DDR2 SDRAM MEMORY CONTROLLER

ARUN CC

INTRODUCTION

- High Speed Memory Controller Design Considerations

- The Signal integrity is an challenging issue in High speed design.
- The following effects are more important in High Speed Design and can cause data corruption.
  - Reflection
  - Crosstalk and interference
  - SSN (simultaneously switching noise)
- Following solutions are employed to improve signal integrity.
  - On die Termination (ODT)
  - Off chip Driver Calibration (OCD)
  - On die Decoupling

What is DDR?

- DDR (Double Data Rate) memory is the new generation SDRAM.
- Like SDRAM, DDR is synchronous with the system clock.
- The big difference between DDR and SDRAM memory is that DDR reads data on both the rising and falling edges of the clock signal.
- SDRAM only carries information on the rising edge of a signal.
- Basically this allows the DDR module to transfer data twice as fast as SDRAM.
  - For example, instead of a data rate of 133MHz, DDR memory transfers data at 266MHz.
- DDR SDRAM also consumes less power, which makes it ideal for notebook computers.

Why DDR2?

- Normal DDR limitations at higher frequencies:
  - Signal integrity
  - Power Consumption
- DDR2 Addresses these challenges by:
  - Operating voltage is reduced from 2.5V to 1.8V
  - Reduced core operating frequency
  - Core frequency = 1/2 the I/O frequency
- Special New Features:
  - 4-bit pre-fetch
  - On-die termination
  - Off-chip driver calibration

DDR Memory Controller Requirements

- Memory burst sizes up to 1024 bytes needs to be supported.

