Blake Jonson
9663980
R: 5-7:50

ECE156a HW5

Introduction:

In this homework, I created 4 new modules including test benches to test each one. The first module PiggyBank keeps track of and updates the credit register. The second module is the purchase manager which determines if there is enough credit to purchase an item as well as which item to purchase. The third module is the display manager and it uses the seven segment decoder to display error, which item was purchased and the available credit. The fourth and final module is the Vending machine which is a compilation of the other three modules as well as the coin sensor.

Testing:

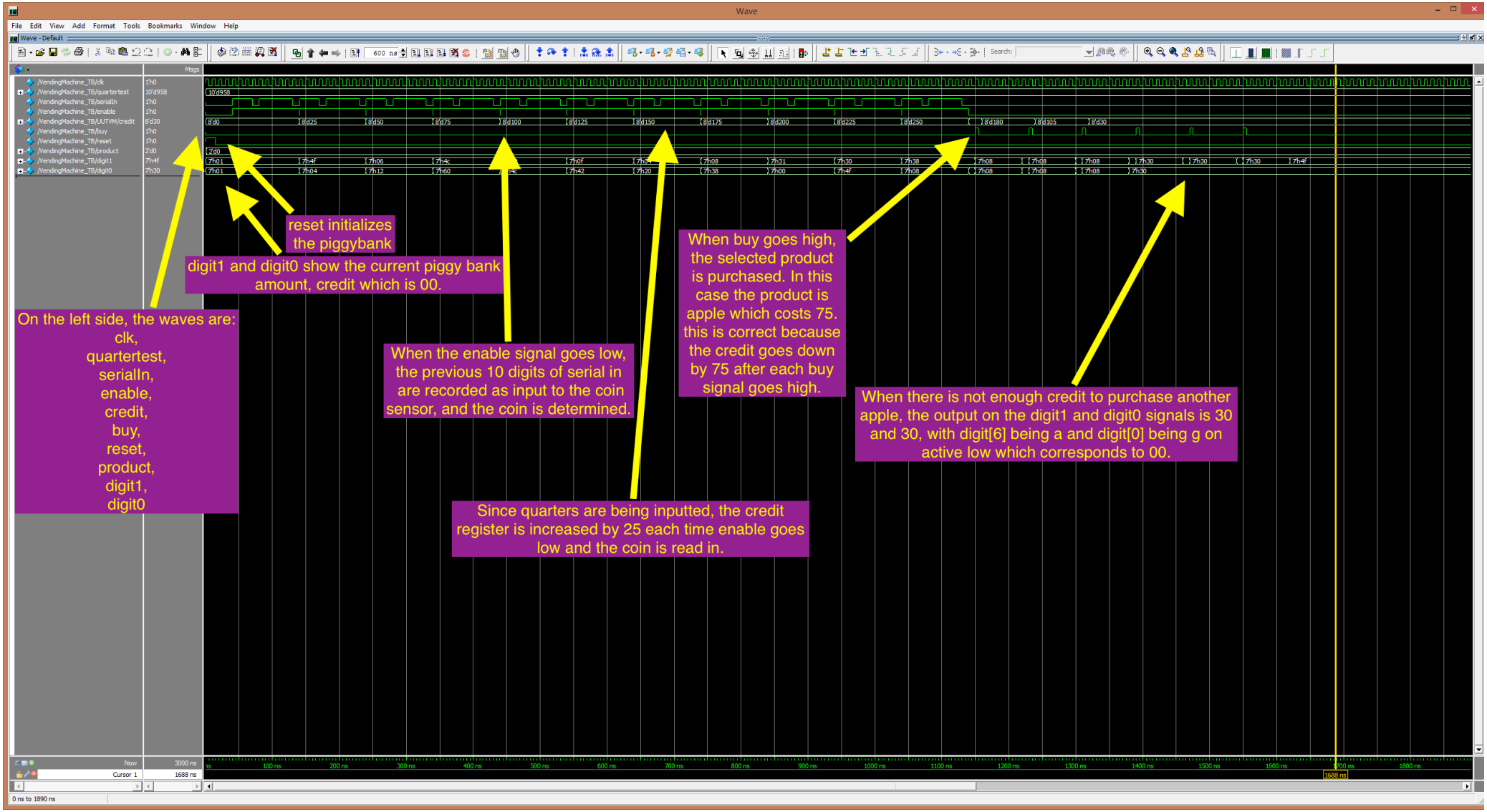
For each individual module, I made tests to test the functionality. Display manager was the most difficult individual module to test due to the tricky behavior with the downcounter and reset. The main testing was performed in the VendingMacine_TB module, because this testing the behavior of all the modules put together. In this module, I used randomized values for the coin diameters to make sure the inputs could take all accepted values and make the internal signals be what they are intended. After that, I made a register variable and I assign to it the different coins' registers to test the coin input ability for different coins. That way I can quickly change the behavior of the module to test many scenarios. I also have a spot where I assign product to different values so that I can quickly cycle through the different available products. I clearly label where to change the value of a for loop so that you can change the number of coins inputted to test different cases. In my first test where I purchase apples with quarters, I test the overflow functionality of credit.

