Introduction to Embedded Systems

EHB326E

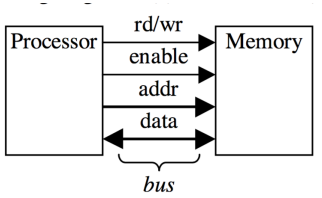
Lectures

Prof. Dr. Müştak E. Yalçın

Istanbul Technical University

mustak.yalcin@itu.edu.tr

Memory Interface



Sources:

- LogiCORE IP Block Memory Generator v7.3, Product Guide
  `https://www.xilinx.com/support/documentation/ip_documentation/blk_mem_gen/v7_3/pg058-blk-mem-gen.pdf`

Memory

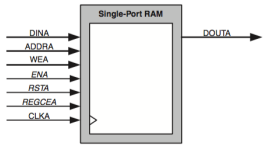


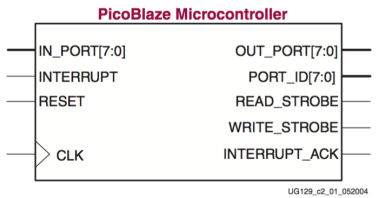
Figure 3-3: **Single-port RAM**

Table 2-21: **Core Signal Pinout**

Name	Direction	Description
CLKA	Input	**Port A Clock**: Port A operations are synchronous to this clock. For synchronous operation, this must be driven by the same signal as CLKB.
ADDRA	Input	**Port A Address**: Addresses the memory space for port A Read and Write operations. Available in all configurations.
DINA	Input	**Port A Data Input**: Data input to be written into the memory via port A. Available in all RAM configurations.
DOUTA	Output	**Port A Data Output**: Data output from Read operations via port A. Available in all configurations except Simple Dual-port RAM.
ENA	Input	**Port A Clock Enable**: Enables Read, Write, and reset operations via port A. Optional in all configurations.
WEA	Input	**Port A Write Enable**: Enables Write operations via port A. Available in all RAM configurations.
RSTA	Input	**Port A Set/Reset**: Resets the Port A memory output latch or output register. Optional in all configurations.
REGCEA	Input	**Port A Register Enable**: Enables the last output register of port A. Optional in all configurations with port A output registers.

Memory

Connecting the processor with the memory ?



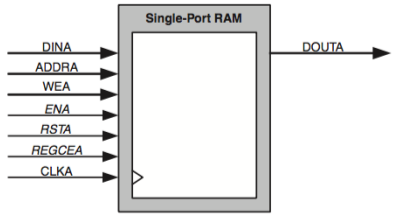


Figure 3-3: **Single-port RAM**

Memory: Write

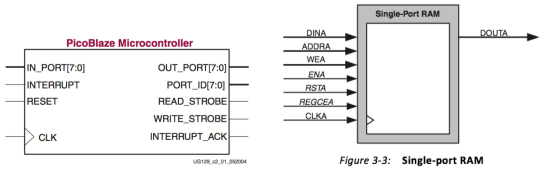


Figure 3-3: **Single-port RAM**

Timing diagram:

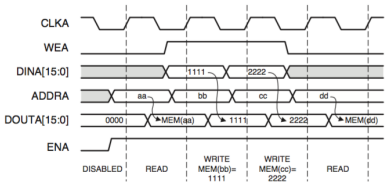
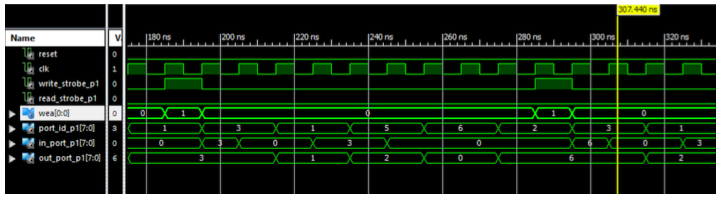


Figure 3-9: **Write First Mode Example**

PS: ENA is always Enable !