  >> Support write operation

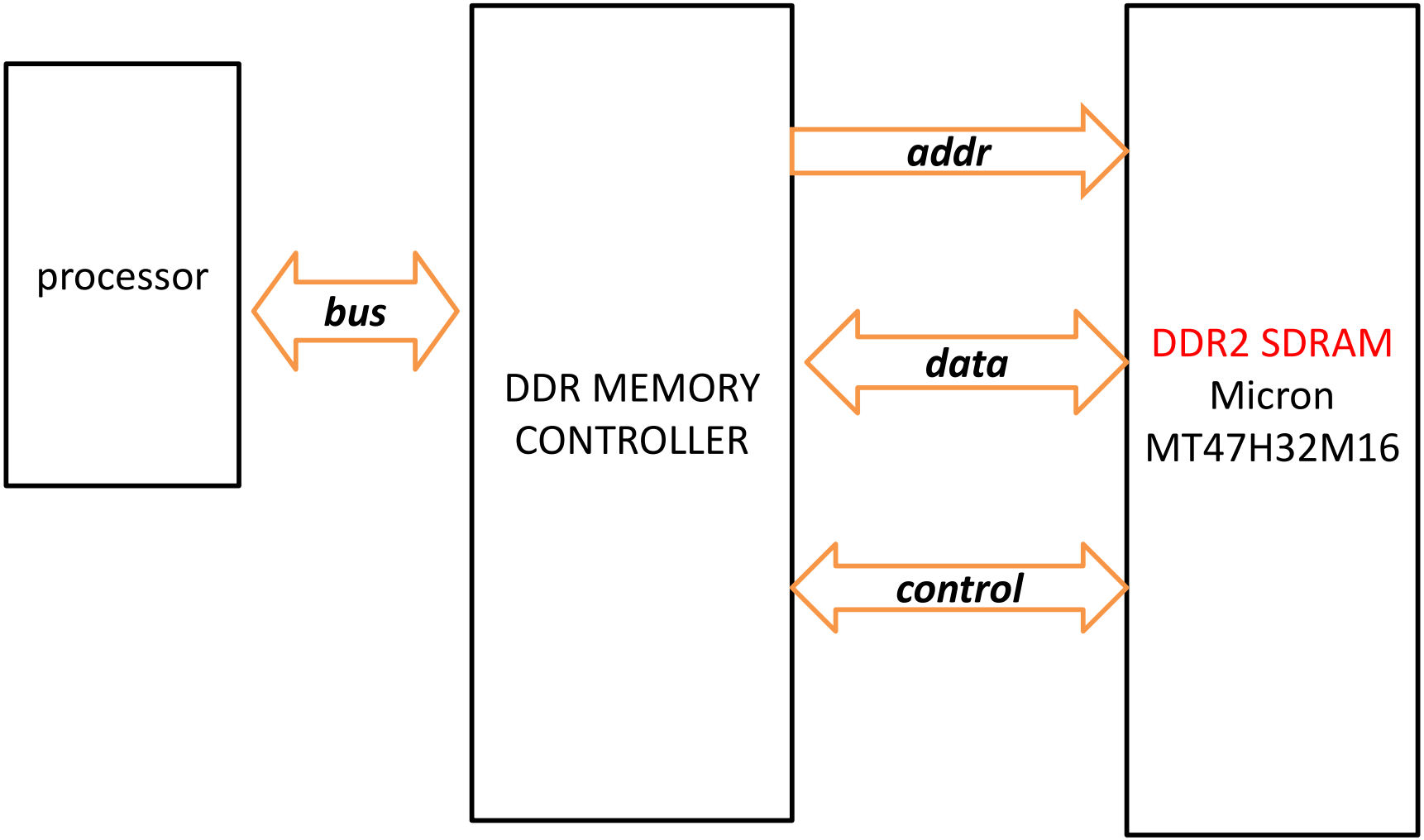
  >> Support read operation

  >> Memory is operated in auto precharge mode, the controller doesn't need to implement the precharge and refresh operations.