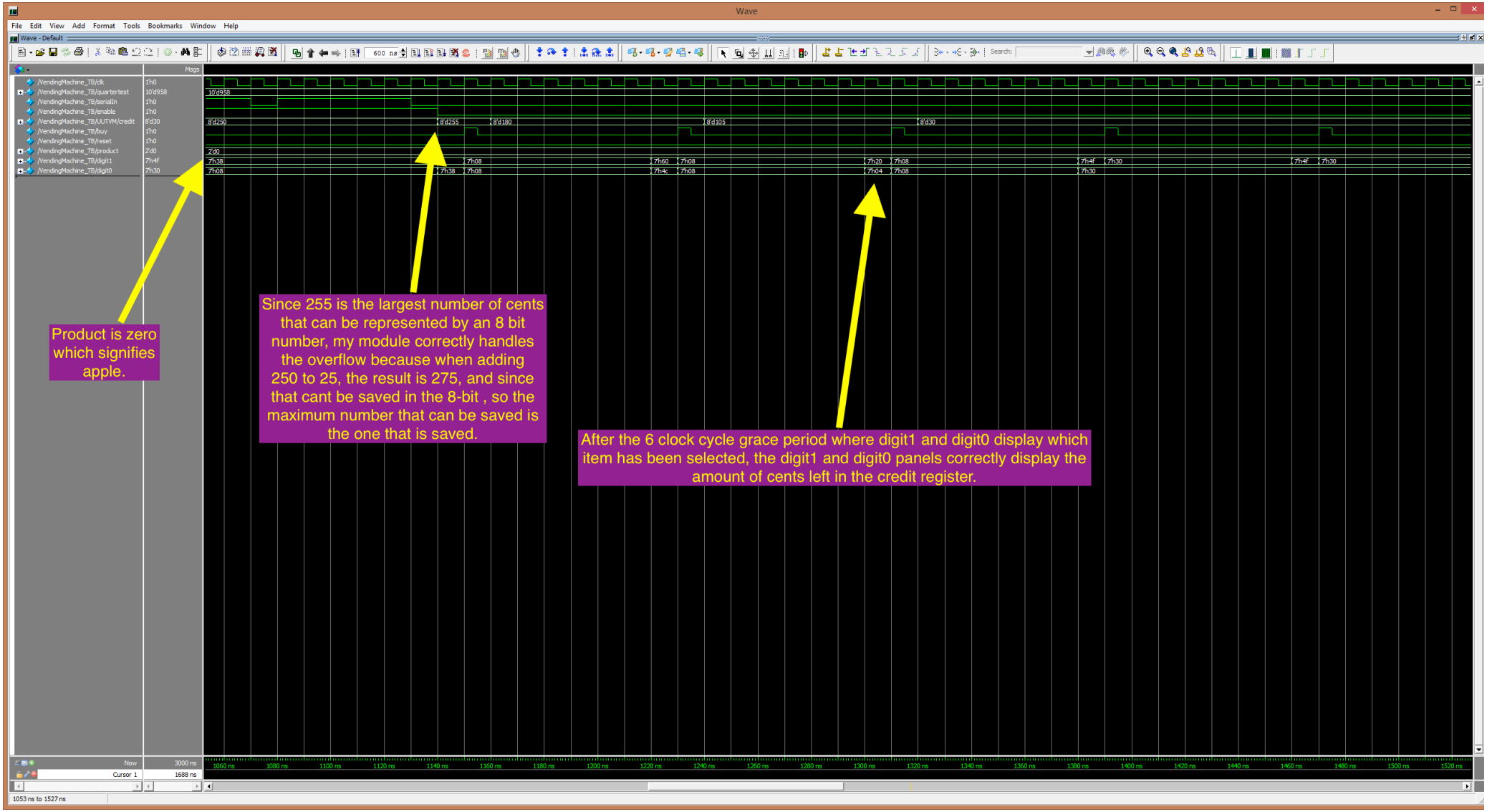
Results:

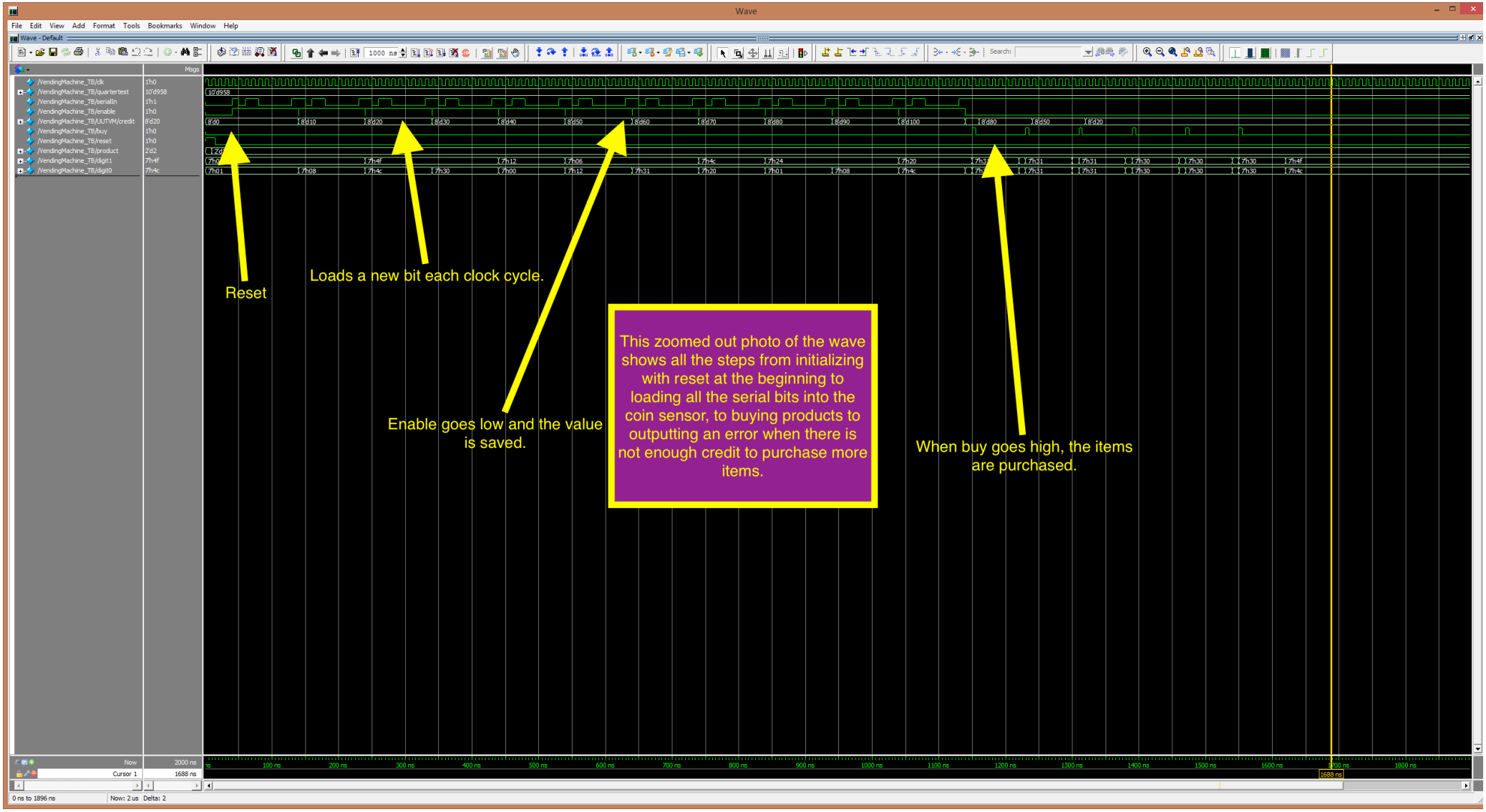
After looking at the waveforms and tweaking certain things in my modules, my results are correct and cover the error edge case, overflow, the display change to show the value of credit after 6 cycles, as well as the expected case where the display would show the item purchased. My vending machine also correctly inputs all types of coins, updates credit to the correct value, sells all the required products, and decrements credit after purchasing items.

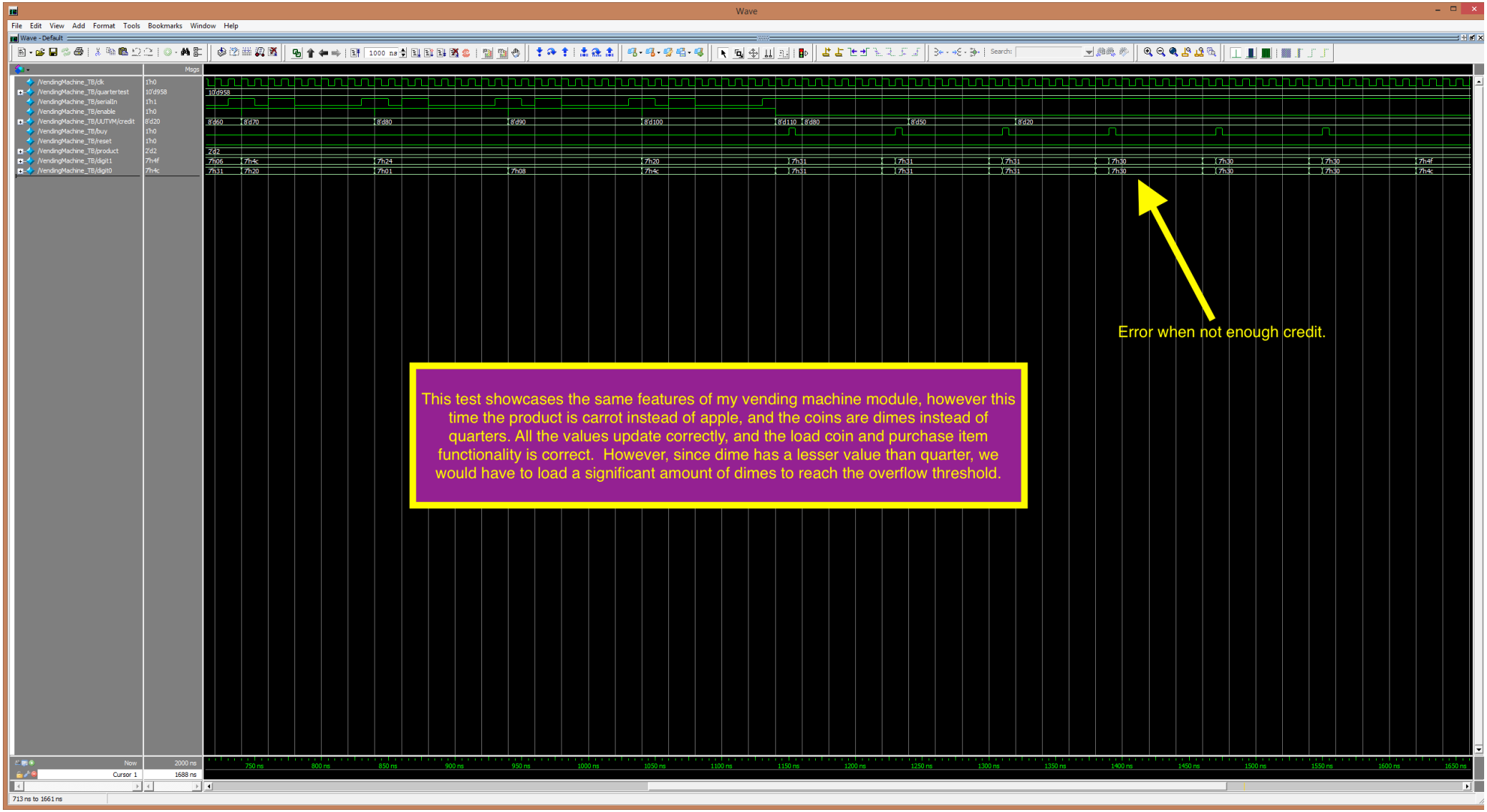
Conclusion:

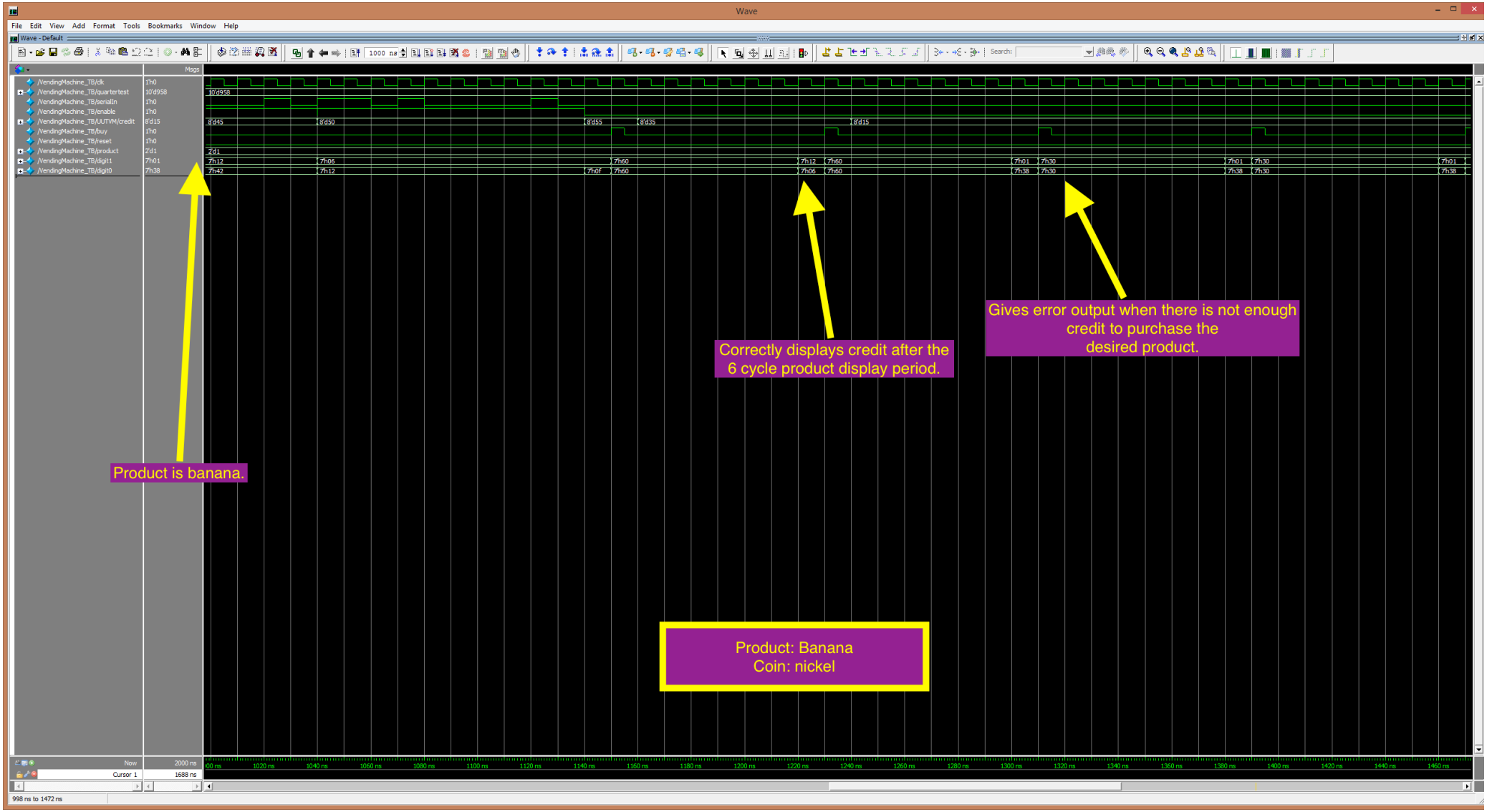
Vending Machine is the compilation of multiple sub-modules to form one main vending machine module. PiggyBank correctly updates credit, DisplayManager correctly displays the output or error, coinSensor from the last homework still works correctly, and purchase manager sends an error signal if there is not enough credit and designates which product is to be purchased. Each module has been thoroughly tested and is confirmed to work as it should.