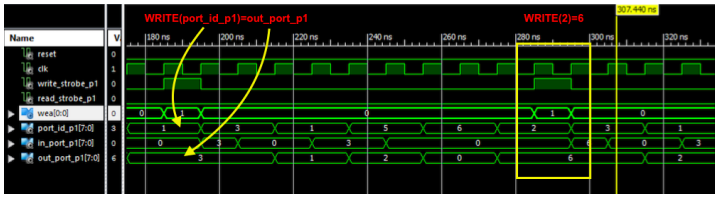
Memory: Write

```
init:
        AND RAM_data, 00
        AND BRAM_ADDR, 00
        AND SRAM_ADDR, 10
WR_BRAM:
        OUTPUT RAM_data,(BRAM_ADDR)
        ADD RAM_data, 03
        ADD BRAM_ADDR, 01
        COMPARE BRAM_ADDR, 05
        JUMP NZ, WR_BRAM
```



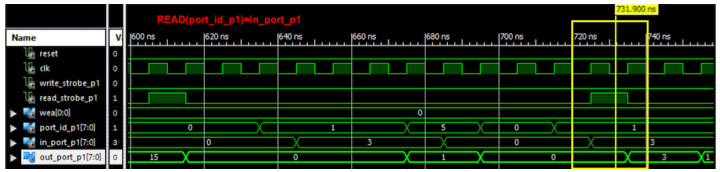
Memory : Write

```
init:
        AND RAM_data, 00
        AND BRAM_ADDR, 00
        AND SRAM_ADDR, 10
WR_BRAM:
        OUTPUT RAM_data,(BRAM_ADDR)
        ADD RAM_data, 03
        ADD BRAM_ADDR, 01
        COMPARE BRAM_ADDR, 05
        JUMP NZ, WR_BRAM
```



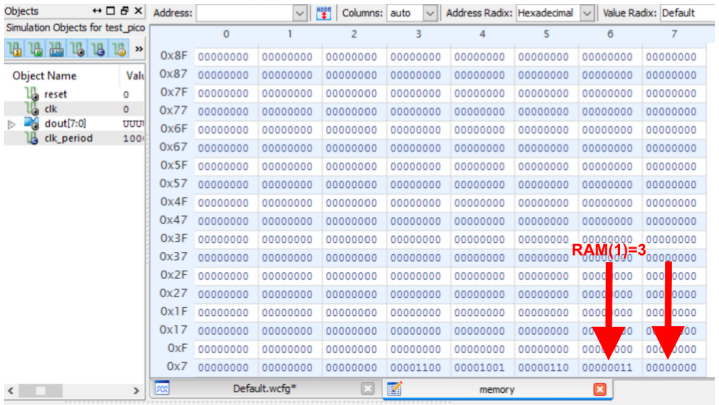
Memory : Read

```
        SUB BRAM_ADDR, 05
READ_BRAM:
        INPUT RAM_data,(BRAM_ADDR)
        STORE RAM_data,(SRAM_ADDR)
        ADD BRAM_ADDR, 01
        ADD SRAM_ADDR, 01
        COMPARE BRAM_ADDR, 05
        JUMP NZ, READ_BRAM
```



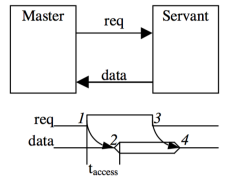
Memory : Read

```
        SUB BRAM_ADDR, 05
READ_BRAM:
        INPUT RAM_data,(BRAM_ADDR)
        STORE RAM_data,(SRAM_ADDR) ; store to
Scratchpad RAM
        ADD BRAM_ADDR, 01
        ADD SRAM_ADDR, 01
        COMPARE BRAM_ADDR, 05
        JUMP NZ, READ_BRAM
```



Memory with control method

8-bit data transfer between two processors with following protocol:



1. Master asserts *req* to receive data
2. Servant puts data on bus **within time** t_{access}
3. Master receives data and deasserts *req*
4. Servant ready for next request