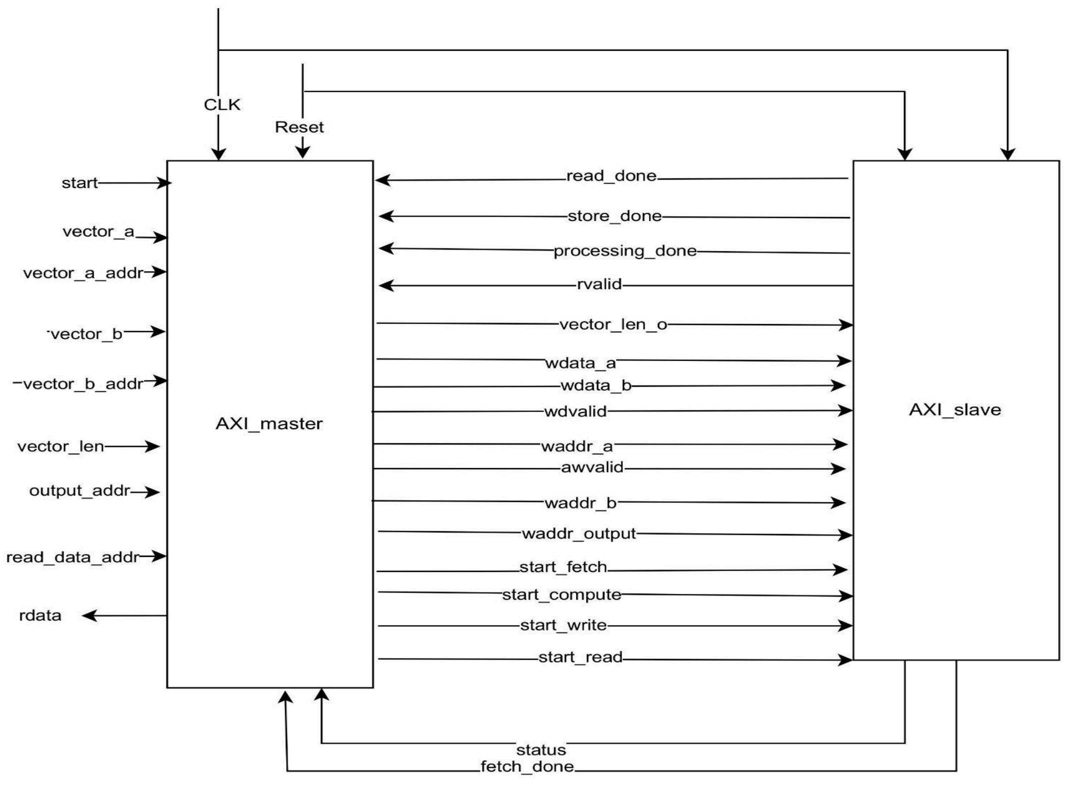
Signals	
Clk	Clock signal
Reset	Reset signal for the
Start	Start signal to start the AXI protocol
vector_a	Input A array
vector_b	Input B array
vector_a_addr	Input A array addresses this is the location where A array is stored.
vector_b_addr	Input B array addresses this is the location where B array is stored.
vector_len	This is a signal that tells how many iterations in the loop.
output_addr	This is address signal which tells where the result will be stored
read_data_addr	This is the address signal where the output data is read from the memory.
read_data	This is the output data read from the memory
status	This is signal thats tells whether the AXI slave is busy or ready
processing_done	This signal is high when the multiplication operation is done
store_done	This signal is high when the result is stored in the read_data_addr
read_done	This signal is high when data is read from the memory address
rvalid	Read valid signal, indicates when read_data is available for the AXI master to use.
wdata_a	Data value of vector_a written into memory
wdata_b	Data value of vector_b written into memory
waddr_a	Address where the vector_a is stored
waddr_b	Address where the vector_b is stored
waddr_output	Address where the computation result should be written
vector_len_o	Number of elements to be used in computation
Wdvalid	Write data valid: When HIGH (1),wdata_a and wdata_b contain valid data.
awvalid	Address write valid: When HIGH (1),wdata_a and wdata_b contain valid Address
rdata	Data read from memory and available for the user.

Signals	
start_fetch	HIGH (1) when the AXI master is fetching data from memory.
start_compute	HIGH (1) when the AXI master is starting computation.
start_write	HIGH (1) when the AXI master is writing results to memory.
start_read	HIGH (1) when the AXI master is reading the computed results from memory.

ARCHITECTURE:



How different module works:

AXI_Master:

- The AXI master acts as the FSM for the protocol it takes in the input addresses,vector_len and input data and stores them until the fetch state is reached.
- When the fetch state is reached the master sends the values,vector_len and addresses to the slave.
- There are 6 states in the master finite state machine.
- In the IDLE state the values are reset to zero.
- And when the start signal is high the IDLE state is moved into the next state that is START_STATE.
- And when the START_STATE is reached all the addresses are copied into the registers and waits until the start fetch is called.
- And the START_STATE automatically moves into the FETCH_STATE.
- And in the FETCH_STATE the start_fetch signal is assigned high and the slave takes the start_fetch value.
- And from the FETCH_STATE the FSM moves into COMPUTE_STATE.
- Where in the COMPUTE_STATE the start_compute signal is assigned high.
- And if the processing_done signal is high the FSM moves to WRITE_STATE else it will remain in the same state.
- And when the FSM is in the WRITE_STATE start_write signal is asserted high.
- If the store_done signal is high then the FSM moves to the READ_STATE else it remains in the same state.
- And in the READ_STATE if the rvalid signal is high, the read_data value is read into the rdata and the signal is outputted.
- And if the read_done signal is high then the READ_STATE is moved into the IDLE state.

AXI_Slave:

- The AXI_slave has 3 memory locations storage_a to store the wdata_a input ,storage_b to store the wdata_b input and storage_out to store the result after the computation.
- And the slave takes the outputs from the AXI_master and if the start_fetch is high then the wdata_a, wdata_b is stored into the addresses sequentially the counter counts the number of times the values are stored.
- And the stored values are then Dot produced with a counter number of iterations if the start_compute signal is high.
- And from there the result is stored into the storage_out memory.
- And if the start_read is high then the result is read into the read_data.

TOP_MODULE:

- The input values and output values are given into the module .
- And the modules AXI_Master and the AXI_Slave are connected .

- The Master contains the FSM for the AXI ,the values are read into the slave and computation happens in the slave and result from the slave is send to the master from there it is read out.