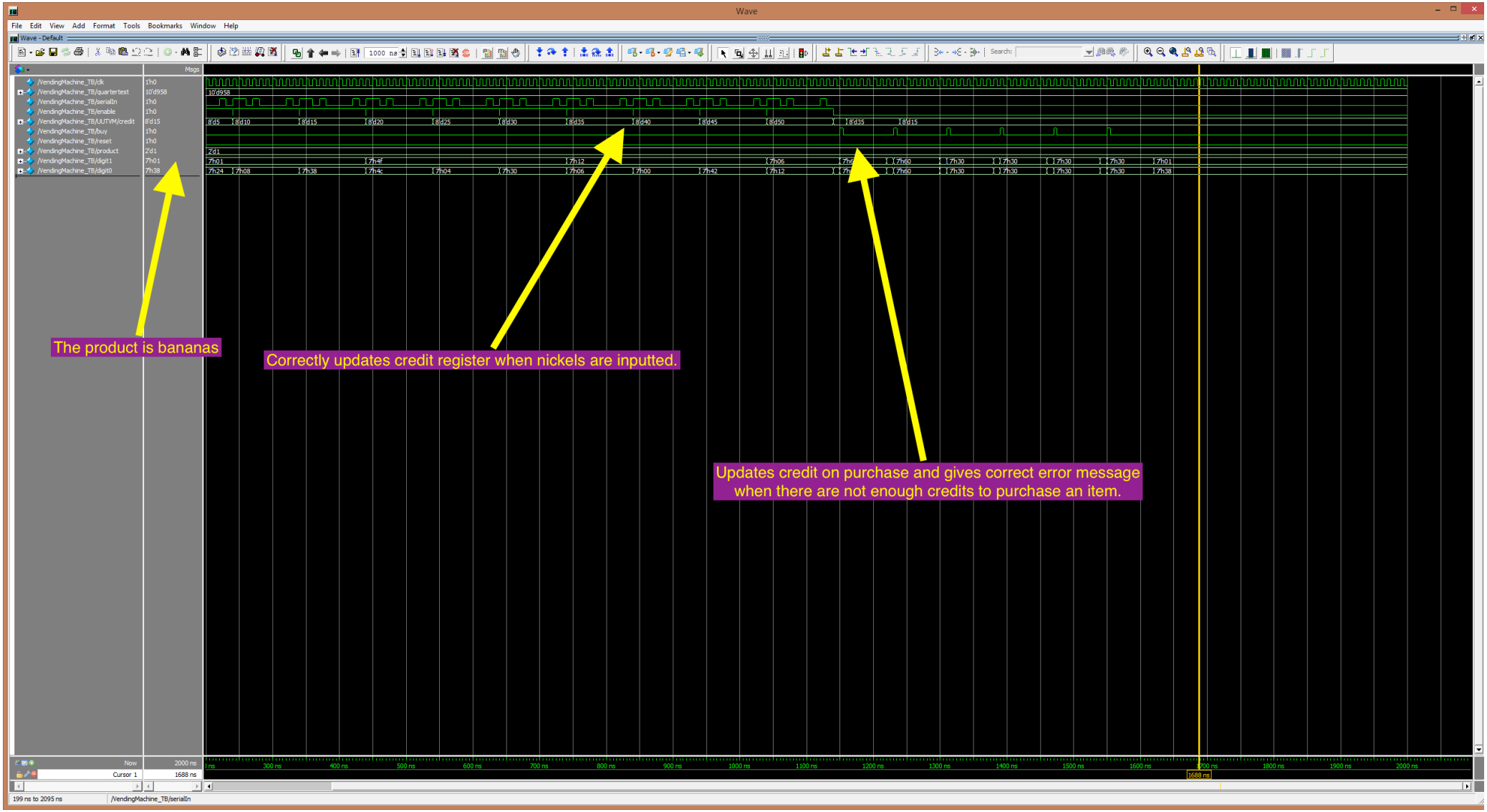

Wave
File Edit View Add Format Tools Bookmarks Window Help
/vendingMachine_TB/clk
/vendingMachine_TB/quartertest
/vendingMachine_TB/serialIn
/vendingMachine_TB/enable
/vendingMachine_TB/UUTVM/credit
/vendingMachine_TB/buy
/vendingMachine_TB/reset
/vendingMachine_TB/product
/vendingMachine_TB/digit1
/vendingMachine_TB/digit0
On the left side, the waves are: clk, quartertest, serialIn, enable, credit, buy, reset, product, digit1, digit0
reset initializes the piggybank
digit1 and digit0 show the current piggy bank amount, credit which is 00.
When the enable signal goes low, the previous 10 digits of serial in are recorded as input to the coin sensor, and the coin is determined.
Since quarters are being inputted, the credit register is increased by 25 each time enable goes low and the coin is read in.
When buy goes high, the selected product is purchased. In this case the product is apple which costs 75. this is correct because the credit goes down by 75 after each buy signal goes high.
When there is not enough credit to purchase another apple, the output on the digit1 and digit0 signals is 30 and 30, with digit[6] being a and digit[0] being g on active low which corresponds to 00.

