Description:

1. The entity gets the incoming data and trigger signals from the signal generator, calculating the address in the RAM that the data will be saved in, and sending the data and address to the RAM.
2. According the configurations that are saved in the core registers, detecting trigger rise and calculating the start address of the outputting data, and send it to the read controller.

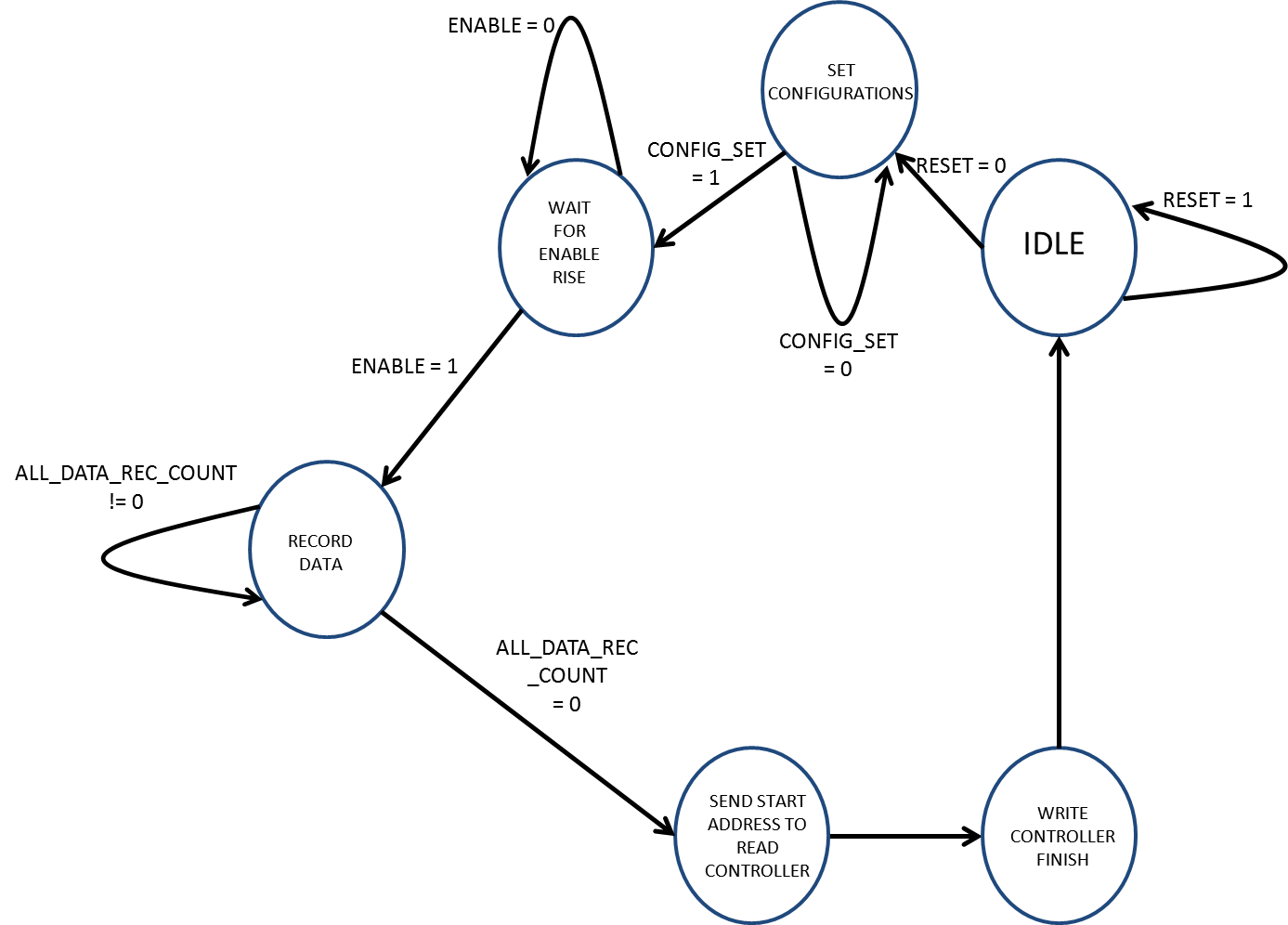
Generic table

|  |  |  |
| --- | --- | --- |
| Name | Width | Description |
| reset\_polarity\_g | 1 | '1' reset active high, '0' active low |
| enable\_polarity\_g | 1 | '1' the entity is active high, '0' entity is active low |
| signal\_ram\_depth\_g | 3 | depth of single RAM is 2^signal\_ram\_depth\_g |
| signal\_ram\_width\_g | 8 | width of single RAM |
| record\_depth\_g | 4 | number of bits that are recorded from each signal is 2^record\_depth\_g |
| data\_width\_g | 8 | defines the width of the data lines of the system |
| Add\_width\_g | 8 | width of addr word in the RAM |
| num\_of\_signals\_g | 8 | number of signals that will be recorded simultaneously |

Signals table

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Direction | Width | Description |
| clk | In | 1 | System clock |
| Reset | In | 1 | System reset |
| Enable | In | 1 | Enabling the entity. if (enable = enable\_polarity\_g) -> start working, else-> do nothing |
| trigger\_position\_in | In | 7 | The percentage of the data that is recorded before trigger rise |
| trigger\_type\_in | In | 7 | Type of trigger |
| config\_are\_set | In | 1 | '1'-> configurations from registers are ready to be read (trigger position + type). '0'-> config are not ready |
| data\_in | In | num\_of\_signals\_g | Data in from signal generator |
| Trigger | In | 1 | Trigger signal from signal generator |
| data\_out\_of\_wc | Out | num\_of\_signals\_g | sending the data to be saved in the RAM |
| addr\_out\_to\_RAM | Out | Add\_width\_g | the address in the RAM to save the data |
| write\_controller\_finish | Out | 1 | '1' ->WC has finish working and saving all the relevant data (RC will start work), '0' ->WC is still working |
| start\_addr\_out | Out | Add\_width\_g | The start address of the data that we need to send out to the user |
| din\_valid | Out | 1 | Data to RAM valid |

Write controller state machine



Output table

