TEST NUMBER 1

Generics values:

|  |  |  |
| --- | --- | --- |
| Comments | Value | Name |
|  | 1 | reset\_polarity\_g |
|  | 1 | enable\_polarity\_g |
| Each the number og addresses is RAM in | 3 | signal\_ram\_depth\_g |
| "word" width – line in the RAM | 8 | signal\_ram\_width\_g |
| The number of recorded signals is | 5 | record\_depth\_g |
| BUS width of the information entering the component | 8 | data\_width\_g |
| Word width in WB protocol | 8 | Add\_width\_g |
| Number of signals we record in each iteration | 8 | num\_of\_signals\_g |
| Output width is exactly like input width | 0 | power2\_out\_g |
| Output and input width are the same | 1 | power\_sign\_g |
| Type depth. We got 3 WM and 3 WS -> 6 total whisebone entities | 6 | Type\_d\_g |
| Length depth. | 1 | Len\_d\_g |

Variables values:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Comments | Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Name |
| Inputs | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | 50 duty cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | clk |
|  | | 120-30000 | | | | | | | | | | | | | | | | 0-120 | | | | | | | | | | | | | Reset |
| 0 | | | | | | | | | | | | | | | | 1 | | | | | | | | | | | | |
| Counter | | 0-400 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | data\_in |
|  | | 25600-30000 | 7500-25600 | | | | | 5300-7500 | | | | | | 3100-5300 | | | 1000-3100 | | | | | | | 780-1000 | | | | 0-780 | | | Trigger |
| 0 | 1 | | | | | 0 | | | | | | 1 | | | 0 | | | | | | | 1 | | | | 0 | | |
| Address of register to write to | | 1400 -30000 | | | 1000 - 1400 | | | | | | | | | | 700 - 1000 | | | | 400 -700 | | | | | | | | | | 0-400 | | ADR\_I |
|  | | 4 | | | 0 | | | | | | | | | | 3 | | | | 2 | | | | | | | | | | 1 | |  |
| Data stored is registers. (3 register is clk\_to\_cycle which is not in use) | | 3000-30000 | | | 1000-3000 | | | | | | | | | | 700-1000 | | | | 400-700 | | | | | | | | | | 0-400 | | DAT\_I |
| 0 | | | 1 | | | | | | | | | | 7 | | | | 50 | | | | | | | | | | 1 | |
| 1 for write. | | 0-22000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WE\_I |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data in registers valid | | other | | 23100-23200 | | | | | 2500-2600 | | 1300 -1400 | | | | | 1000-1100 | | | | | | 700-800 | | | 400-500 | | | | | 200-300 | STB\_I | |
| 0 | | 1 | | | | | 1 | | 1 | | | | | 1 | | | | | | 1 | | | 1 | | | | | 1 |
| 1 for bus transmition request for input registers data | | 25000-30000 | | | | 23000-25000 | | | | | | 3000-23000 | | | | | | | | 150-3000 | | | | | | 0-150 | | | | | CYC\_I |
| 0 | | | | 1 | | | | | | 0 | | | | | | | | 1 | | | | | | 0 | | | | |
| type of core WBS | | 0-30000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | TGA\_I |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Length of the word (maybe will change) | | 0-30000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | TGD\_I |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| maybe will change | | 0-30000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | stall |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Come from external WBS | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ACK\_I |
| Come from external WBS | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | STALL\_I |
| Come from external WBS | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ERR\_I |
| Outputs | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CYC\_I output to user side. | | 25000-30000 | | | | 23000-25000 | | | | | | 3000-23000 | | | | | | | | 150-3000 | | | | | | 0-150 | | | | | TOP\_active\_cycle |
| 0 | | | | 1 | | | | | | 0 | | | | | | | | 1 | | | | | | 0 | | | | |
| Follow STB\_I | | other | | 23100-23200 | | | | | 2500-2600 | | 1300 -1400 | | | | | 1000-1100 | | | | | | 700-800 | | | 400-500 | | | | | 200-300 | ACK\_O |
| 0 | | 1 | | | | | 1 | | 1 | | | | | 1 | | | | | | 1 | | | 1 | | | | | 1 |
| We do not output data from WBS | | 0-30000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WS\_DAT\_O |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| We do not output data from WBS | | 0-30000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | STALL\_O |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Maybe an error | | 0-30000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | wm\_end\_out |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Maybe an error | | 7650-30000 | | | | | | | | 7500-7650 | | | | | | | | | | | | | 0-7500 | | | | | | | | ADR\_O |
| 2 | | | | | | | | 1 | | | | | | | | | | | | | 0 | | | | | | | |
| Output the relevant data | | 20650-30000 | | | | | | | | 7750-20650 | | | | | | | | | | | | | 0-7750 | | | | | | | | WM\_DAT\_O |
| 0 | | | | | | | | Relevant data | | | | | | | | | | | | | 0 | | | | | | | |
|  | | 7650-30000 | | | | | | | | 7450-7650 | | | | | | | | | | | | | 0-7450 | | | | | | | | WE\_O |
| 0 | | | | | | | | 1 | | | | | | | | | | | | | 0 | | | | | | | |
|  | | 7650-30000 | | | | | | | | 7350-7650 | | | | | | | | | | | | | 0-7350 | | | | | | | | STB\_O |
| 0 | | | | | | | | 1 | | | | | | | | | | | | | 0 | | | | | | | |
| Maybe an error | | 7350-30000 | | | | | | | | | | | | | | 0-7350 | | | | | | | | | | | | | | | CYC\_O |
| 1 | | | | | | | | | | | | | | 0 | | | | | | | | | | | | | | |
| Output the data to WBS number 4 | | 7350-30000 | | | | | | | | | | | | | | 0-7350 | | | | | | | | | | | | | | | TGA\_O |
| 4 | | | | | | | | | | | | | | 0 | | | | | | | | | | | | | | |
| Length of data (in words) | | 7350-30000 | | | | | | | | | | | | | | 0-7350 | | | | | | | | | | | | | | | TGD\_O |
| 1 | | | | | | | | | | | | | | 0 | | | | | | | | | | | | | | |
| Internal signals | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Enable. Come from ENABLE - SM | | 27650-30000 | | | | | 23350-27650 | | | | | | 7350-23350 | | | | | | | | 2750-7350 | | | | | | 0-2750 | | | | Enable\_s |
| 0 | | | | | 1 | | | | | | 0 | | | | | | | | 1 | | | | | | 0 | | | |
| Output data from RC | | 28050-30000 | | | | | | | 20650-28050 | | | | | | | 7750-20650 | | | | | | | | | 0-7750 | | | | | | data\_from\_rc\_to\_wm\_s |
| DATA | | | | | | | 0 | | | | | | | DATA | | | | | | | | | 0 | | | | | |
| Validity of output data from RC | | Rise for one clock cycle in every new data that output through RC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | data\_from\_rc\_to\_wm\_valid\_s |
| Rise for one cycle when WC end saving all the data | | 27650-30000 | | | | | 27550-27650 | | | | | | 7350-27550 | | | | | | | | 7250-7350 | | | | | | 0-7250 | | | | write\_controller\_finish\_s |
| 0 | | | | | 1 | | | | | | 0 | | | | | | | | 1 | | | | | | 0 | | | |
| Rise for one cycle when RC end outputing all the data | | 20650-30000 | | | | | | | | 20550-20650 | | | | | | | | | | | | | 0-20550 | | | | | | | | read\_controller\_finish\_s |
| 0 | | | | | | | | 1 | | | | | | | | | | | | | 0 | | | | | | | |
| Rise after trigger found until WC finish working | | 37650-30000 | | | | | 25650-27650 | | | | | | 7350-25650 | | | | | | | | 5350-7350 | | | | | | 0-5350 | | | | trigger\_found\_s |
| 0 | | | | | 1 | | | | | | 0 | | | | | | | | 1 | | | | | | 0 | | | |

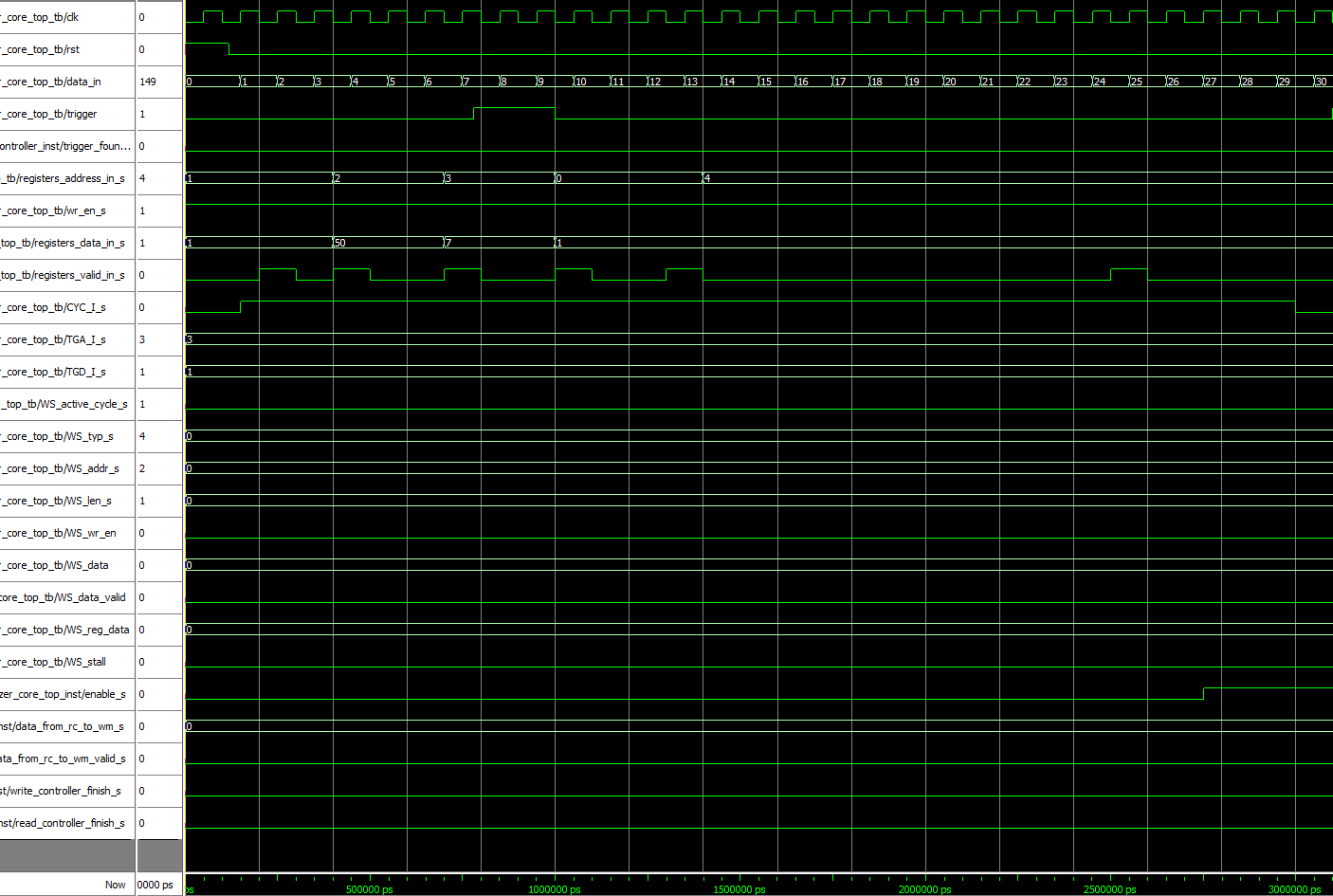
Explanation:

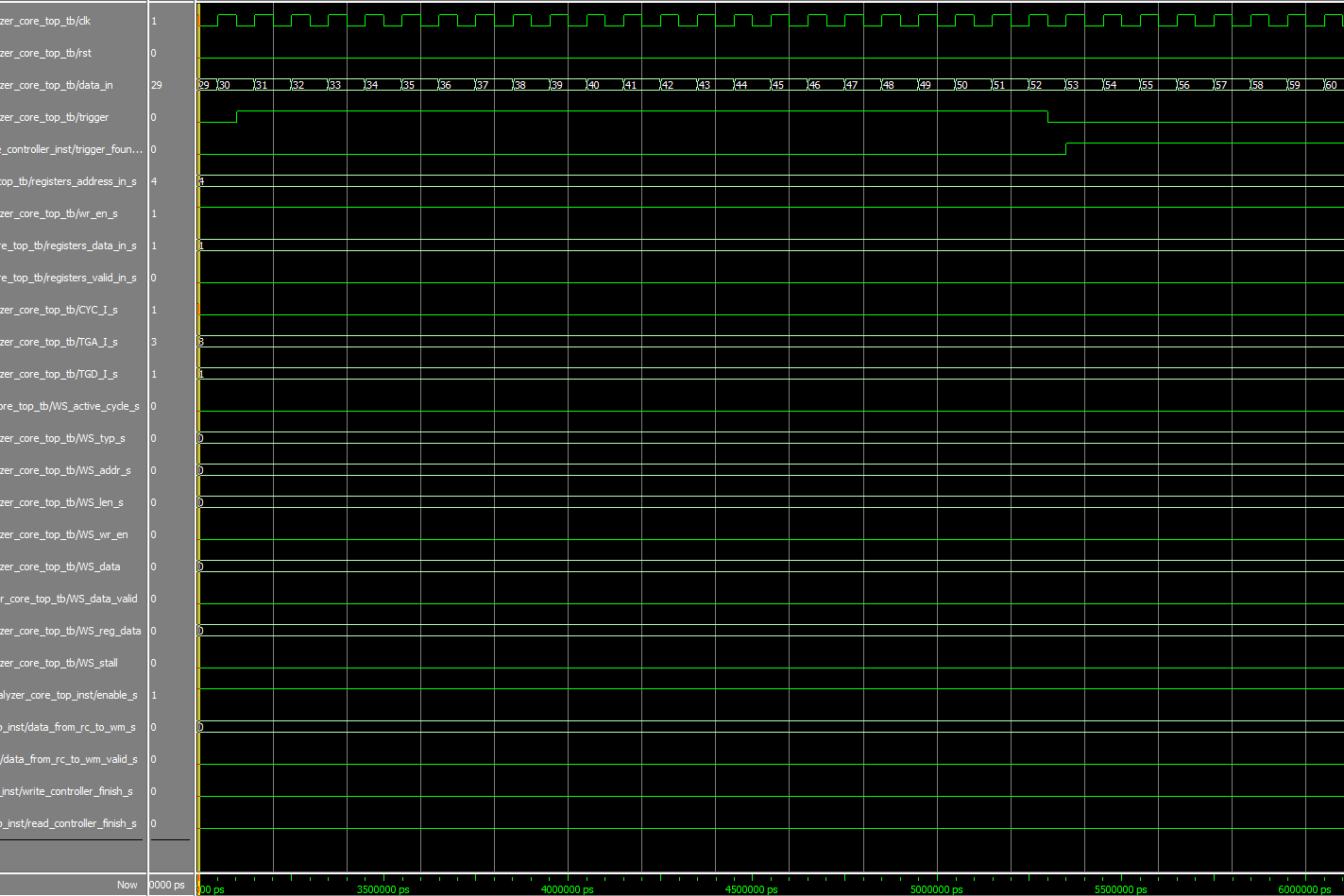
We input the user configurations through the WBS and save it in the registers, after that we wait for enable rise and the system start to save the data since that. When we detect trigger rise we continue to save the relevant data and after that output it through the WBM.