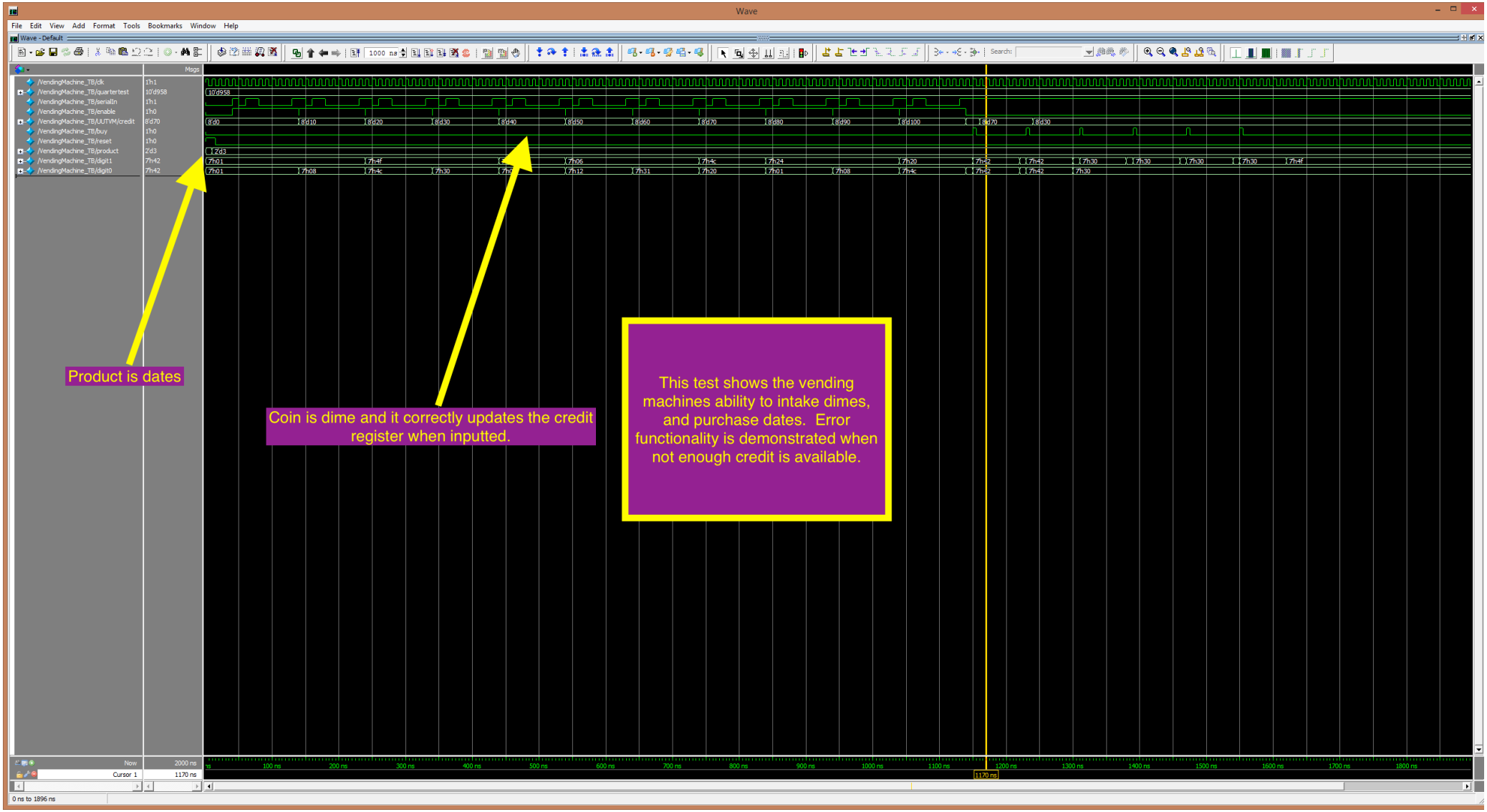

Wave
File Edit View Add Format Tools Bookmarks Window Help
/VendingMachine_TB/clk
/VendingMachine_TB/quartertest
/VendingMachine_TB/serialIn
/VendingMachine_TB/enable
/VendingMachine_TB/UUTVM/credit
/VendingMachine_TB/buy
/VendingMachine_TB/reset
/VendingMachine_TB/product
/VendingMachine_TB/digit1
/VendingMachine_TB/digit0
Product is zero which signifies apple.
Since 255 is the largest number of cents that can be represented by an 8 bit number, my module correctly handles the overflow because when adding 250 to 25, the result is 275, and since that cant be saved in the 8-bit , so the maximum number that can be saved is the one that is saved.
After the 6 clock cycle grace period where digit1 and digit0 display which item has been selected, the digit1 and digit0 panels correctly display the amount of cents left in the credit register.

