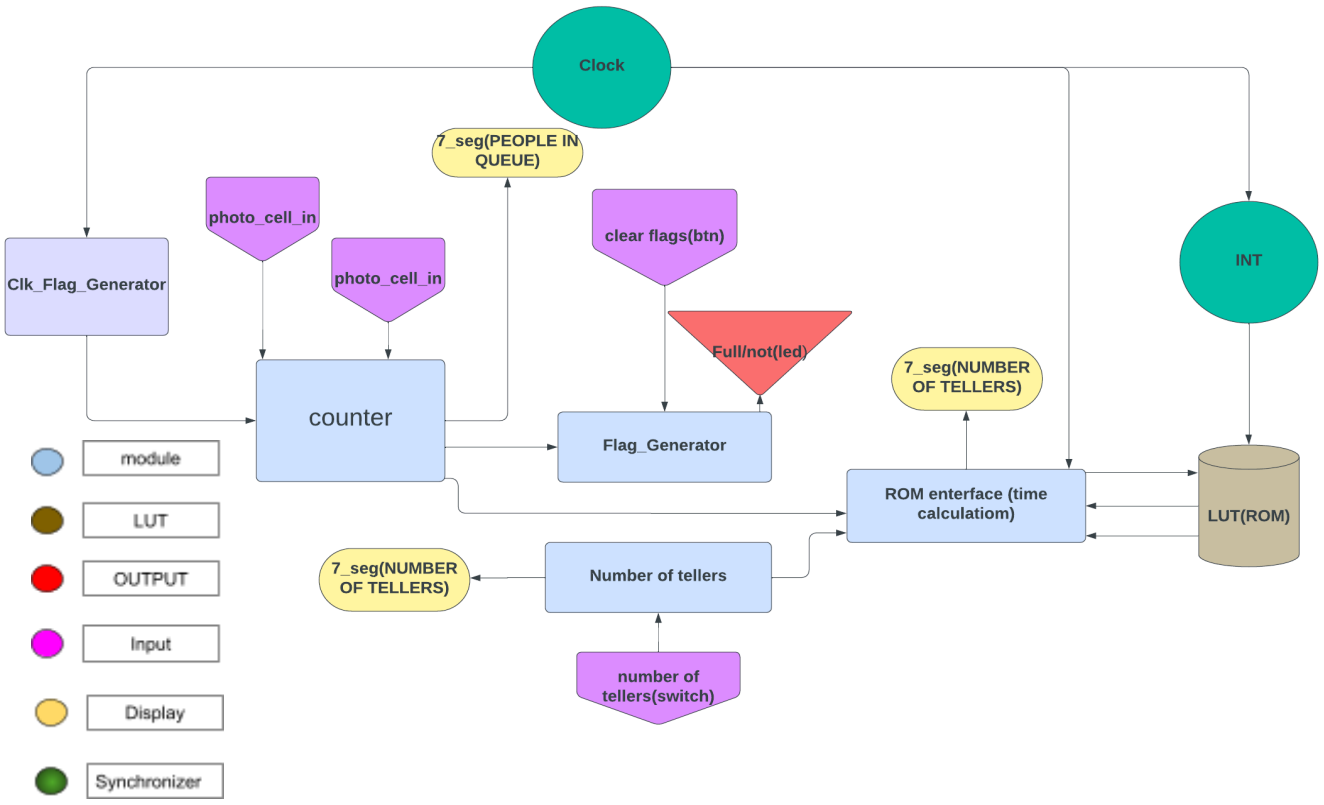
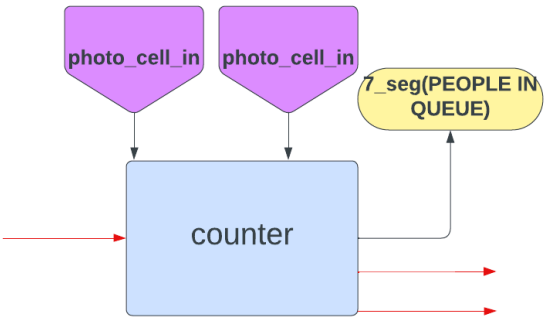
ABQM Bank System DOCUMENTATION

System Architecture and Intuition:

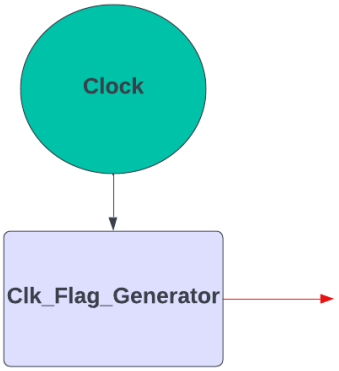
The following figure demonstrates the general design of the system architecture, highlighting the modules used, their inputs/outputs, and how they interface with each other. in this section, each module will be explained in depth.



