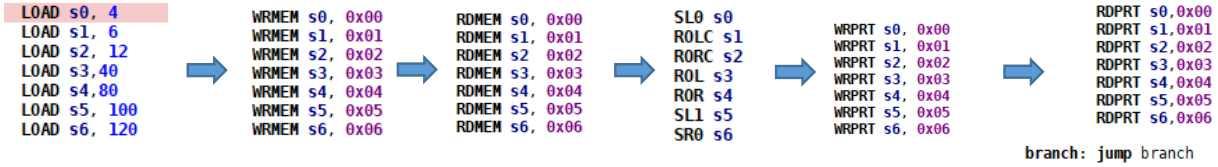
İLAYDA KÖSEOĞLU
040180742

2. WRITING TO AND READING FROM MEMORY

2.1 IMPLEMENTATION ON FIDEX

1. The registers were loaded with arbitrary values.
2. The memory was written to addresses.
3. The memory was read from those addresses.
4. Some arbitrary operations were applied to each register.
5. The values were written to the output ports.
6. The outputs were read from those ports.

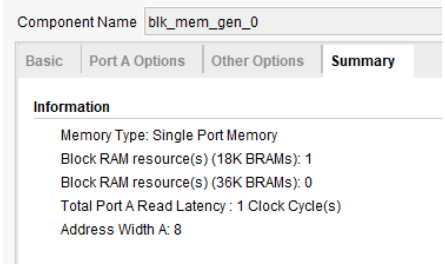
7. A jump branch was declared to see the outputs infinitely on the waveform of VIVADO.



2.2 GENERATING A WAVEFORM ON VIVADO

1. BRAM0.vhd file was generated on FIDEX.
2. A new VIVADO project was created and BRAM0, kcpsm6, sim_top and top files (.vhd) were added to the project.

3. A block memory was generated with 8 bit write width & read width and 256 bit write depth & read depth. Primitives output register was removed to reduce latency to 1 clock cycle. The summary of this block memory is as follows:



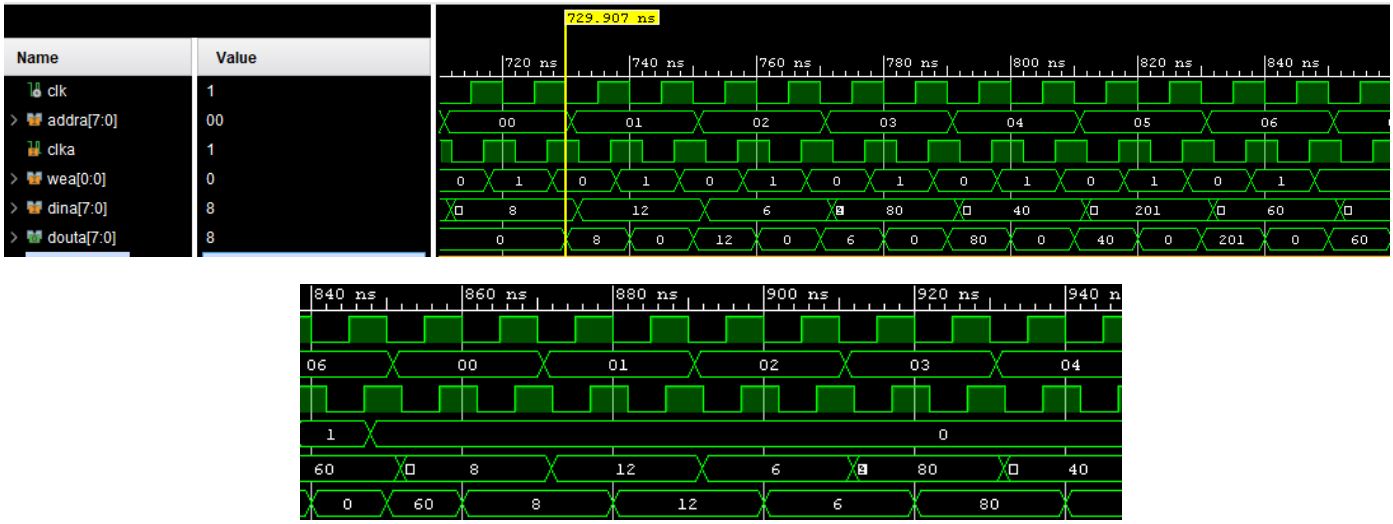
4. The block memory was declared as a component at the top module. Then it was port mapped by using appropriate signals. The relevant signals were changed in kcpsm generic map accordingly.

```
component blk_mem_gen_0 IS
  PORT (
    clka : IN STD_LOGIC;
    wea : IN STD_LOGIC;
    addra : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
    dina : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
    douta : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
  );
end component;
```

```
blk_gen: blk_mem_gen_0
          port map (
            clka=>clk,
            wea =>write_strobe,
            addra =>port_id_2,
            dina =>sig_out_port_2,
            douta =>sig_in_port
          );

end Behavioral;
```

5. The circuit was synthesized successfully and then simulated. The simulation



6. The elaborated design of the circuit is as follows: