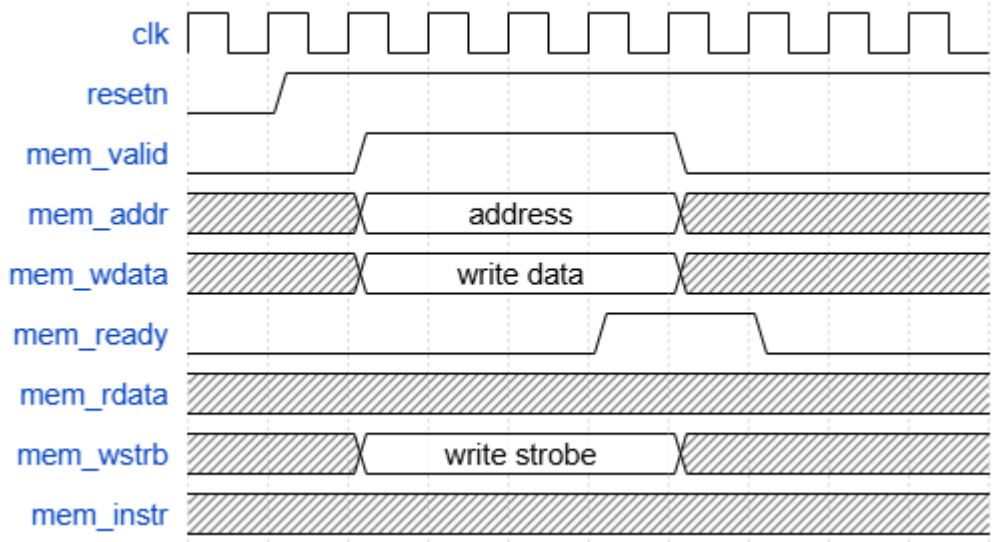
PICORV32 SRAM SPECIFICATION

Port list	Direction	Description
clk	input	IP clock (125 MHZ)
resetn	input	synchronous active low reset
mem_valid	input	0: no memory access request 1: request for accessing memory, mem_addr, mem_wdata, mem_wstrb, mem_instr are valid when mem_valid = 1
mem_ready	output	In read access 0: output rdata is not valid 1: output rdata is valid In write access 0: write not complete 1: write complete
mem_addr	input	16 bit memory address
mem_wdata	input	32 bit write data from CPU for write access
mem_rdata	output	32 bit read data from SRAM for read access
mem_wstrb	input	4’b0000: read access Another is write access. This signal has 4 bits to mask 4 bytes, LSB

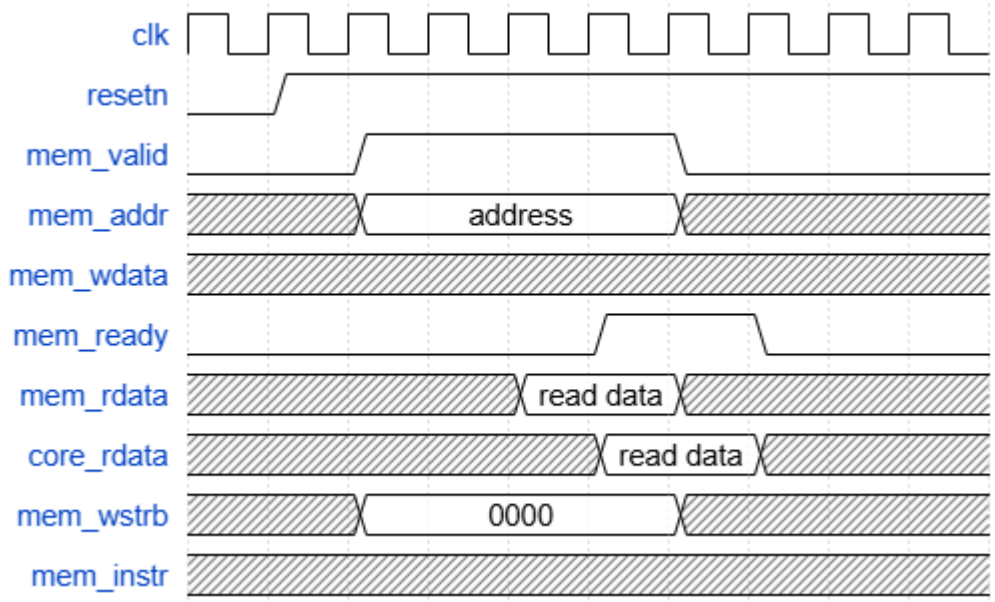
		bit is LSB byte, MSB bit is MSB byte. If bit is 0, meaning that this byte in memory does not change. If bit is 1, meaning that this byte in memory is written with corresponding byte in mem_wdata.
mem_instr	input	1: access for instruction 0: access for data

WRITE ACCESS



Explanation: When master or CPU requests write access, CPU drives mem_valid to 1, drives mem_addr, mem_wdata and mem_wstrb to a valid value. And then the CPU keeps these values until mem_ready is asserted to 1, after that CPU stops requesting and drives mem_valid to 0. In this system, because we know and define ourselves where to store instructions and where to store data in this unified memory, we don't use the mem_instr signal.

READ ACCESS



Explanation:
When master or CPU requests read access, CPU drives mem_valid to 1, drives mem_addr to a valid value and drives mem_wstrb to 4'b0. And then the CPU keeps these values until mem_ready is asserted to 1, after that CPU stops requesting and drives mem_valid to 0, CPU reads data from core_rdata. In this system, there is a register core_rdata that stores stable data of memory and UART read access, the data of mem_rdata must go through this core_rdata register before going to the CPU, so the read data is delayed 1 cycle and mem_ready must delay 1 cycle and is asserted to 1 when core_rdata is valid, not mem_rdata. core_rdata register is defined in picorv32_top module.

LITTLE ENDIAN

0x34	0x12	0xCD	0xAB	**DATA**
8	**9**	**10**	**11**	**ADDRESS**

Explanation: When the CPU writes 1 word (a chunk of 4 bytes) 32'hABCD_1234 at address 8, LSB byte 0x34 must be stored at lowest address (that is 8), MSB byte 0xAB must be stored at highest address (that is 11) as shown in the above figure. Meaning that when the CPU later executes load byte instruction at address 10, read data must be 0xCD.