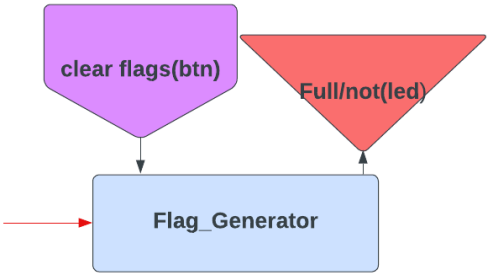
First to be discussed is the “counter” module which as the name implies its role is to count the number of customers in the queue, it receives two inputs which are the photocells placed at the end and front of the queue, BPH and FPH respectively, a clock flag and a reset flag. where if the BPH is high the counter increments while on the other hand if FPH is high the counter decrements essentially working as an up-down counter with an option to reset the counter to zero if the reset flag is triggered.



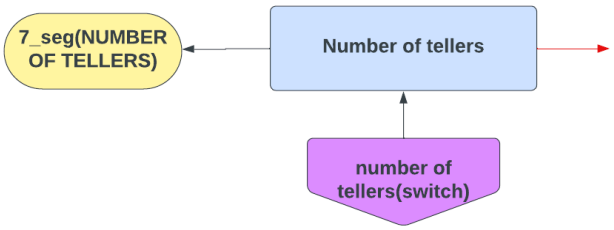
Concerning the requirement to raise the alarm when the queue is empty and the FPH detects a movement the up-down counter approach was modified to contain two counters both counters keep the number of customers in the queue but only one can be reset to zero while the other is only initialized with zero when at the first time the system is run while it also decrements and increments, the reset button does not affect the counter_check, thus if both counters are not equal this means that the system was reset when there were customers in the queue counter1 will 0 while counter_check will be holding the correct number of customers when both are compared and there difference is not zero the alarm will be raised.



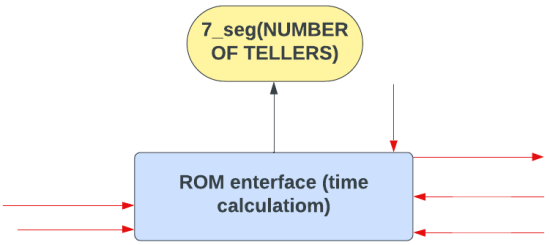
The clock flag is the output of the “Clk_Flag_Generator” and receives the FPGA’s clock as an input the sole purpose of this module is to raise a flag when the system is first up to set the counter_check as zero to be more specific the flag is sent at the seventh cycle of the FPGA clock, the flag is an active low for power consumption reason. This ensures that the counter_check is only reset when the whole system is initiated.



The “Flag_Generator” module only raises alarms when it receives the flag from the counter module if the queue is full or empty or when the difference between counter and counte_check is not zero. The only way to lower the alarms is by resetting (clear flags) from an authorized person.



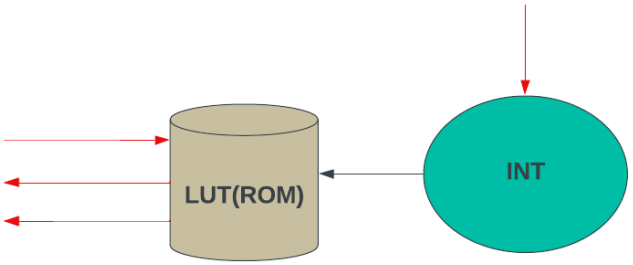
In the “Numbers of tellers” module decoding from the teller switches takes place.the module consisstes of a lookup table containing all possible combination for available teller and returning the actual binary represrntaion (number of active tellers) ,for example 011 maps to binary 10 and 111 maps for binary 11.
The input for this module is only the switches from the teller’s end, while its outputs are a 2 bit bus heading to the “ROM_Interface” module and and another 4 bit bus heading to the BCD module as a display for this stage.



Connecting the pipeline to LUT (“ROM”), ROM_Interface module was designed for interfacing job with the LUT it receives a 4 bit bus containing the current number of customers in the queue from the “counter” module and receives a 2 bit bus from the “number of tellers” module and a cock signal and an interupt signal sent from the LUT(“ROM”) which only sent when there is data in the register connecting both ROM_interface module and the LUT(“ROM”) ,and the final output is the wtime to be displayed sent from the LUT(“ROM”) as requested, the most significant bit of the 4 it bus is ignored ,the both buses are concatinated to form a 5 bit address. Then it is sent to the LUT(“ROM”).

Finally the ROM module which is a lookup tabel (LUT) containing addresses and a saved precalculated waiting time.the ROM outputs the time

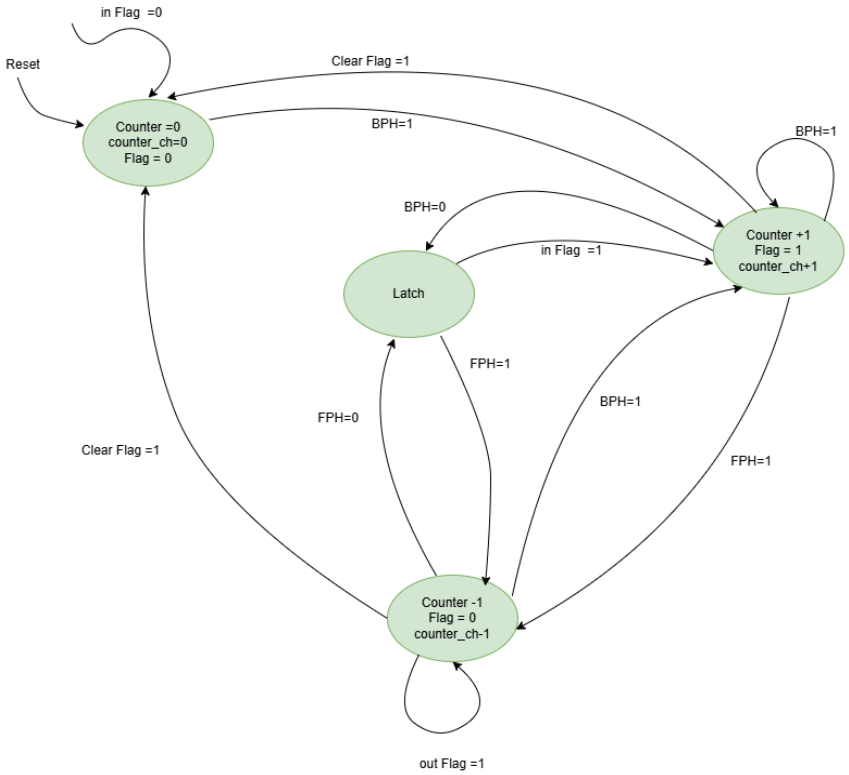
associated with the received address and an interrupt signal to flag for the memory interface that new data was sent. The “int” module creates the interrupt signal and pases it to the “ROM” module.



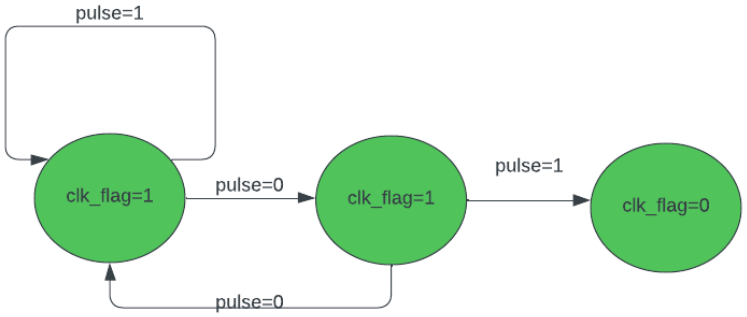
FSM Deep Dive:

To be noted modules not presented in this section are implemented as LUTS.

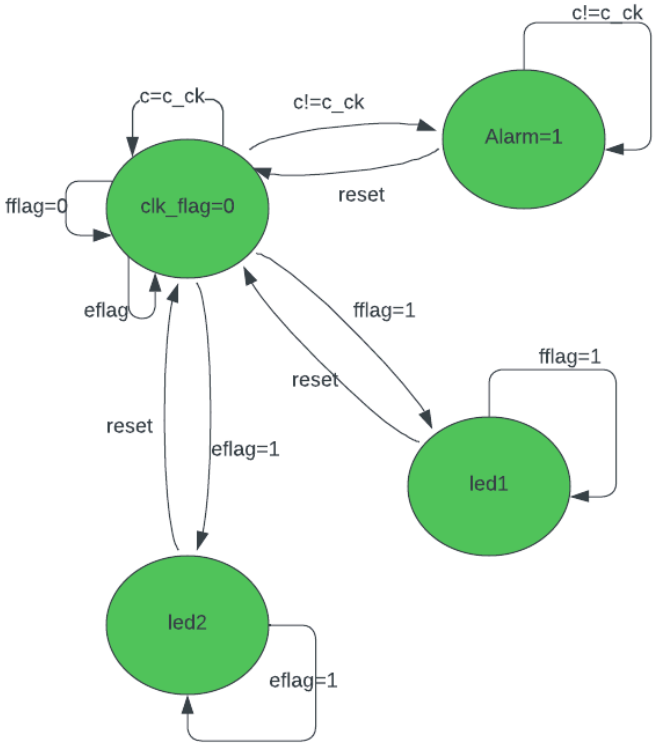
-” Counter module”.FSM



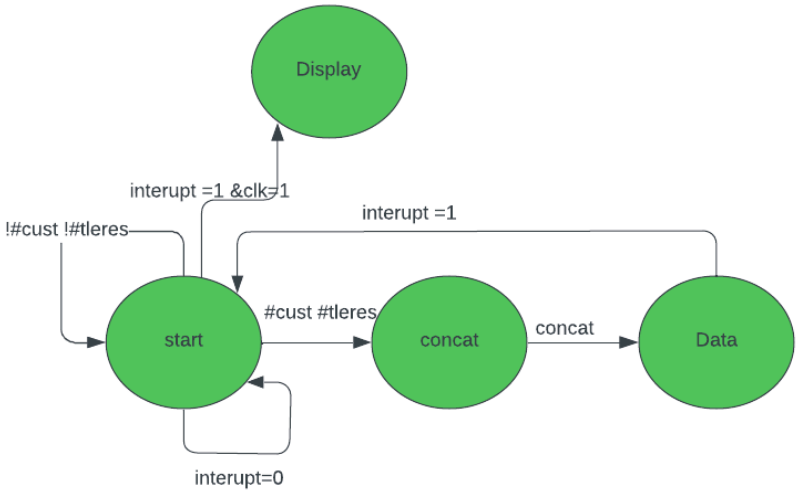
-’”Clk_flag_Generator module”.FSM