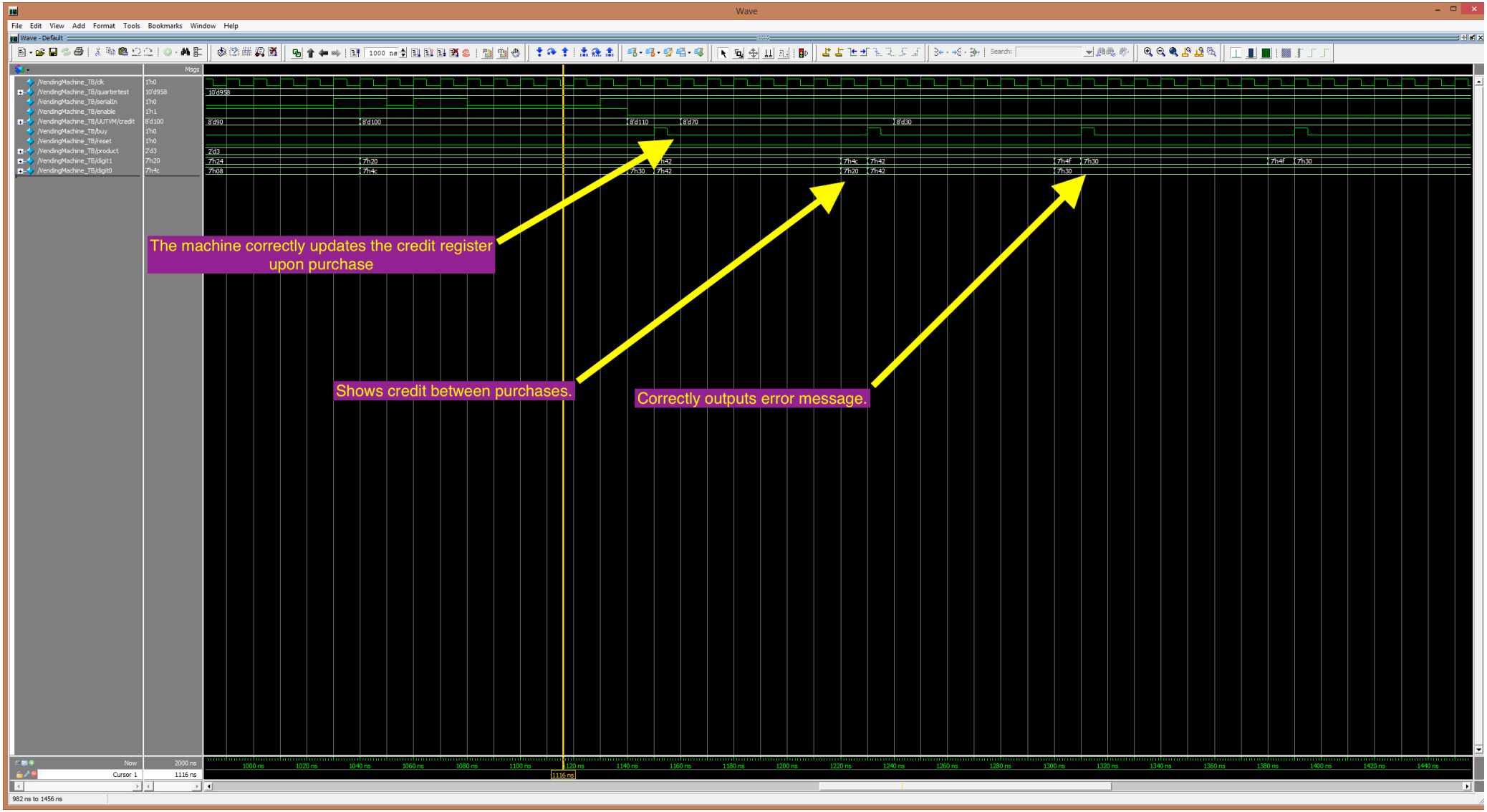

Wave
File Edit View Add Format Tools Bookmarks Window Help
/VendingMachine_TB/clk
/VendingMachine_TB/quartertest
/VendingMachine_TB/serialIn
/VendingMachine_TB/enable
/VendingMachine_TB/UUTVM/credit
/VendingMachine_TB/buy
/VendingMachine_TB/reset
/VendingMachine_TB/product
/VendingMachine_TB/digit1
/VendingMachine_TB/digit0
Reset
Loads a new bit each clock cycle.
Enable goes low and the value is saved.
This zoomed out photo of the wave shows all the steps from initializing with reset at the beginning to loading all the serial bits into the coin sensor, to buying products to outputting an error when there is not enough credit to purchase more items.
When buy goes high, the items are purchased.