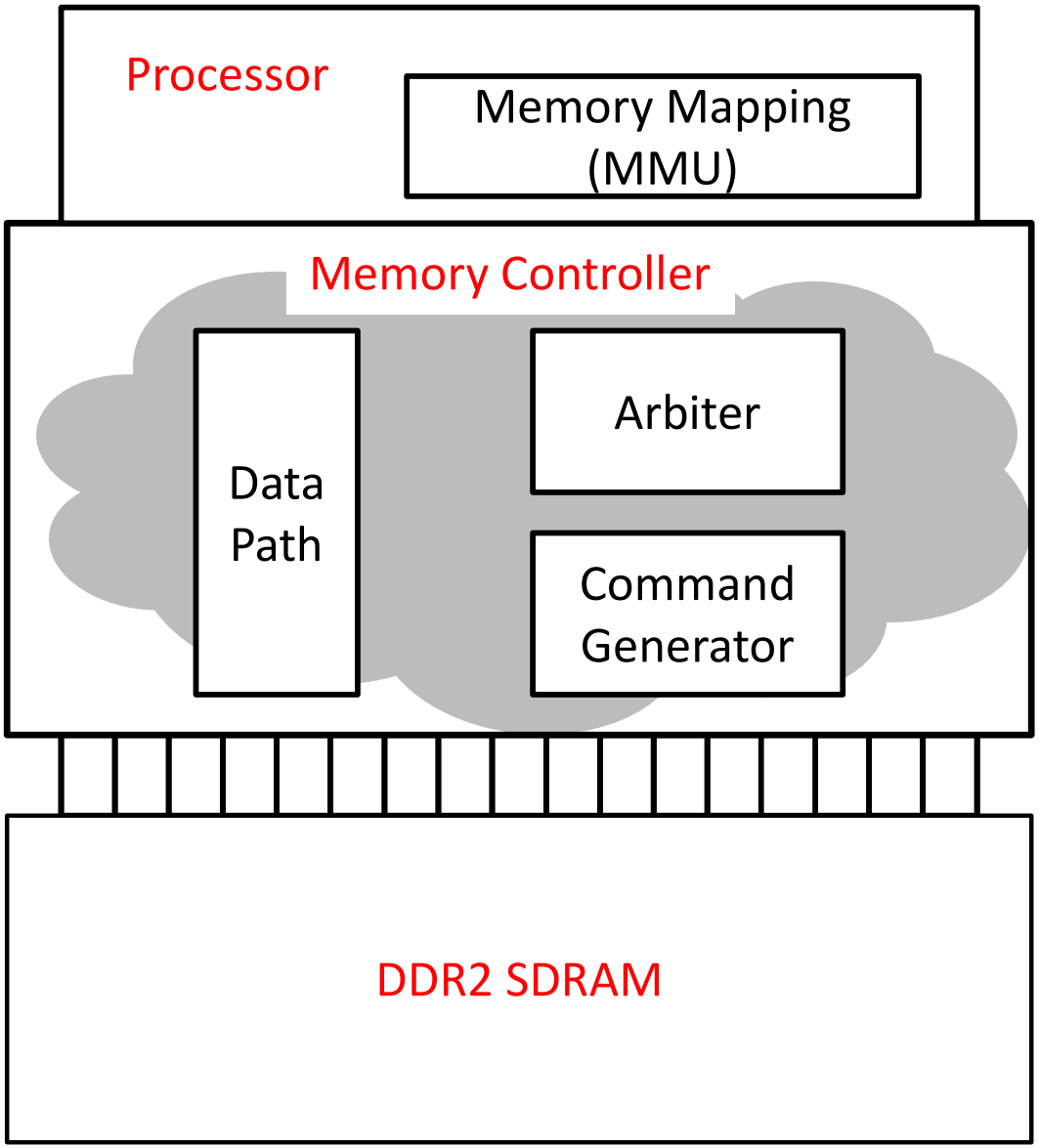
Micron MT47H32M16

- 512Mb x16 DDR2 SDRAM
- Configuration: 8 Meg x 16 x 4 banks
- Refresh count: 8K
- Row address: A[12:0] (8K)
- Bank address: BA[1:0] (4)
- Column address: A[9:0] (1K)
- JEDEC-standard 1.8V I/O (SSTL_18-compatible)
- Differential data strobe (DQS, DQS#) option
- 4n-bit prefetch architecture
- DLL to align DQ and DQS transitions with CK
- 4 internal banks for concurrent operation
- Programmable CAS latency (CL) & Posted CAS additive latency (AL)
- On-die termination (ODT)