“Flag_Generator” module FSM

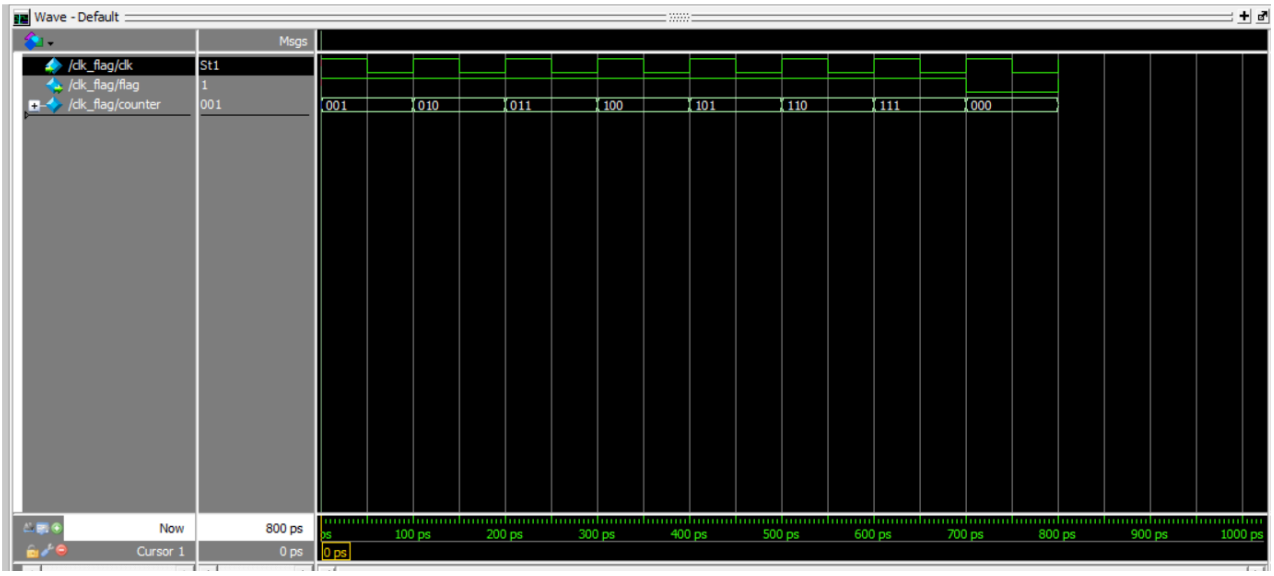


“ROM_interface module”.FSM

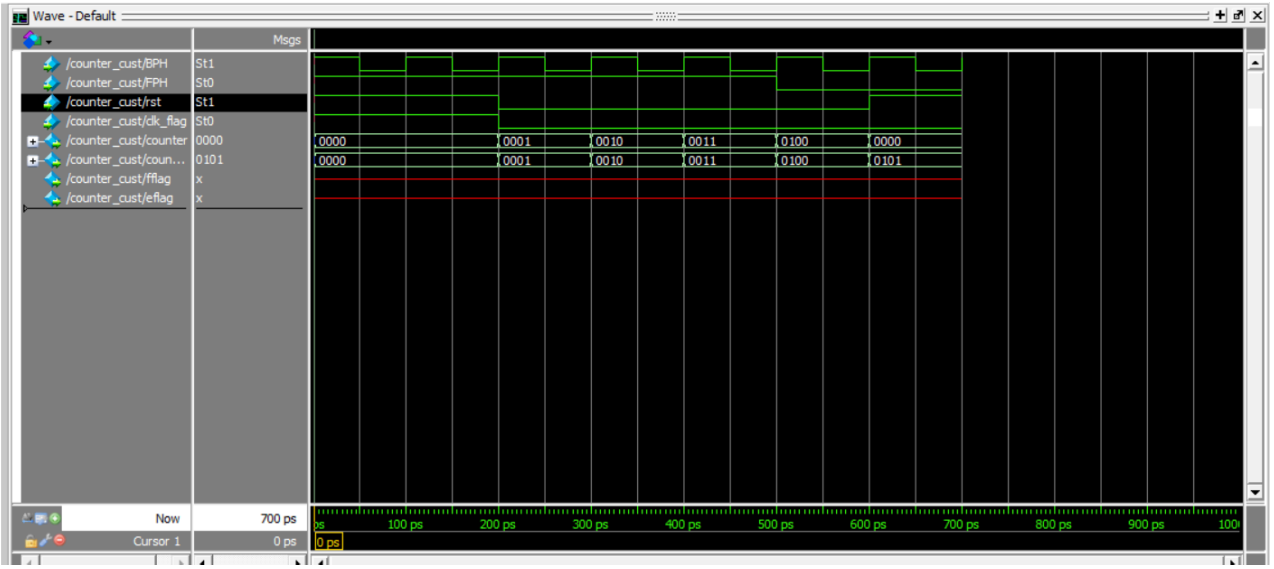


Simulations:

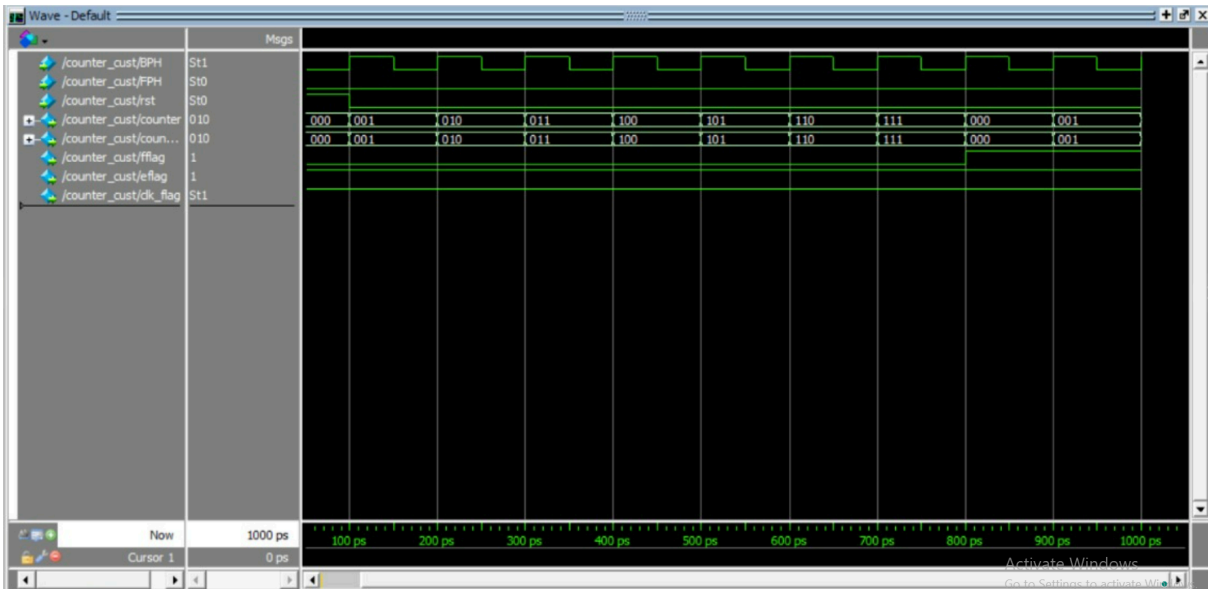
This simulation illustrates the active low clock flag when initializing the system.
-’”Clk_flag_Generator module”.



The below simulation highlights that only one counter is affected by the reset button enabling us to compare the two counters detect any anomalies and raise the alarm -” Counter module”.

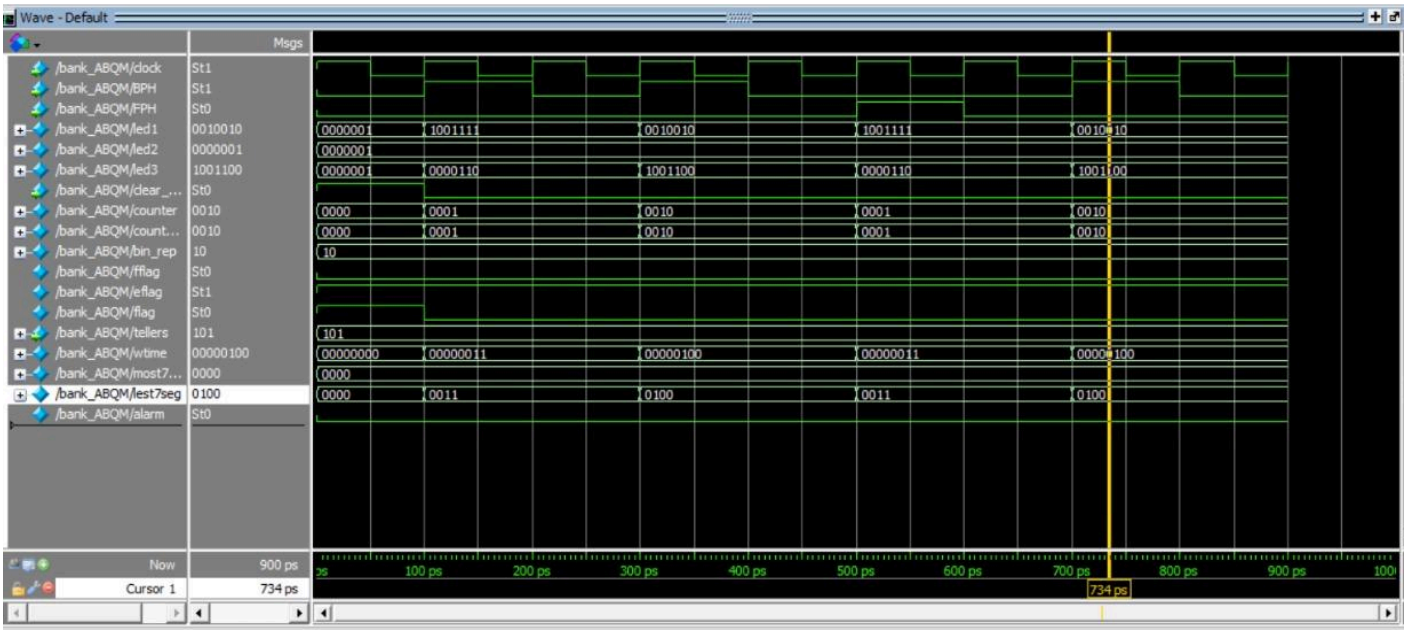


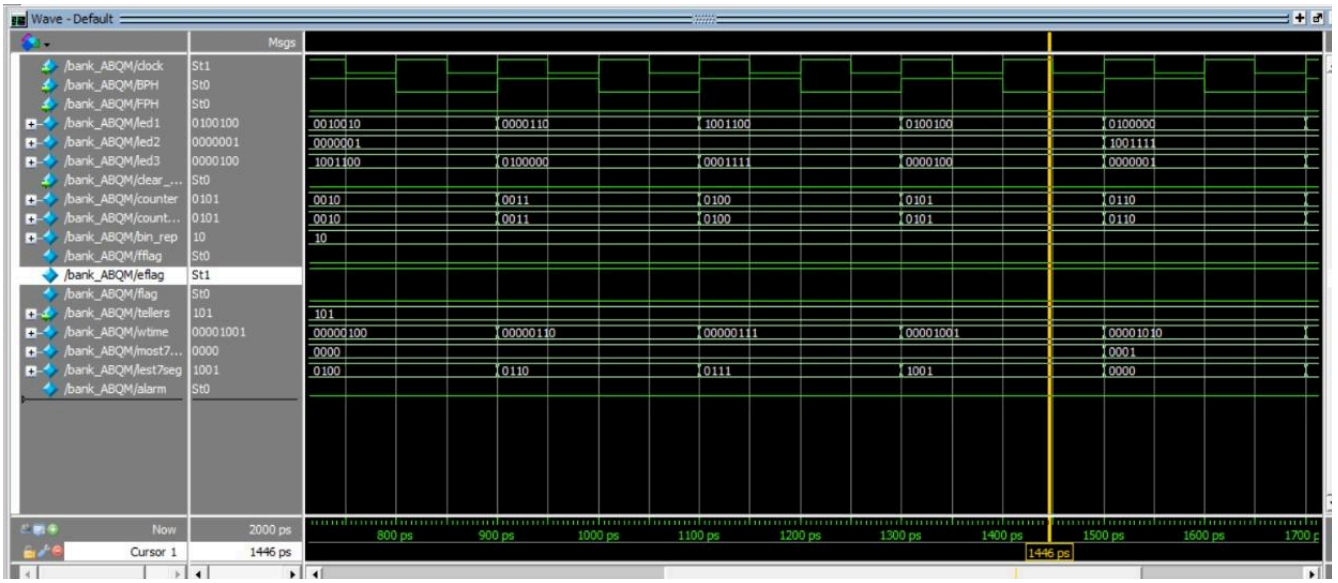
The following simulation demonstrates the counter running in normal condition Both counter and counter_check increment and reset.