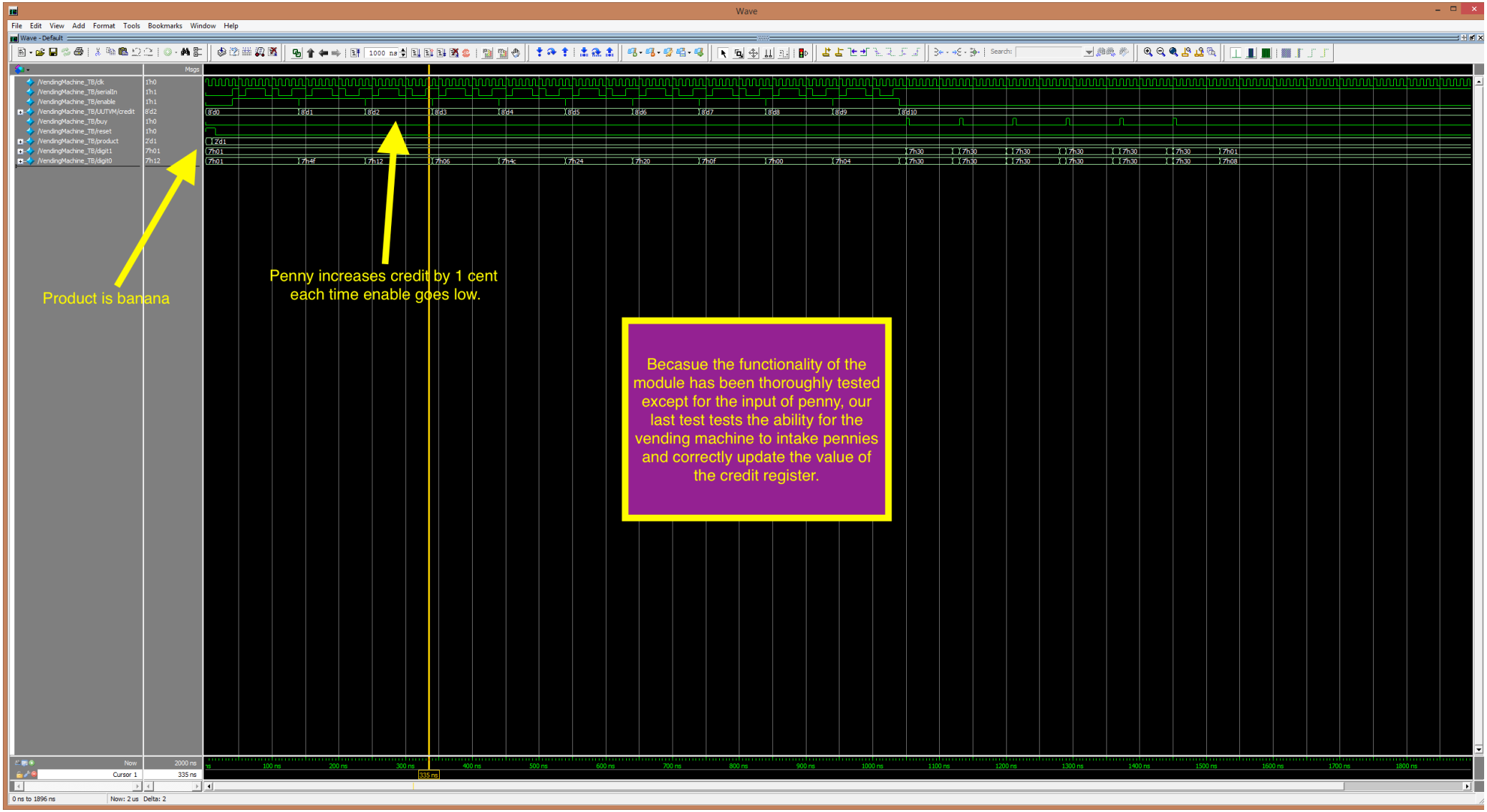

Wave
File Edit View Add Format Tools Bookmarks Window Help
/VendingMachine_TB/clk
/VendingMachine_TB/quartertest
/VendingMachine_TB/serialIn
/VendingMachine_TB/enable
/VendingMachine_TB/UUTVM/credit
/VendingMachine_TB/buy
/VendingMachine_TB/reset
/VendingMachine_TB/product
/VendingMachine_TB/digit1
/VendingMachine_TB/digit0
Error when not enough credit.
This test showcases the same features of my vending machine module, however this time the product is carrot instead of apple, and the coins are dimes instead of quarters. All the values update correctly, and the load coin and purchase item functionality is correct. However, since dime has a lesser value than quarter, we would have to load a significant amount of dimes to reach the overflow threshold.


Wave
File Edit View Add Format Tools Bookmarks Window Help
/VendingMachine_TB/clk
/VendingMachine_TB/quartertest
/VendingMachine_TB/serialIn
/VendingMachine_TB/enable
/VendingMachine_TB/UUTVM/credit
/VendingMachine_TB/buy
/VendingMachine_TB/reset
/VendingMachine_TB/product
/VendingMachine_TB/digit1
/VendingMachine_TB/digit0
Product is banana.
Correctly displays credit after the 6 cycle product display period.
Gives error output when there is not enough credit to purchase the desired product.
Product: Banana
Coin: nickel


Wave
File Edit View Add Format Tools Bookmarks Window Help
/VendingMachine_TB/clk
/VendingMachine_TB/quartertest
/VendingMachine_TB/serialIn
/VendingMachine_TB/enable
/VendingMachine_TB/UUTVM/credit
/VendingMachine_TB/buy
/VendingMachine_TB/reset
/VendingMachine_TB/product
/VendingMachine_TB/digit1
/VendingMachine_TB/digit0
The product is bananas
Correctly updates credit register when nickels are inputted.
Updates credit on purchase and gives correct error message when there are not enough credits to purchase an item.


Wave
File Edit View Add Format Tools Bookmarks Window Help
/vendingMachine_TB/clk
/vendingMachine_TB/quartertest
/vendingMachine_TB/serialIn
/vendingMachine_TB/enable
/vendingMachine_TB/UUTVM/credit
/vendingMachine_TB/buy
/vendingMachine_TB/reset
/vendingMachine_TB/product
/vendingMachine_TB/digit1
/vendingMachine_TB/digit0
Product is dates
Coin is dime and it correctly updates the credit register when inputted.
This test shows the vending machines ability to intake dimes, and purchase dates. Error functionality is demonstrated when not enough credit is available.


Wave
File Edit View Add Format Tools Bookmarks Window Help
/VendingMachine_TB/clk
/VendingMachine_TB/quartertest
/VendingMachine_TB/serialIn
/VendingMachine_TB/enable
/VendingMachine_TB/UUTVM/credit
/VendingMachine_TB/buy
/VendingMachine_TB/reset
/VendingMachine_TB/product
/VendingMachine_TB/digit1
/VendingMachine_TB/digit0
The machine correctly updates the credit register upon purchase
Shows credit between purchases.
Correctly outputs error message.


Wave
File Edit View Add Format Tools Bookmarks Window Help
/VendingMachine_TB/clk
/VendingMachine_TB/serialIn
/VendingMachine_TB/enable
/VendingMachine_TB/UUT/VM/credit
/VendingMachine_TB/buy
/VendingMachine_TB/reset
/VendingMachine_TB/product
/VendingMachine_TB/digit1
/VendingMachine_TB/digit0
Product is banana
Penny increases credit by 1 cent each time enable goes low.
Becasue the functionality of the module has been thoroughly tested except for the input of penny, our last test tests the ability for the vending machine to intake pennies and correctly update the value of the credit register.