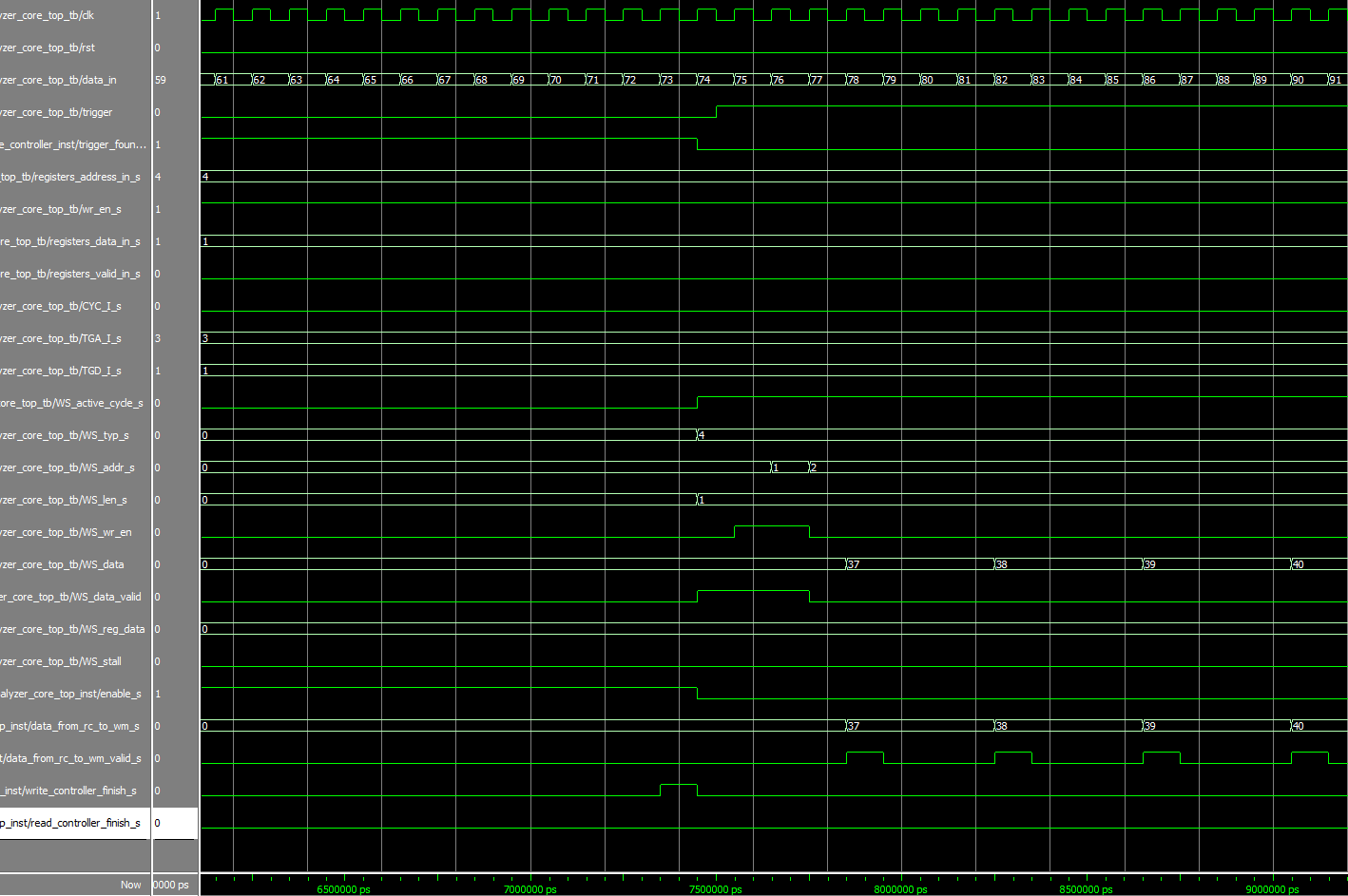
We can see that after that we finish outputting all the data we start again and the system wait for another enable rise (the other configurations do not have to change).

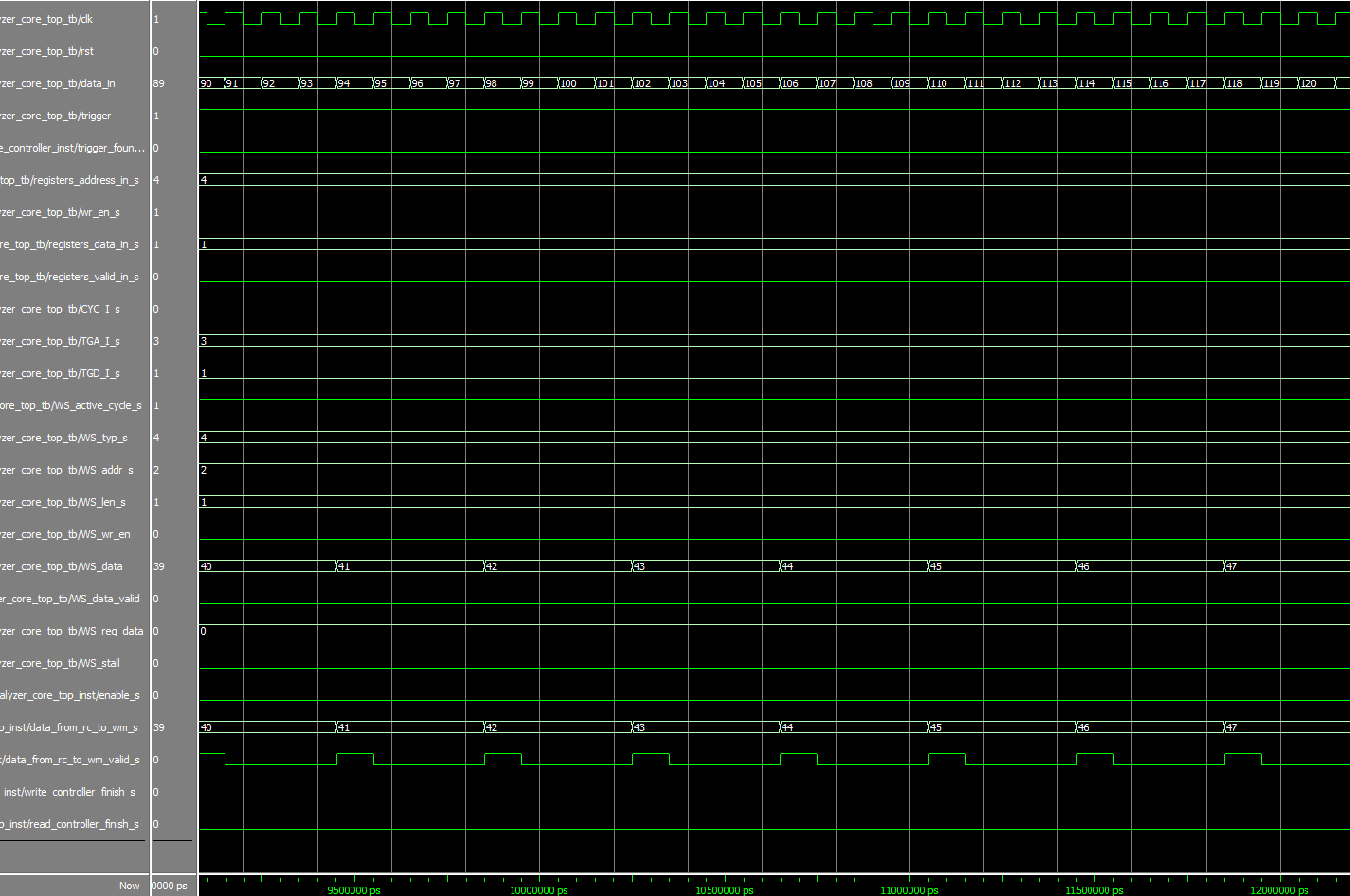
The data and trigger input are being read every cycle, but the data output is every 3 cycles.

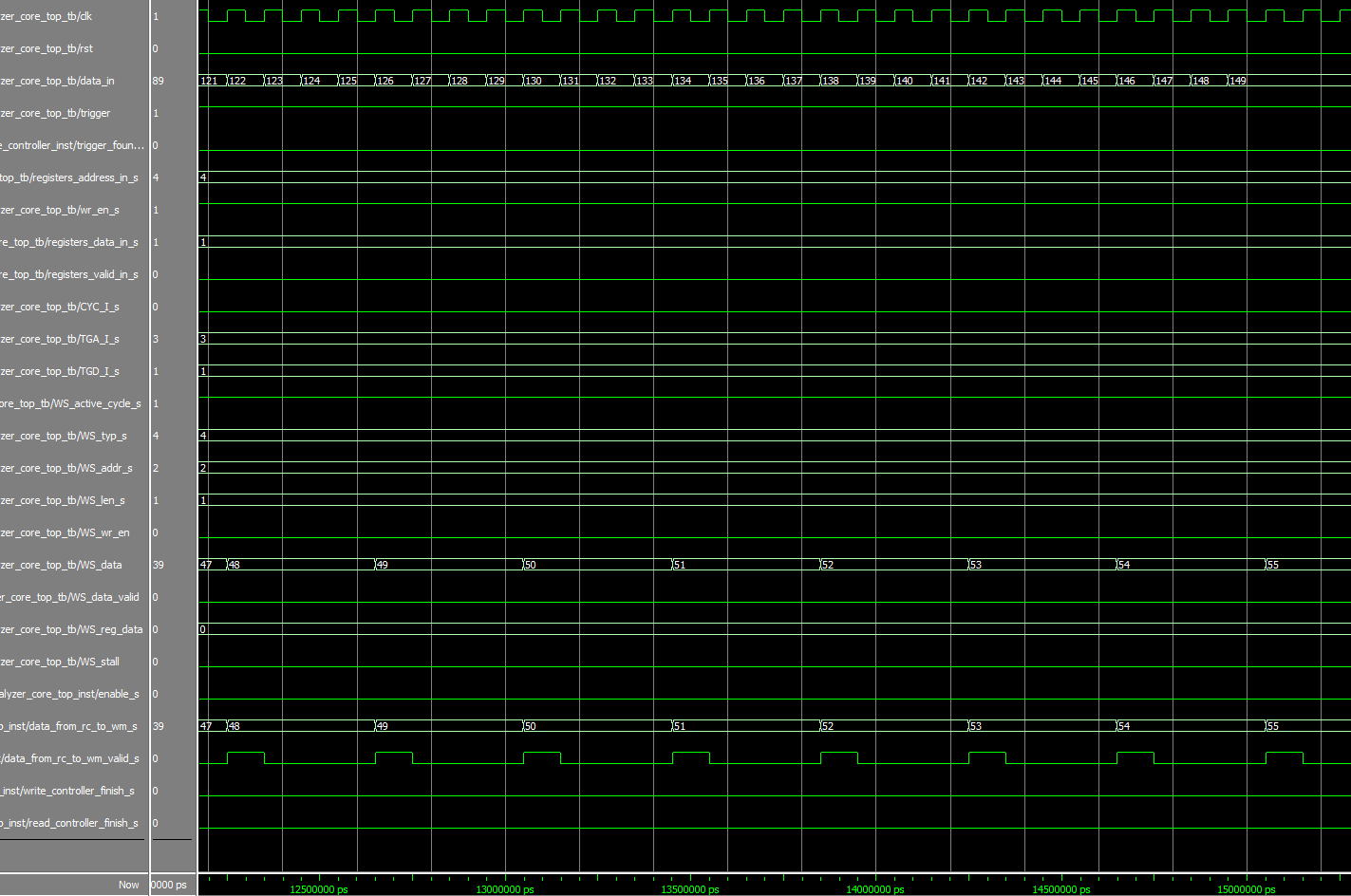
Simulation:

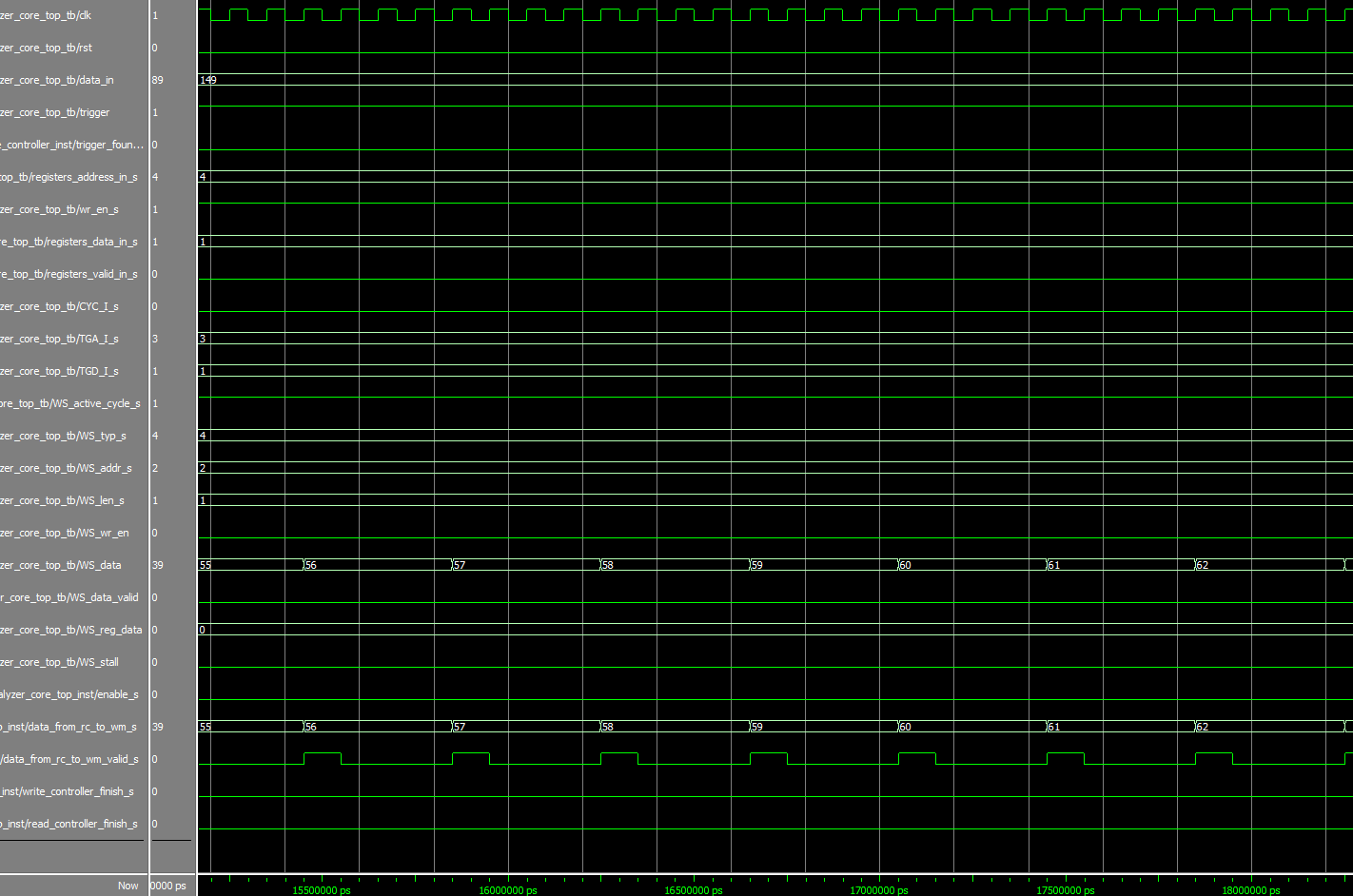


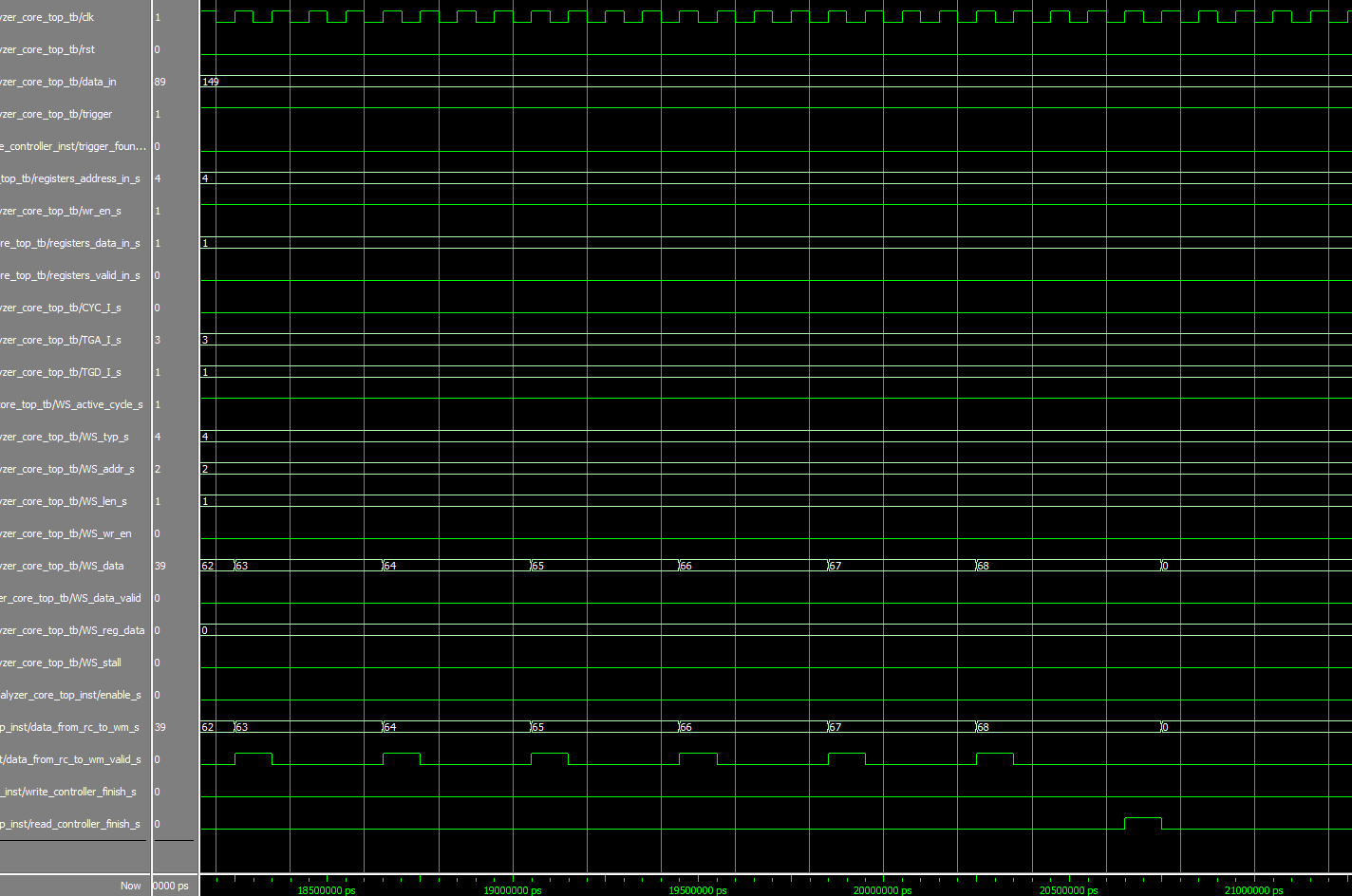












Analysis:

At first we initialize the registers with the user configurations (type is fall, position is 50% ) and then we wait for enable to rise (also save at a register). We then start to save the data at the RAM and searching for trigger fall. After we found the trigger fall we continue to save the rest of the data that is relevant to us and in the end we output the data through the WBM back to the user.

We can see that when the RC is finish outputting the data, the enable register is rested back to 0 in order to allow a new enable rise.

In order that the RC will start just after the WC, the wc\_finish signal is the enable of the RC.

Recorded depth = 5 , so we recording 32 signals when we start from 16 signals before trigger rise (position is 50%). Trigger rise at 53 so 16 cycles before is 37 until 68 (we can change it to 38 to 69), and we can see that this is the output.

TEST NUMBER 2

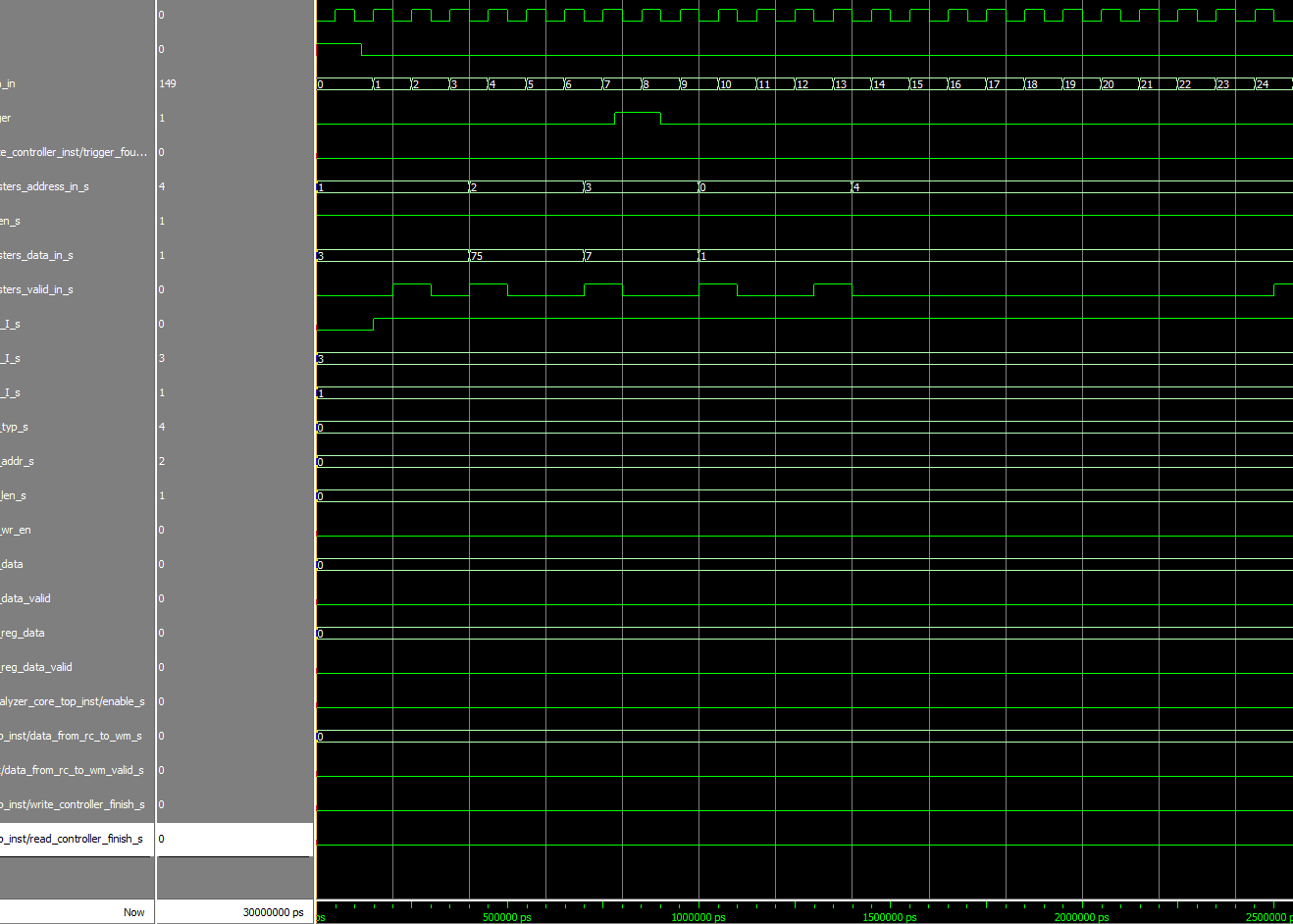
Generics values:

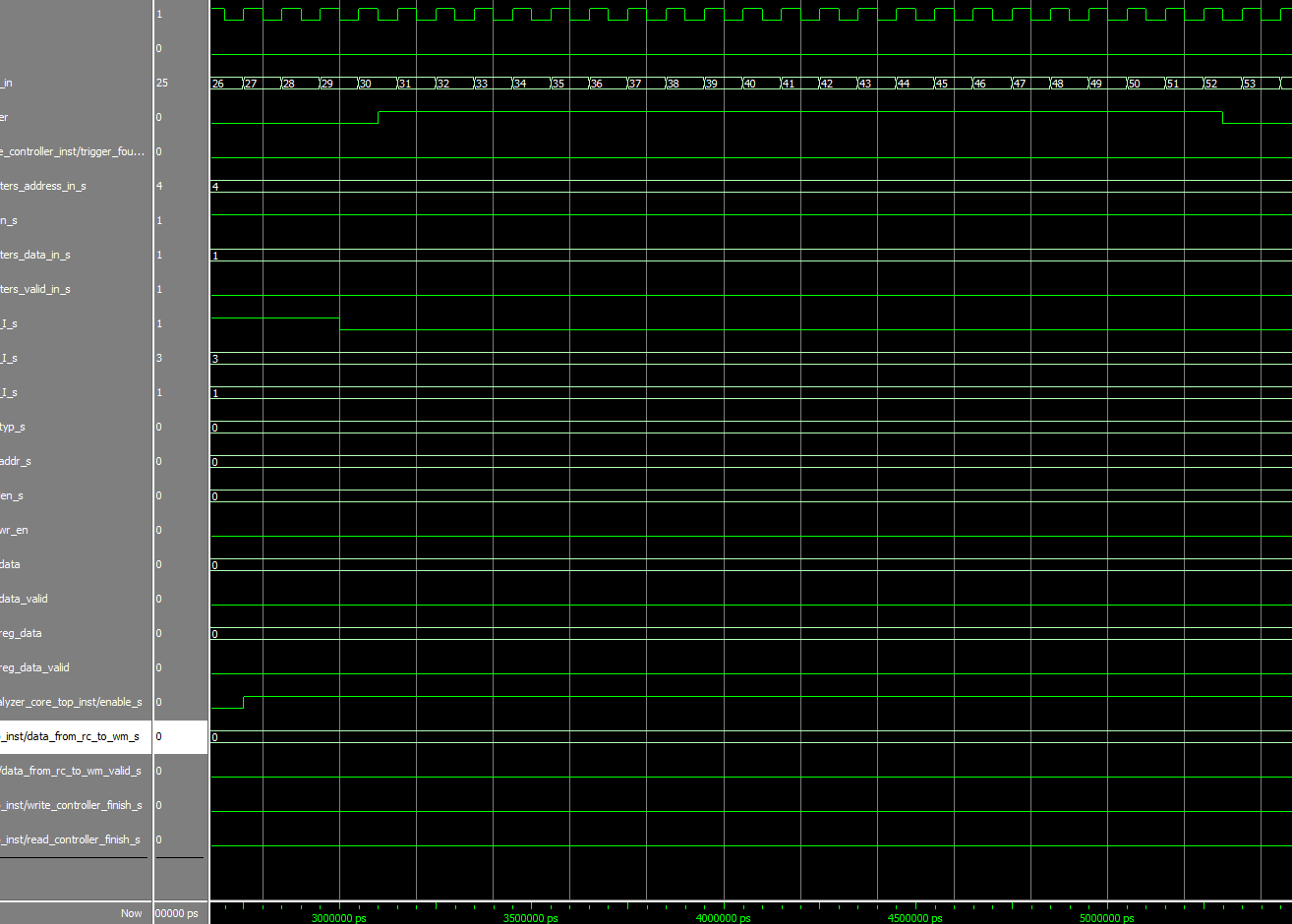
|  |  |  |
| --- | --- | --- |
| Comments | Value | Name |
|  | 1 | reset\_polarity\_g |
|  | 1 | enable\_polarity\_g |
| Each the number og addresses is RAM in | 3 | signal\_ram\_depth\_g |
| "word" width – line in the RAM | 8 | signal\_ram\_width\_g |
| The number of recorded signals is | 3 | record\_depth\_g |
| BUS width of the information entering the component | 8 | data\_width\_g |
| Word width in WB protocol | 8 | Add\_width\_g |
| Number of signals we record in each iteration | 8 | num\_of\_signals\_g |
| Output width is exactly like input width | 0 | power2\_out\_g |
| Output and input width are the same | 1 | power\_sign\_g |
| Type depth. We got 3 WM and 3 WS -> 6 total whisebone entities | 6 | Type\_d\_g |
| Length depth. | 1 | Len\_d\_g |

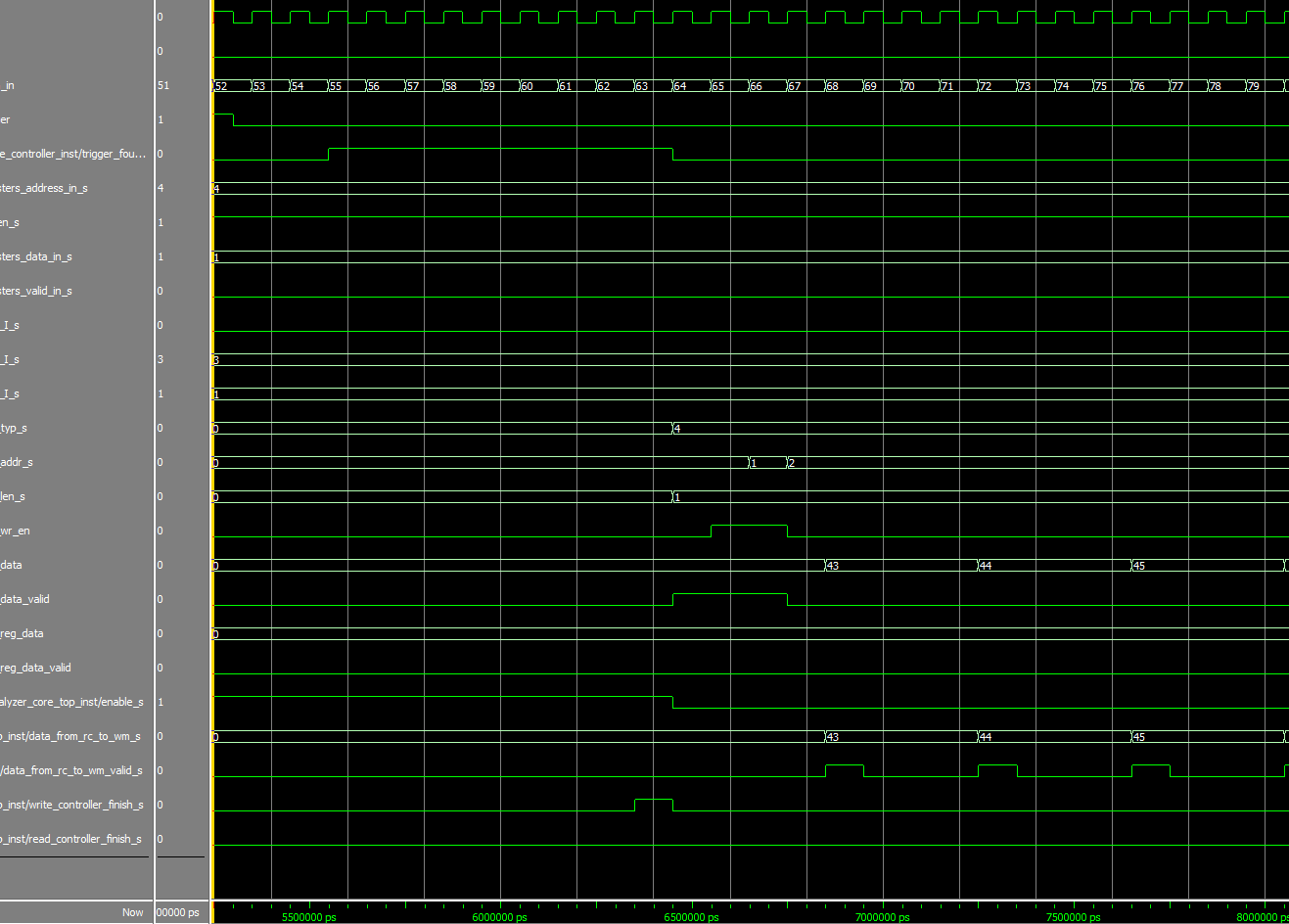
Explanation:

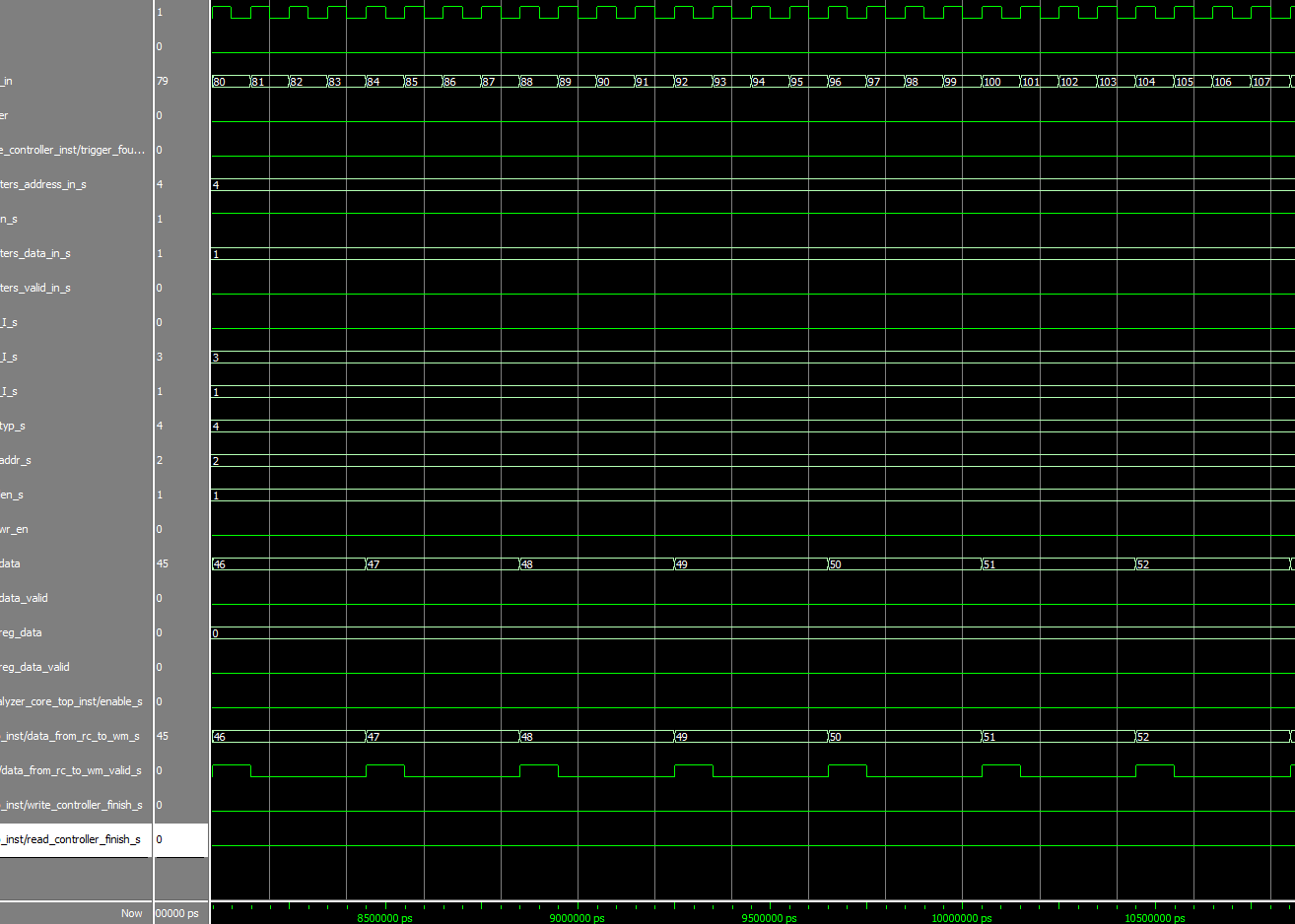
We change the recording depth to 4 (2^4 = 16), position to 75% (meaning that 75% -> 12 bits, will recorded before the trigger and the other after that, we also change the trigger type to zeroes (number 3 ) so trigger will rise after three low sampling.

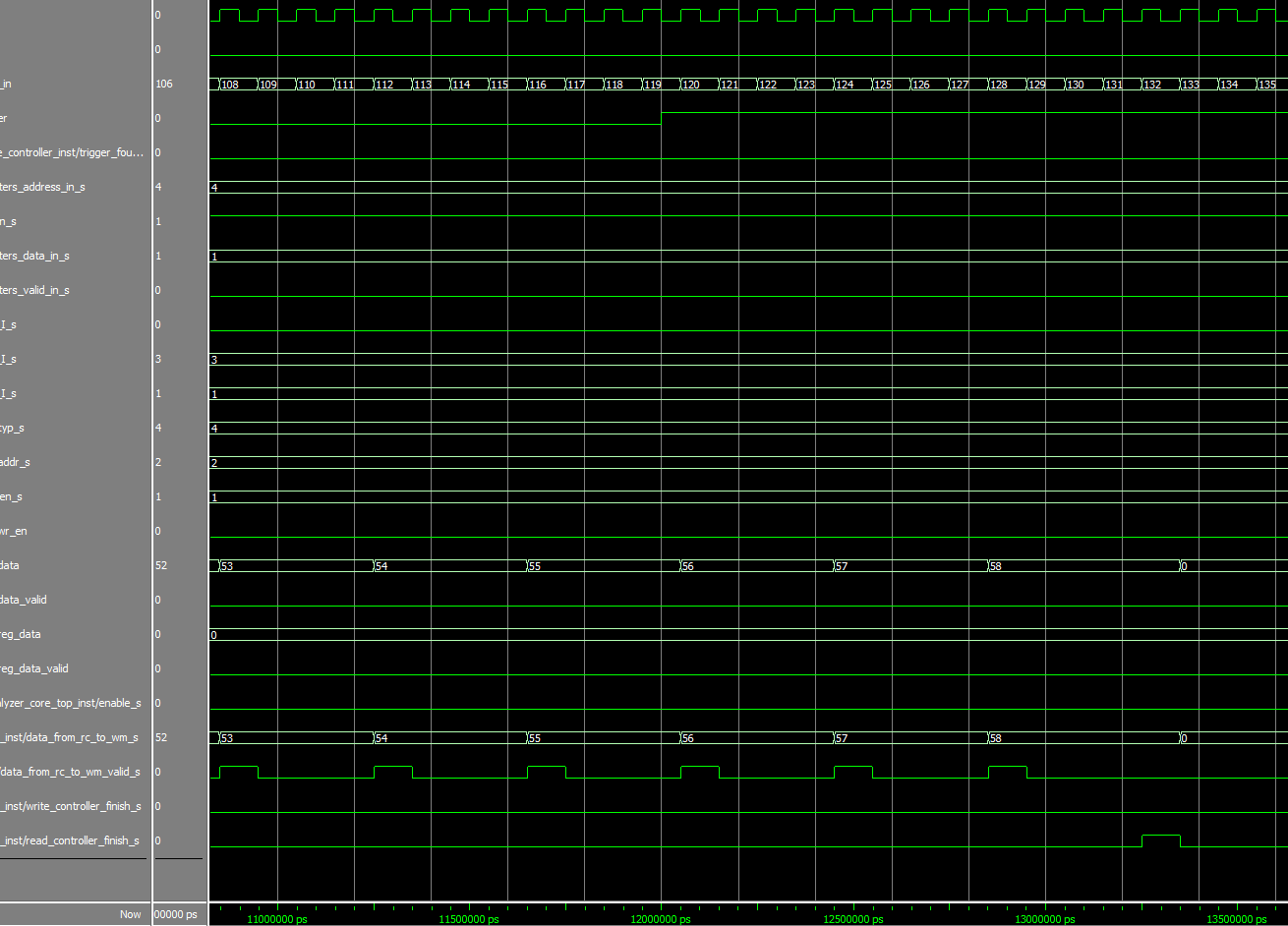
Simulation:

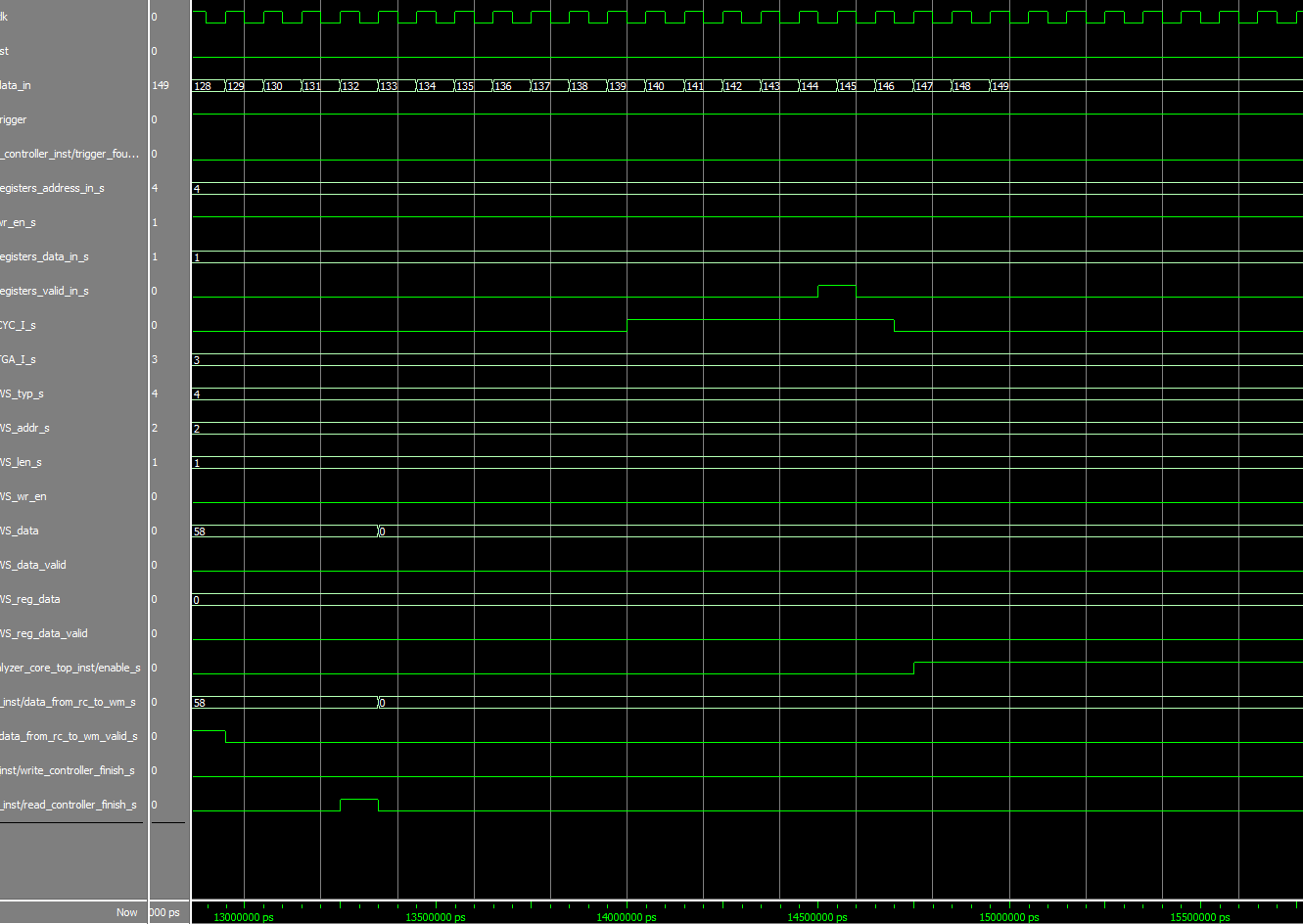


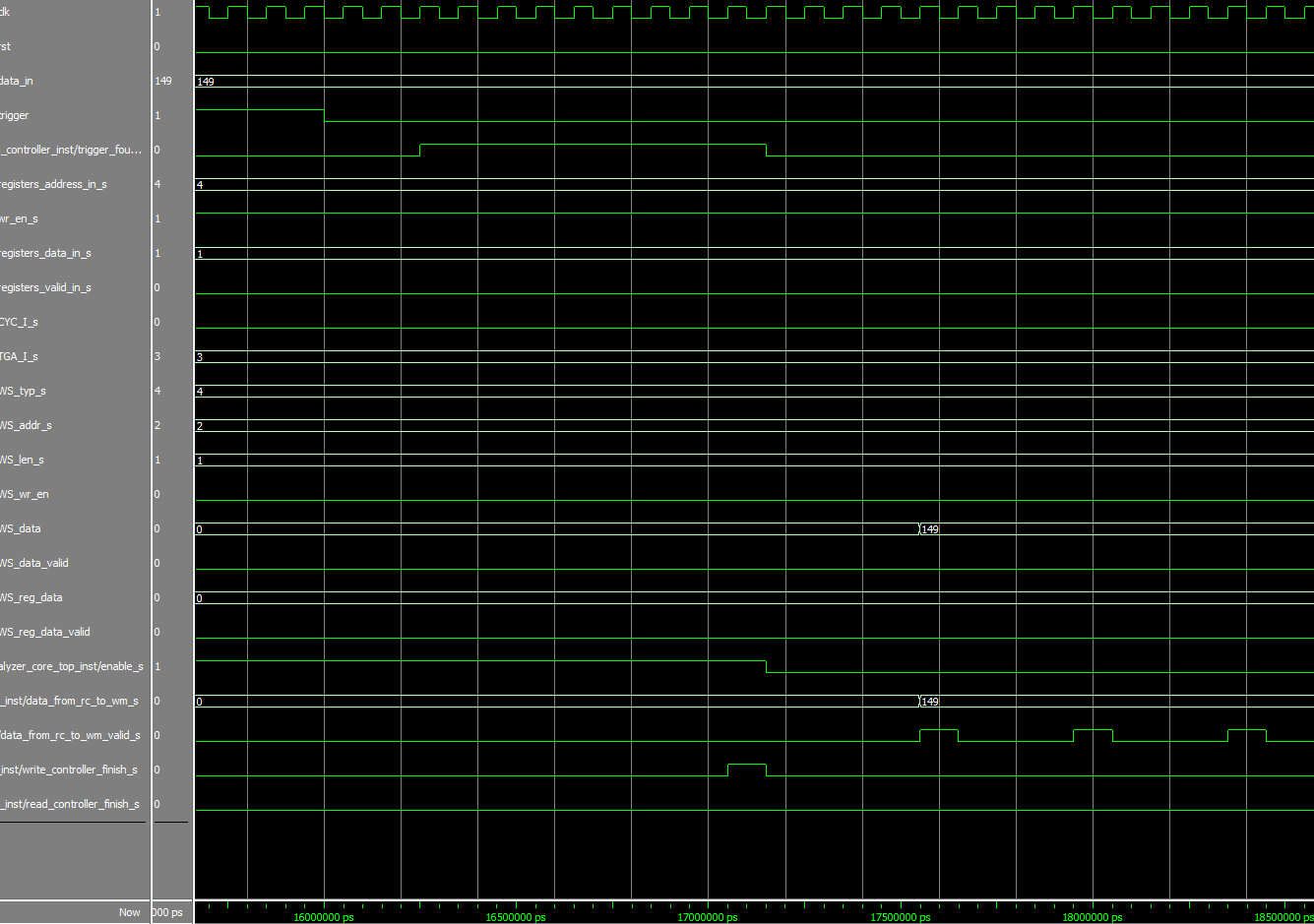


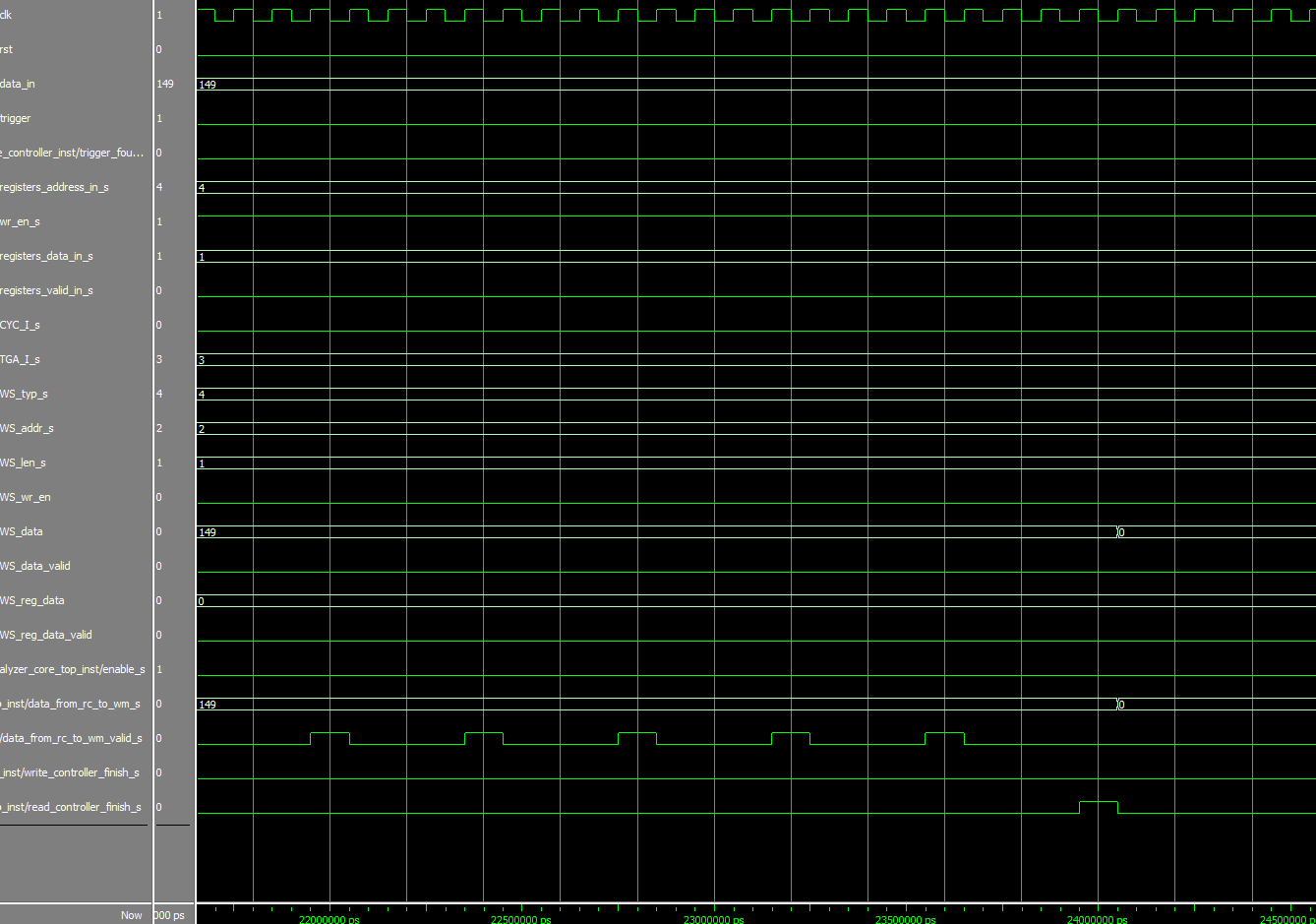












Analysis:

We change recorded depth to 4, meaning we record 2^4= 16 bits. We can see that just after the trigger is low for 3 cycles the system detect that and rise the "trigger found" signal. The system is also output 12 bits (75% of 16) before the trigger rise (first rise at 55 and output starts at 43 -> 55 – 12 = 43) and we output 16 bits after that.

After we output all the data we again enable the system by writing again to the enable register (the address and data to registers is already at the correct values) and couple of cycles after that we again detect "trigger rise" and the whole cycle start over.

TEST NUMBER 3

Generics values:

|  |  |  |
| --- | --- | --- |
| Comments | Value | Name |
|  | 1 | reset\_polarity\_g |
|  | 1 | enable\_polarity\_g |
| Each the number og addresses is RAM in | 3 | signal\_ram\_depth\_g |
| "word" width – line in the RAM | 8 | signal\_ram\_width\_g |
| The number of recorded signals is | 3 | record\_depth\_g |
| BUS width of the information entering the component | 8 | data\_width\_g |
| Word width in WB protocol | 8 | Add\_width\_g |
| Number of signals we record in each iteration | 5 | num\_of\_signals\_g |
| Output width is exactly like input width | 0 | power2\_out\_g |
| Output and input width are the same | 1 | power\_sign\_g |
| Type depth. We got 3 WM and 3 WS -> 6 total whisebone entities | 6 | Type\_d\_g |
| Length depth. | 1 | Len\_d\_g |

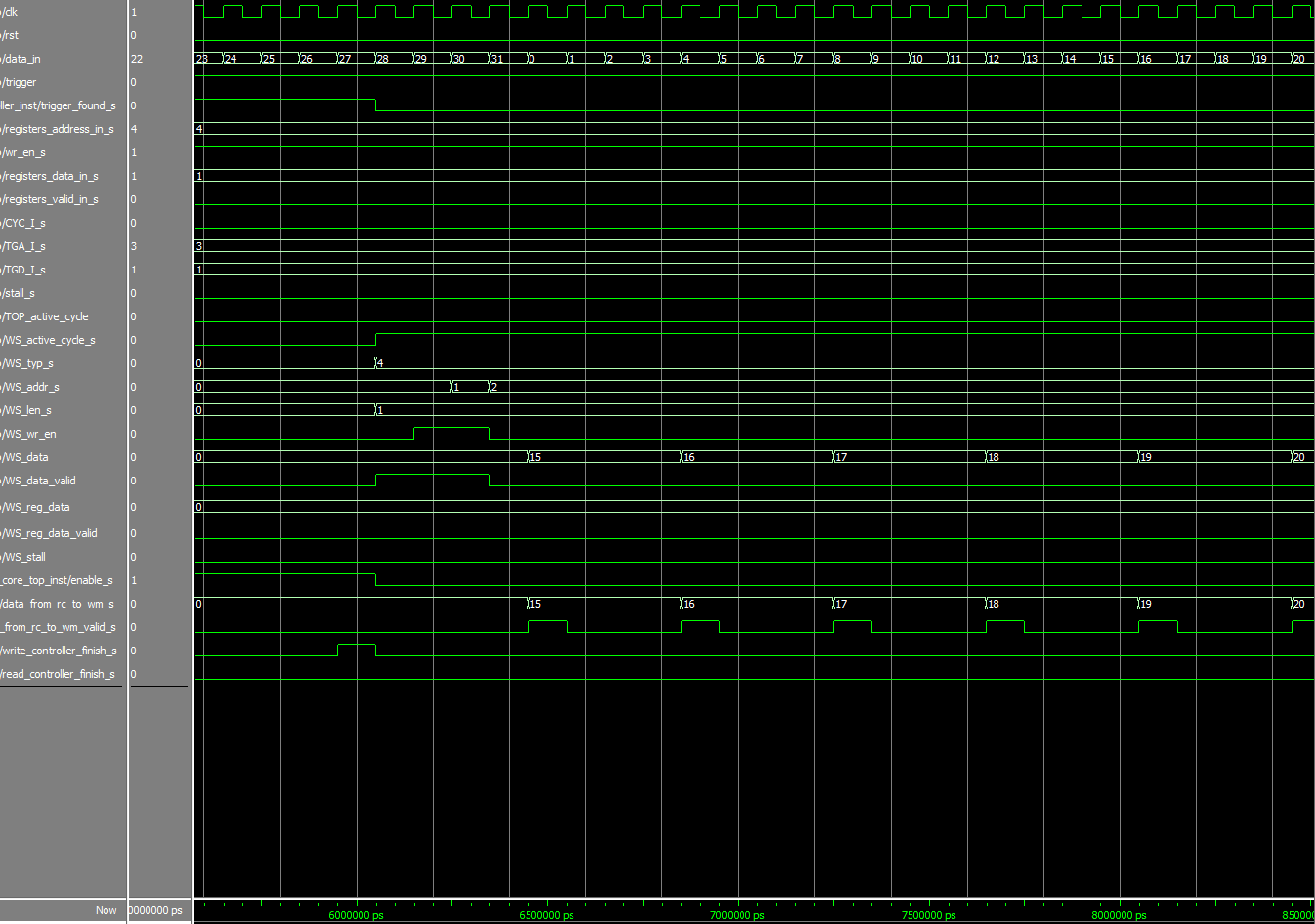
Explanation:

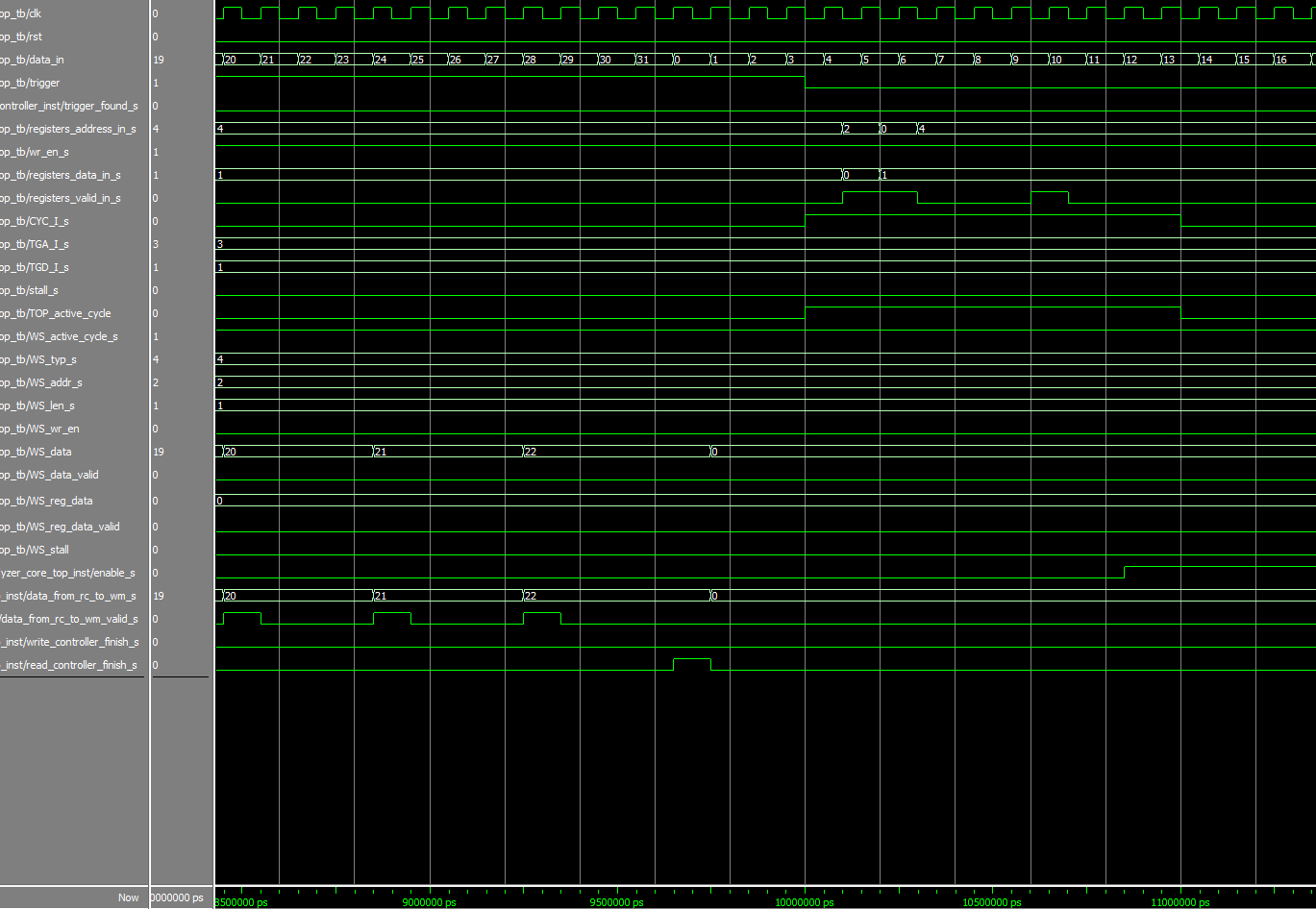
Number of signals is changed to 5, meaning our input data is between 0-32 in decimal (2^5), at first the trigger position is 100 and all the data is recorded before the trigger, and second time the position is 0 and all the data is recorded after the trigger.