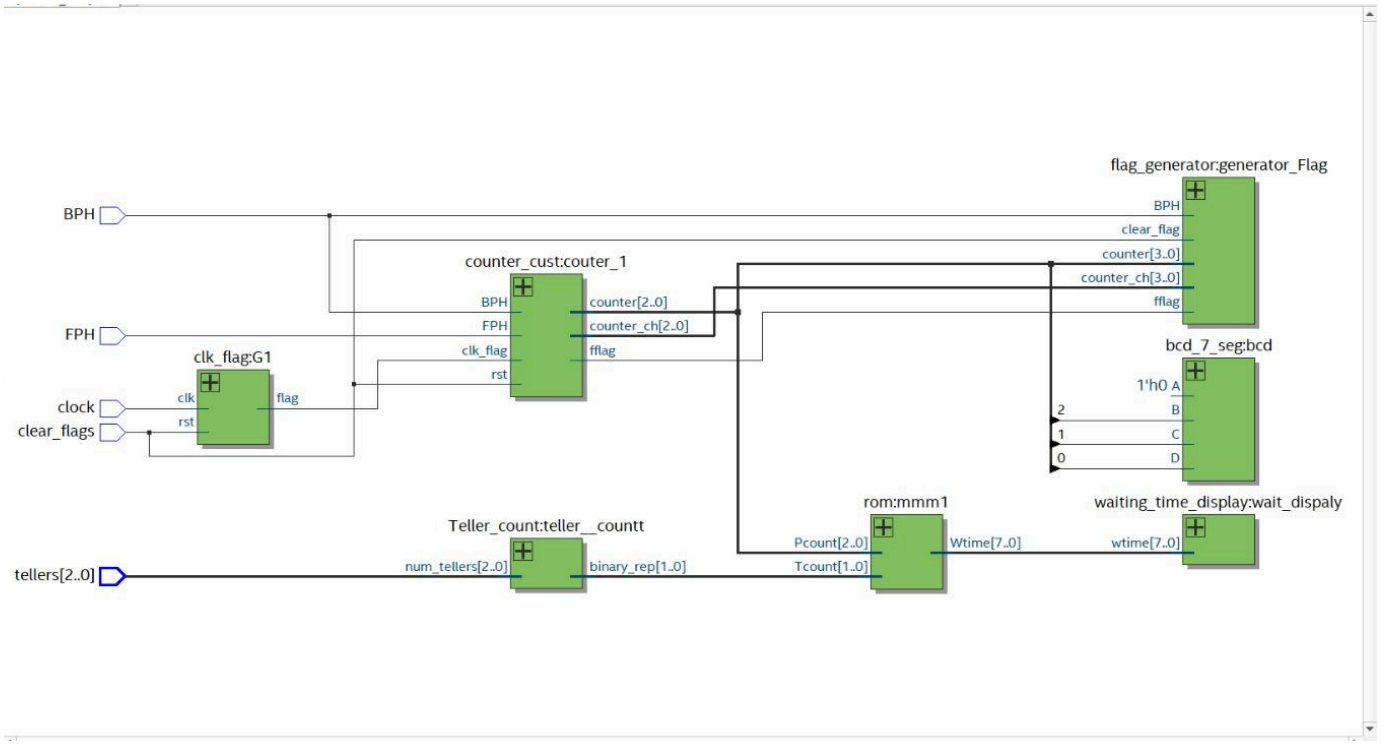


The simulation below demonstrates that the LUT “ROM” receives the required address and returns the saved waiting time.
-’’LUT(ROM)”.

Final full simulation results:




Wave - Default
Msgs
/bank_ABQM/clock
/bank_ABQM/BPH
/bank_ABQM/FPH
/bank_ABQM/led1
/bank_ABQM/led2
/bank_ABQM/led3
/bank_ABQM/clear_...
/bank_ABQM/counter
/bank_ABQM/count...
/bank_ABQM/bin_rep
/bank_ABQM/fflag
/bank_ABQM/eflag
/bank_ABQM/flag
/bank_ABQM/tellers
/bank_ABQM/wtime
/bank_ABQM/most7...
/bank_ABQM/lest7seg
/bank_ABQM/alarm
Now
2000 ps
Cursor 1
1446 ps
800 ps
900 ps
1000 ps
1100 ps
1200 ps
1300 ps
1400 ps
1500 ps
1600 ps
1700 p


flag_generator:generator_Flag
BPH
clear_flag
counter[3..0]
counter_ch[3..0]
fflag
counter_cust:couter_1
BPH
FPH
clk_flag
rst
counter[2..0]
counter_ch[2..0]
fflag
clk_flag:G1
clk
rst
flag
bcd_7_seg:bcd
1'h0 A
B
C
D
rom:mmm1
Pcount[2..0]
Tcount[1..0]
Wtime[7..0]
waiting_time_display:wait_dispaly
wtime[7..0]
Teller_count:teller__countt
num_tellers[2..0]
binary_rep[1..0]
BPH
FPH
clock
clear_flags
tellers[2..0]