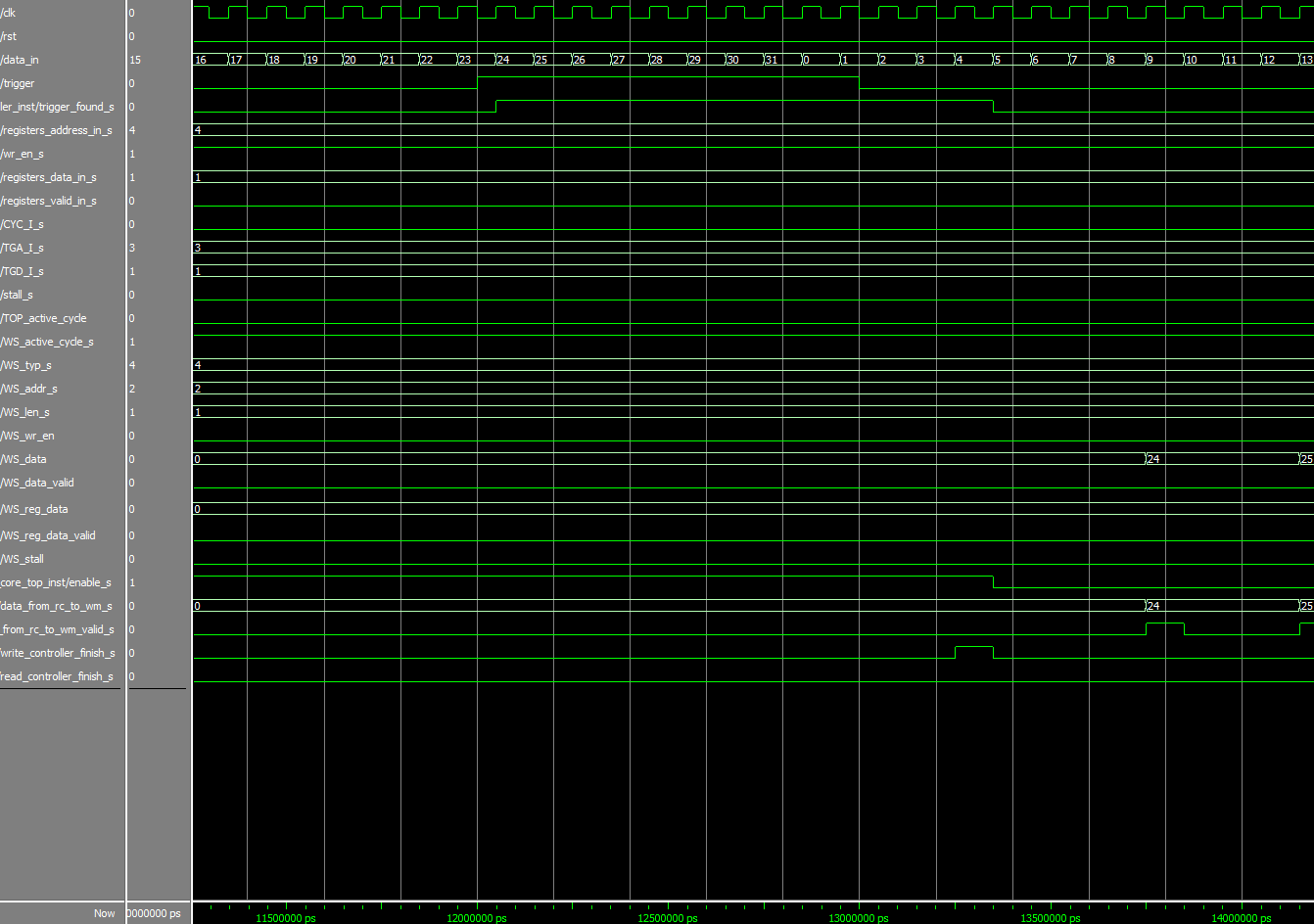
Simulation:

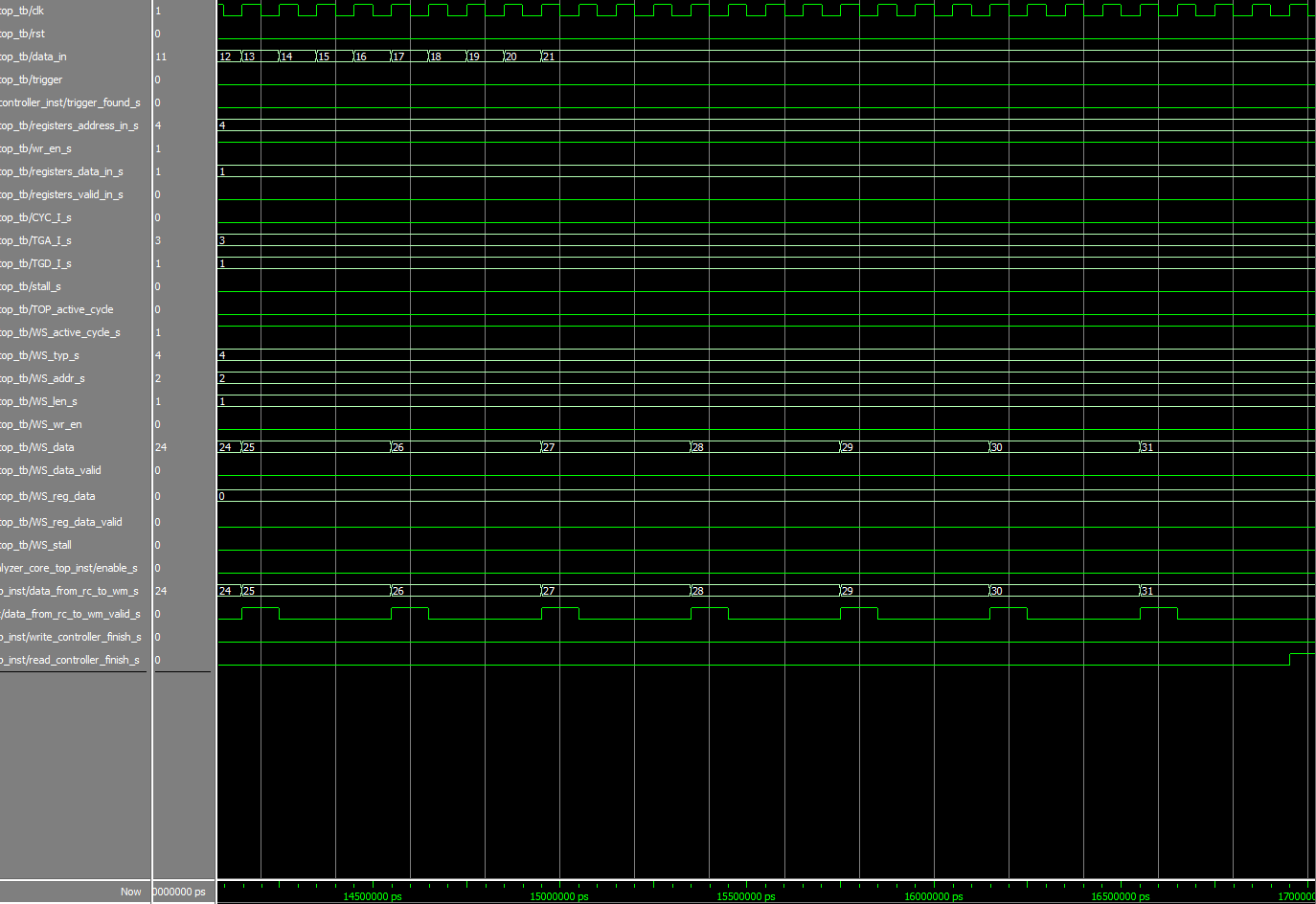


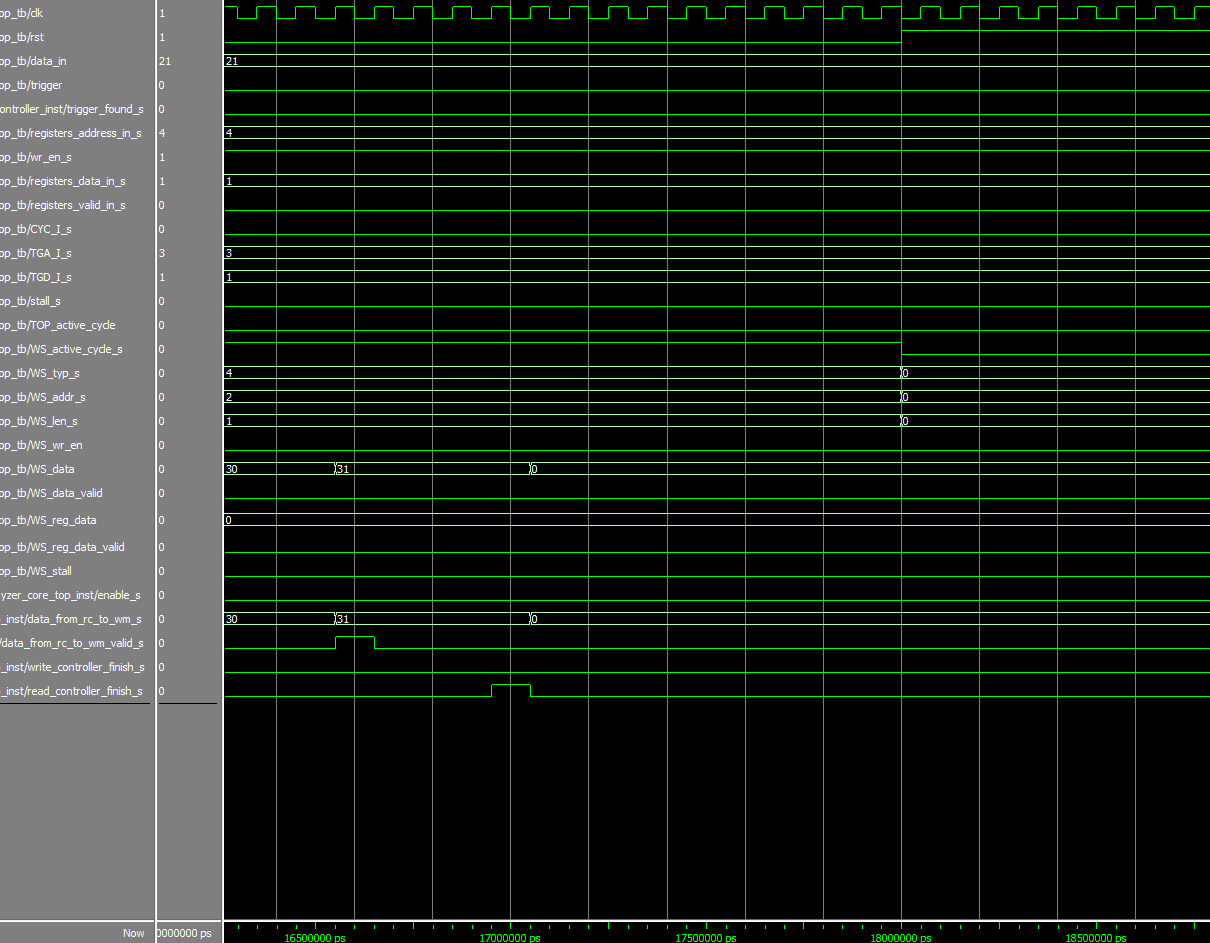












Analysis:

Number of signals is 5 -> 2^5 = 32 therefor the input data is now between 0-32, recording depth is 3 -> we record 2^3 = 8 samples of each signal.

In the first cycle position is 100 and all the data is recorded before the trigger (trigger not included), trigger found at data 23 so the output data is 15-22.

In the second cycle we change the position to 0 and all the data is recorded after the trigger (trigger included), trigger found at data 24 so the output is 24-31.

TEST NUMBER 4

Generics values:

|  |  |  |
| --- | --- | --- |
| Comments | Value | Name |
|  | 1 | reset\_polarity\_g |
|  | 1 | enable\_polarity\_g |
| Each the number og addresses is RAM in | 3 | signal\_ram\_depth\_g |
| "word" width – line in the RAM | 8 | signal\_ram\_width\_g |
| The number of recorded signals is | 4 | record\_depth\_g |
| BUS width of the information entering the component | 8 | data\_width\_g |
| Word width in WB protocol | 8 | Add\_width\_g |
| Number of signals we record in each iteration | 7 | num\_of\_signals\_g |
| Output width is exactly like input width | 0 | power2\_out\_g |
| Output and input width are the same | 1 | power\_sign\_g |
| Type depth. We got 3 WM and 3 WS -> 6 total whisebone entities | 6 | Type\_d\_g |
| Length depth. | 1 | Len\_d\_g |

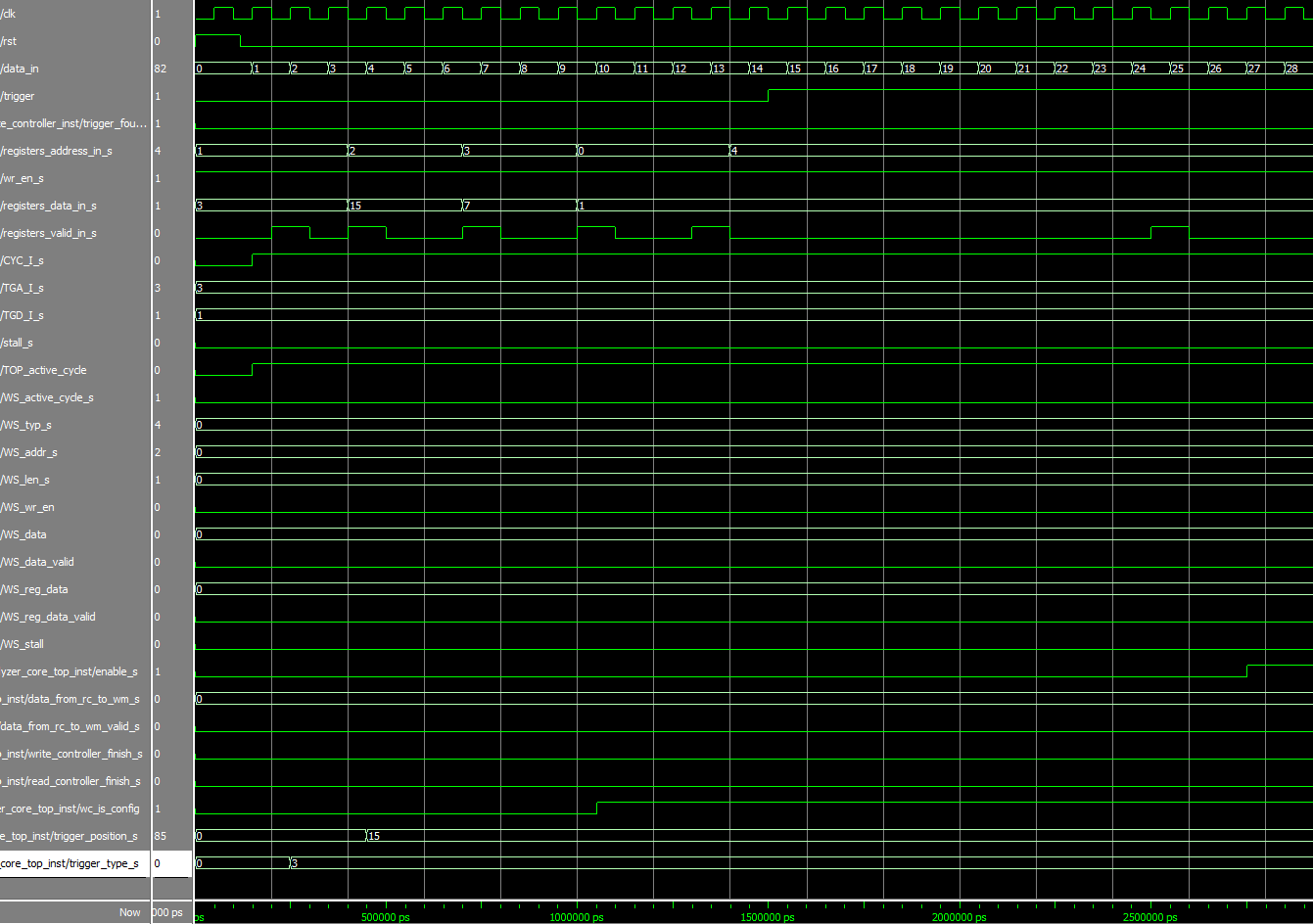
Explanation:

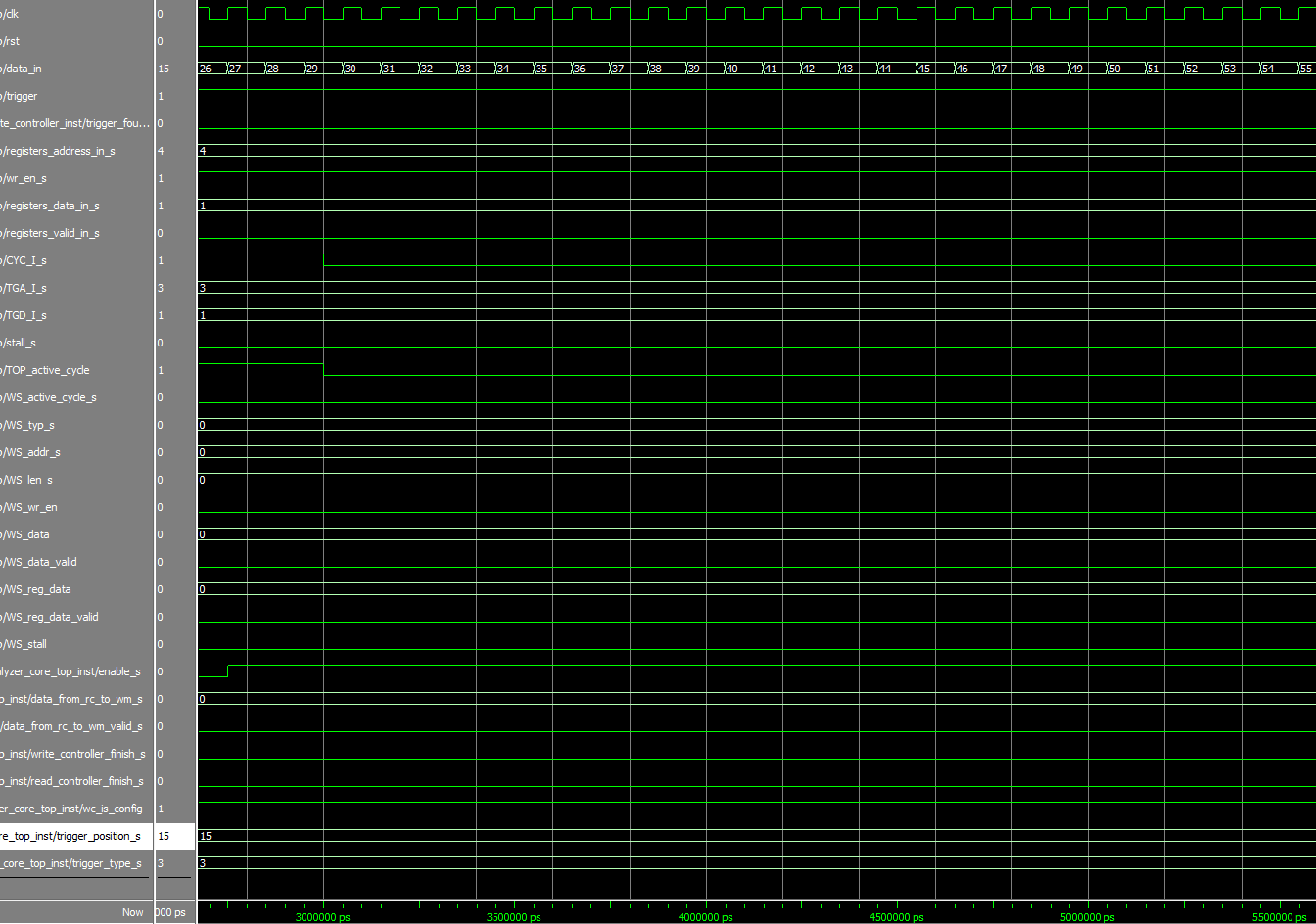
In this simulation we check two trigger positions that are not complete, 15% and 85%.

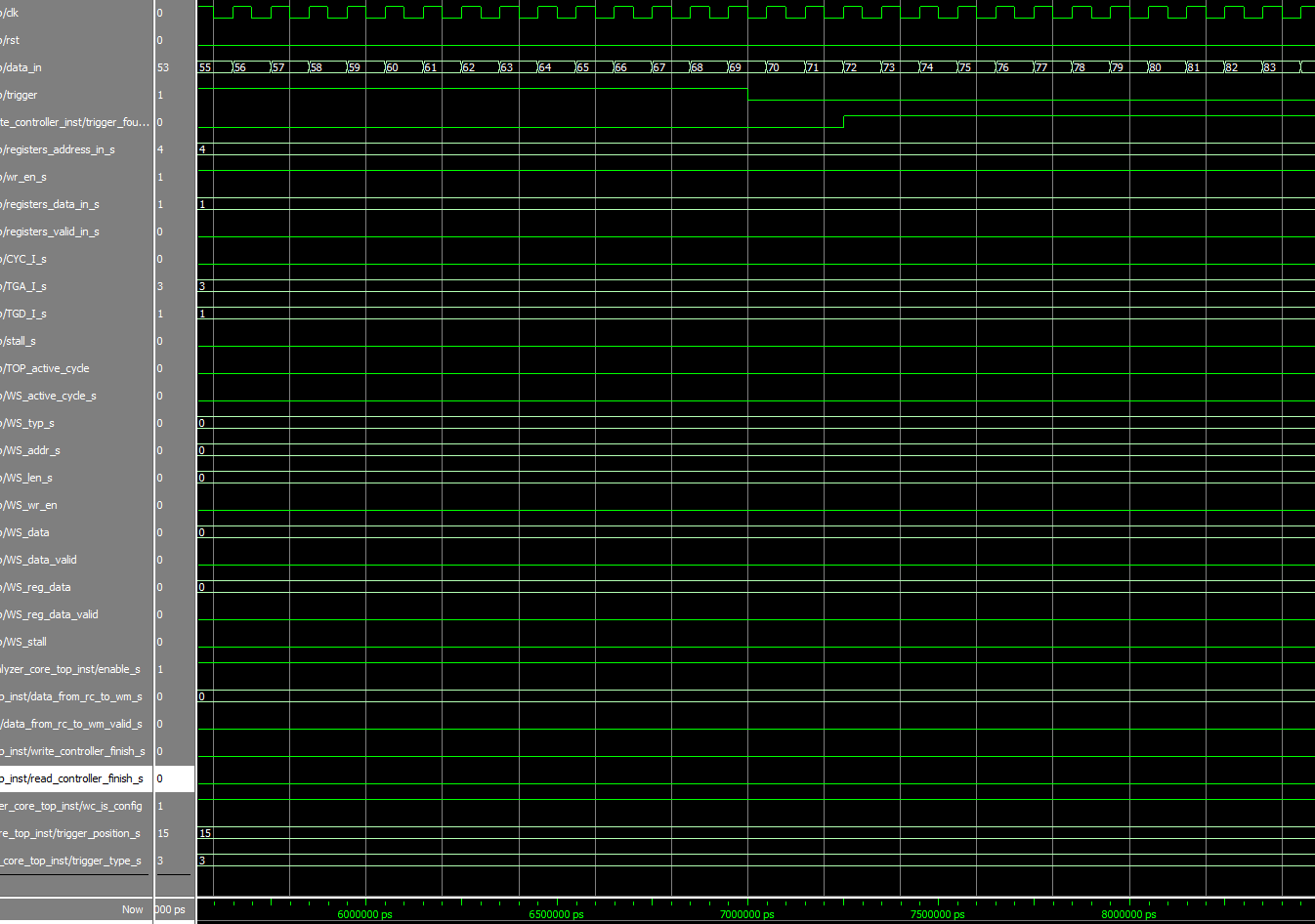
We also change the trigger type, from 'zeroes' (3 low in a row) in the first one to rise in the second.

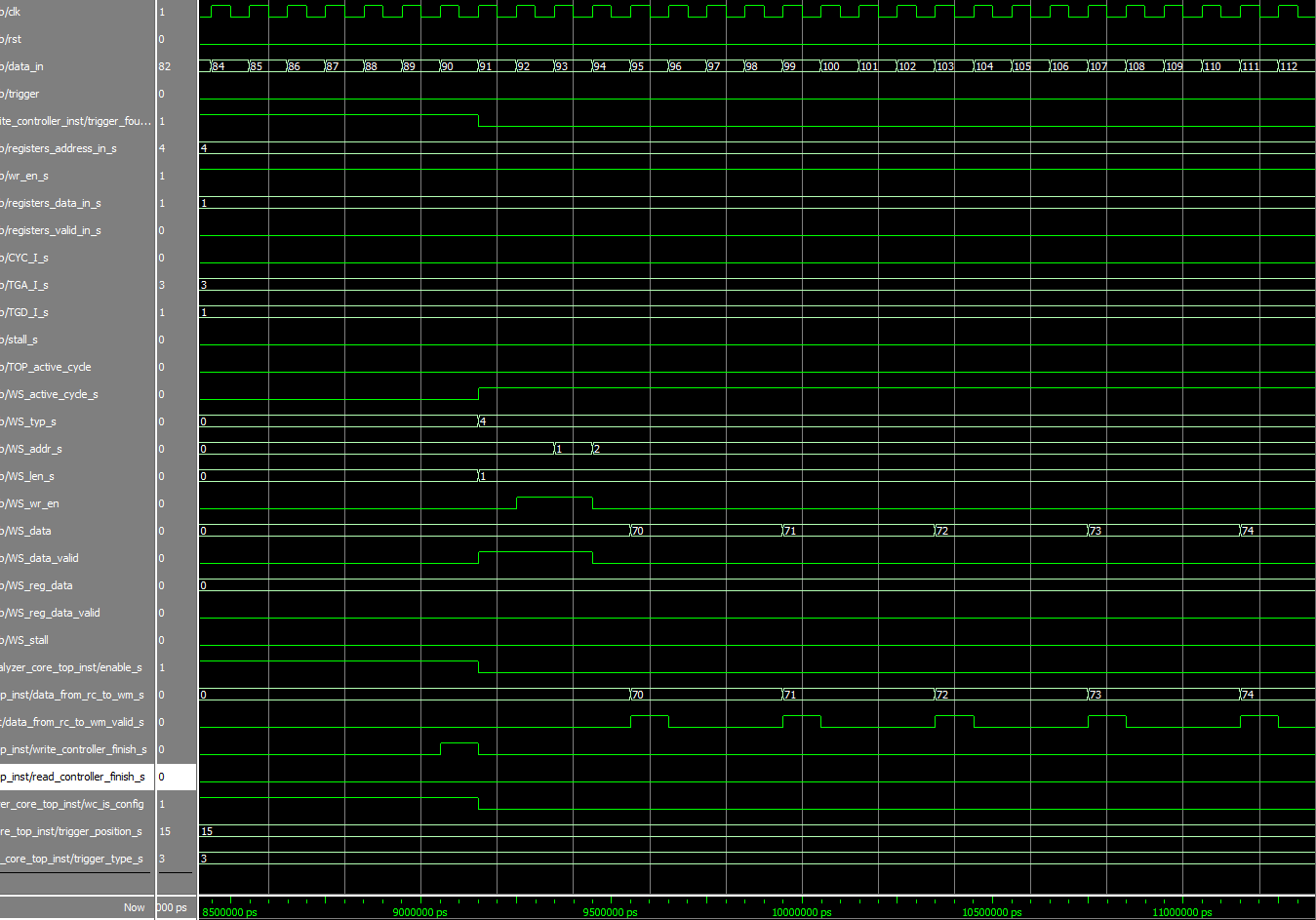
In the end we rise the RESET signal to show that the system is respond to that signal.

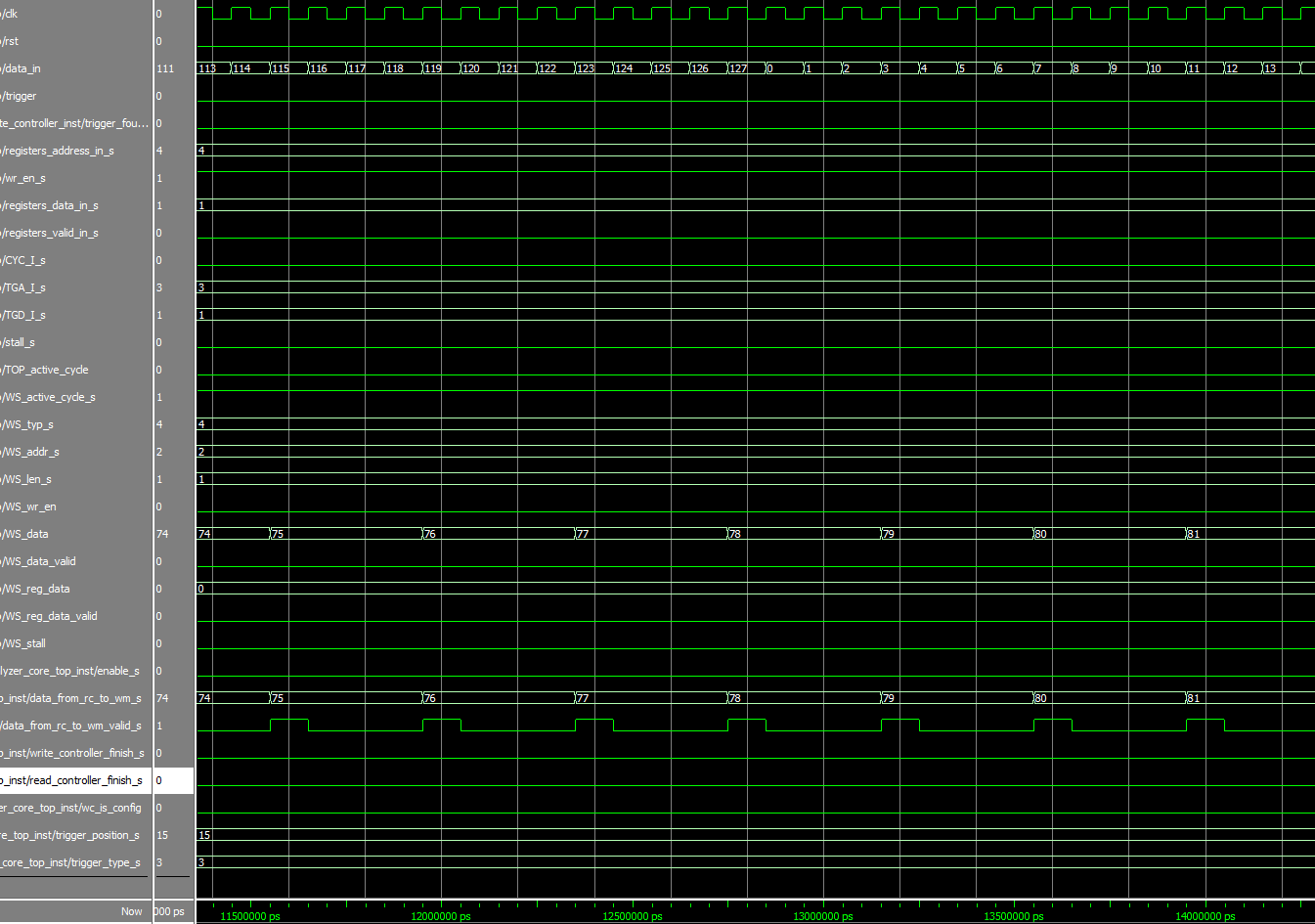
Simulation:

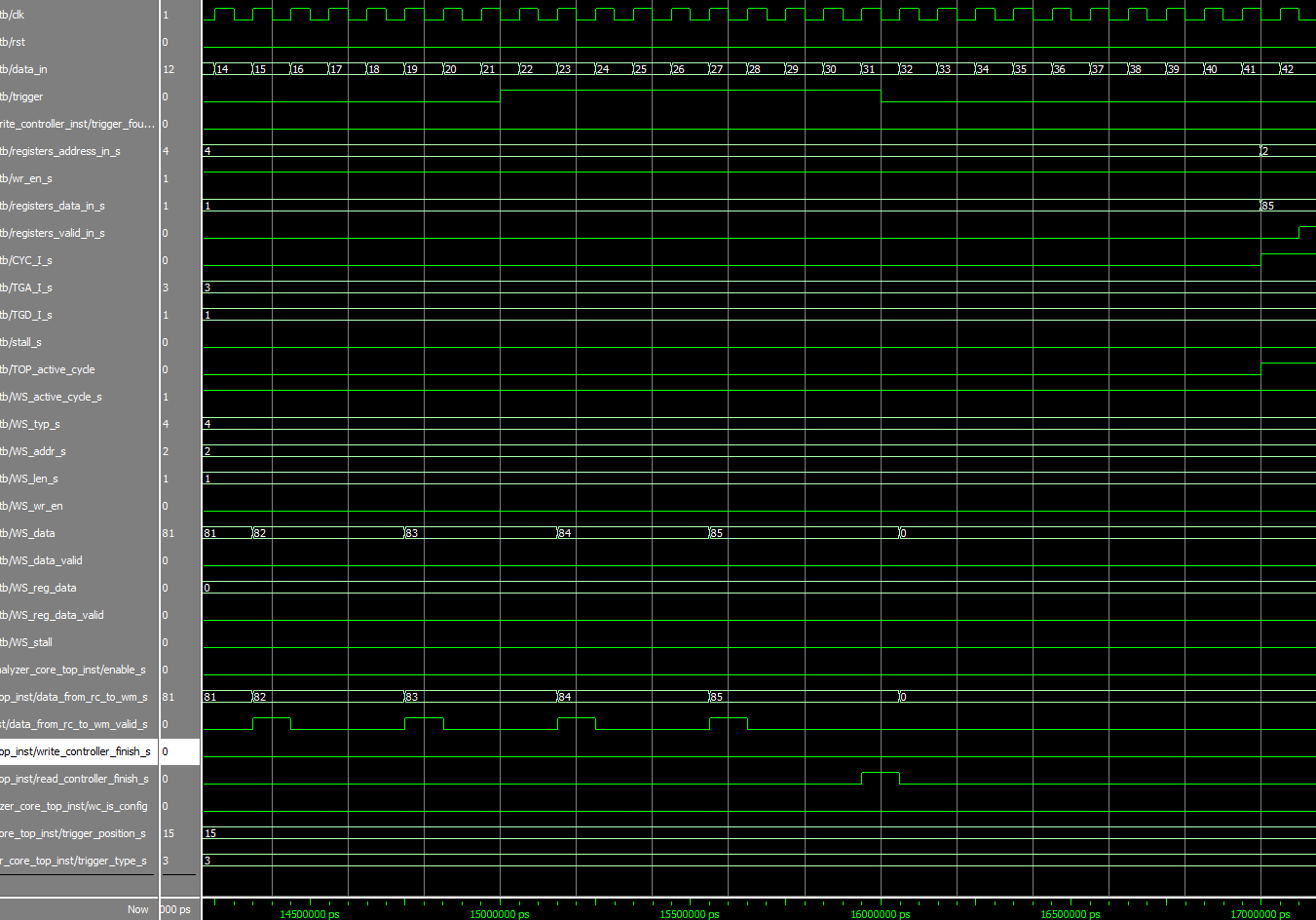


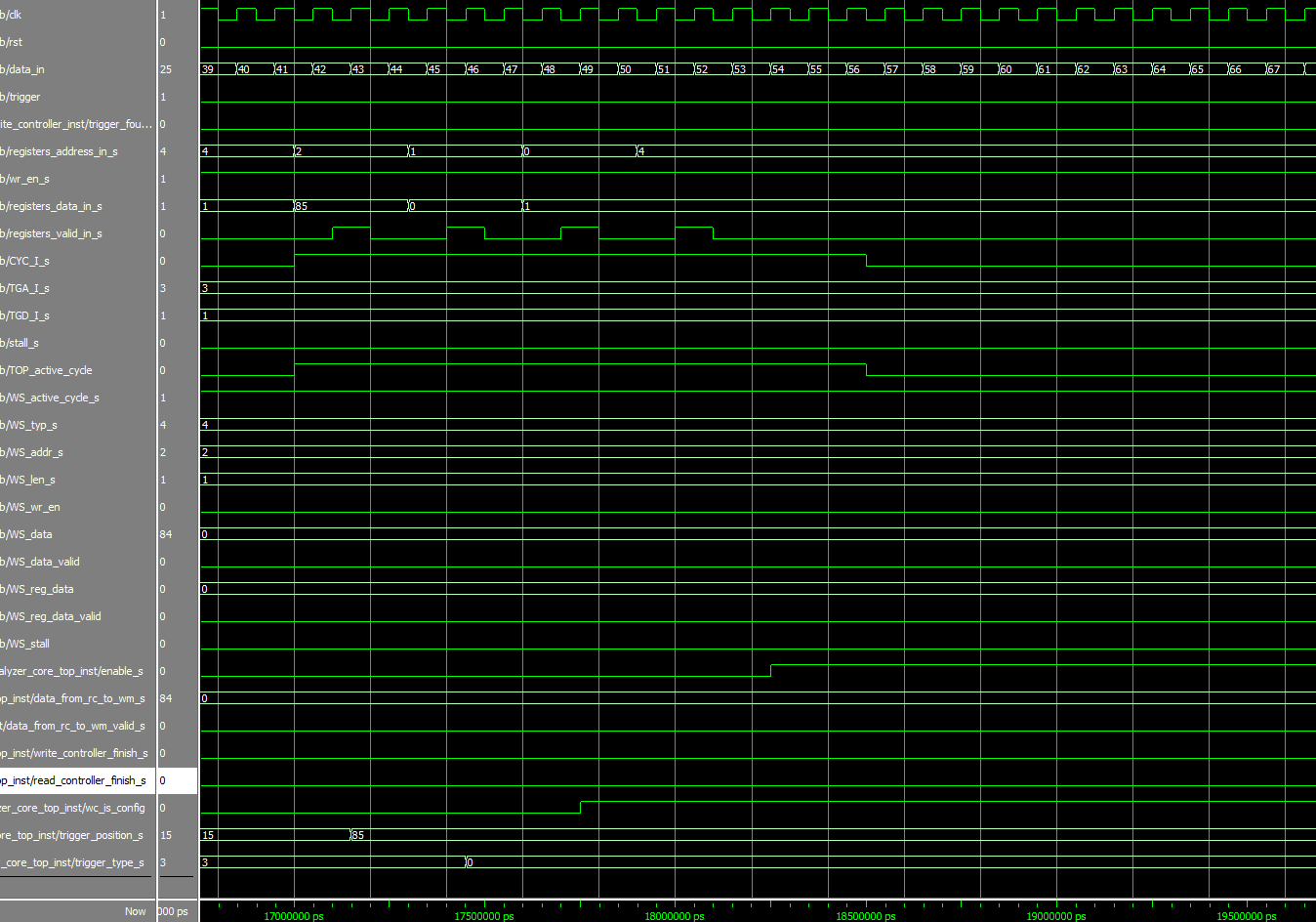


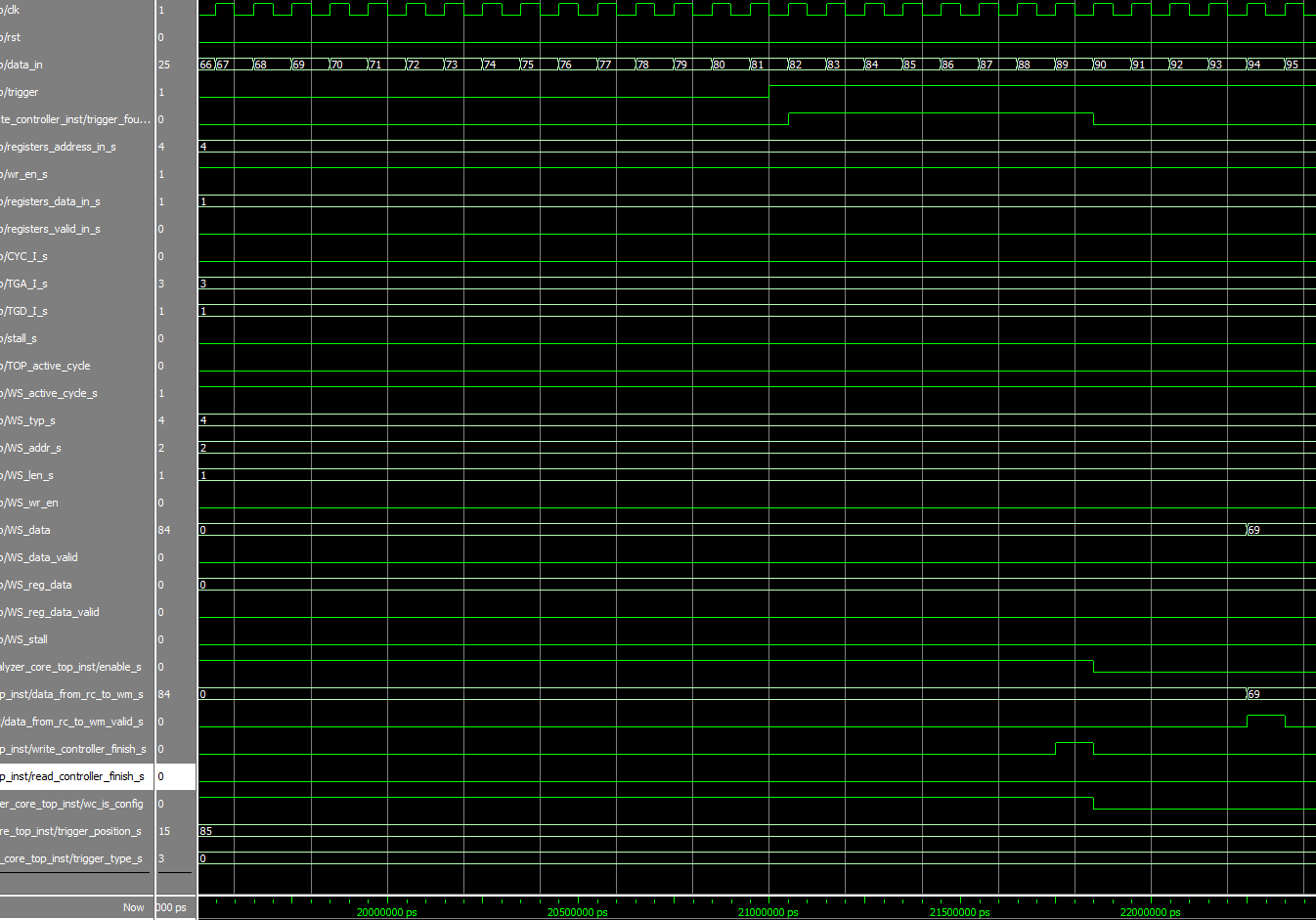


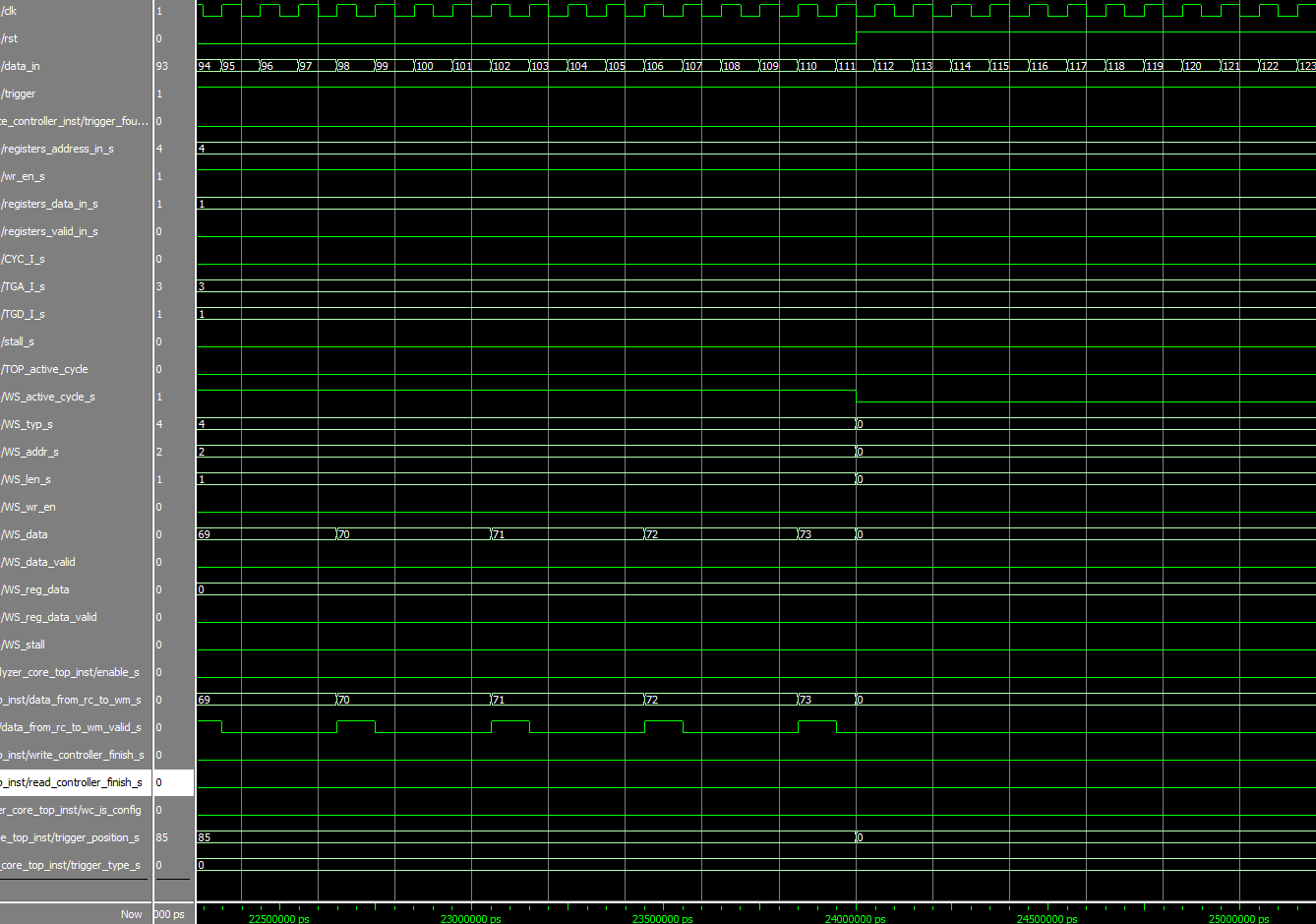












Analysis:

Like all the other simulations, we can see that the system is getting the initial configurations at the beginning. We record 7 signals and 2^4 = 16 samples from each one.

In the first cycle position is 'zeroes', and we seek for three low trigger at a row, we found that when the data is 72, and since the position is 15% we save 16\*(15/100) = 2.4 = 2 samples before the trigger ( 70 to 85).

In the second cycle we change the position to 85%, meaning that we save 16\* (85/100) = 13.6 = 13 samples before the trigger ( 69 to 84 ). We rise the RESET signal after 5 outputs to see that this signal is functional but the output at the start is according our expectation.