BLOCK DIAGRAM

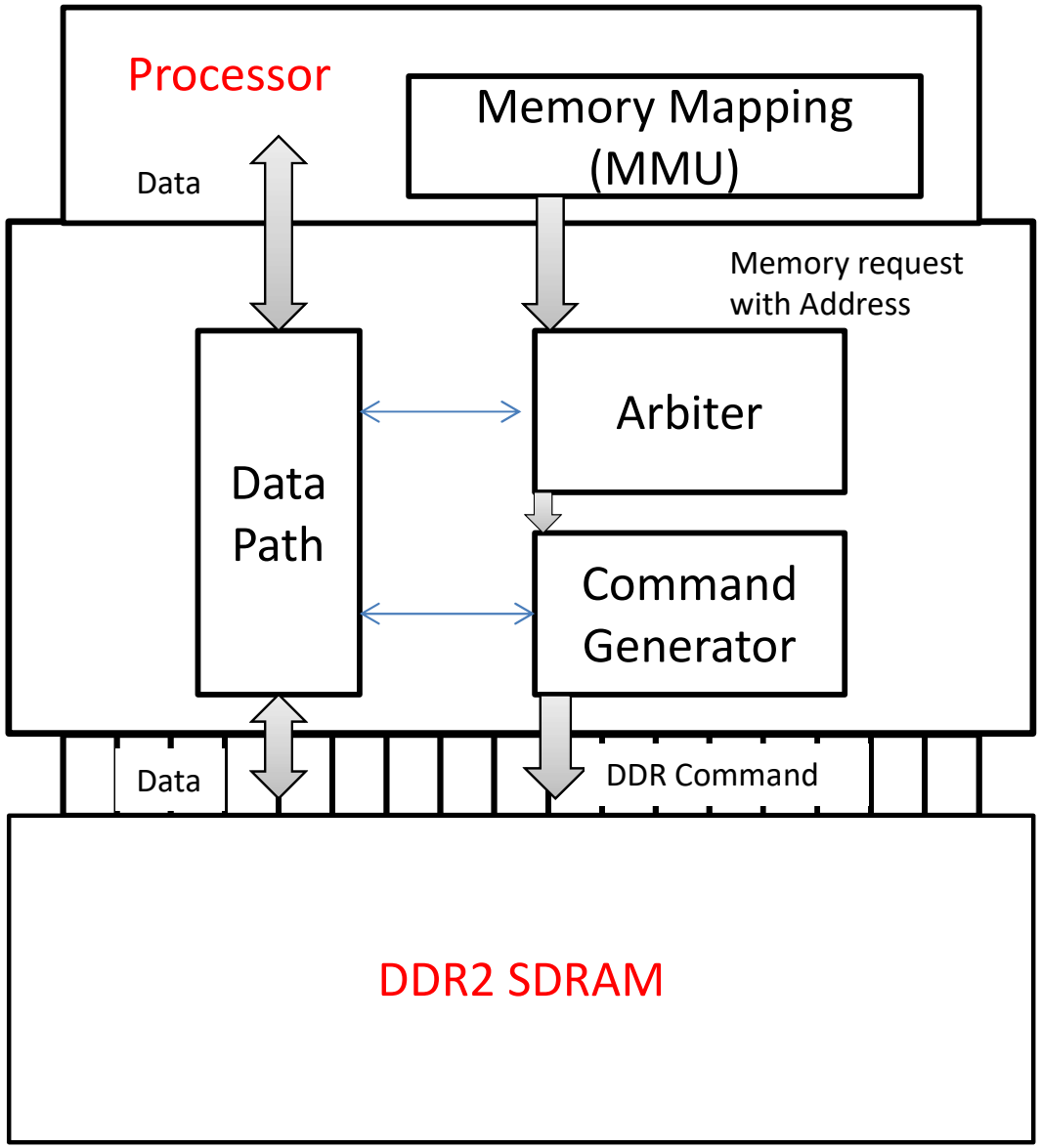


Block Diagram



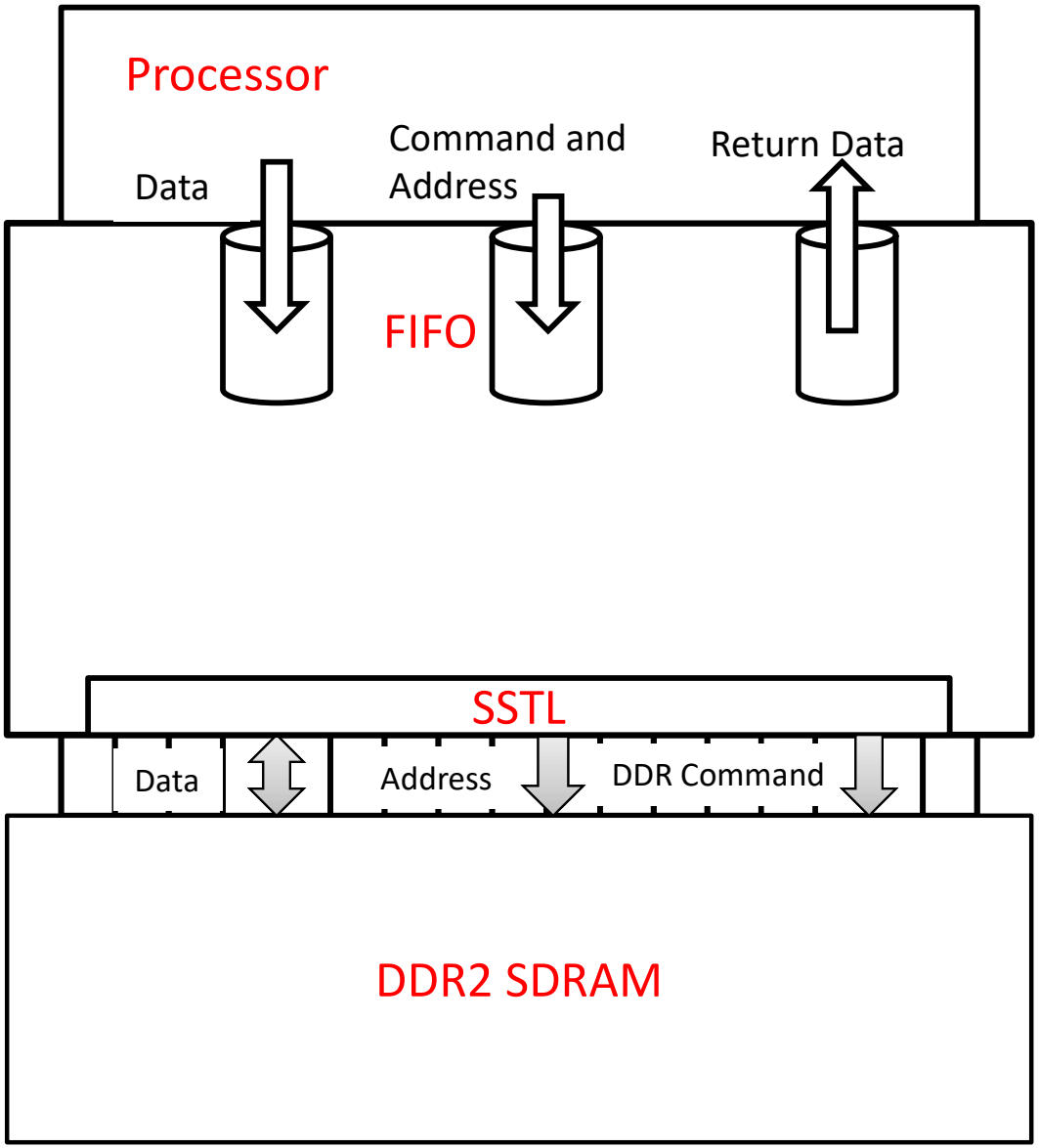
- Processor has the Memory Mapping Unit to generate physical address.
- Controller consists of an arbiter, command generation unit, and the data path.

Block Diagram



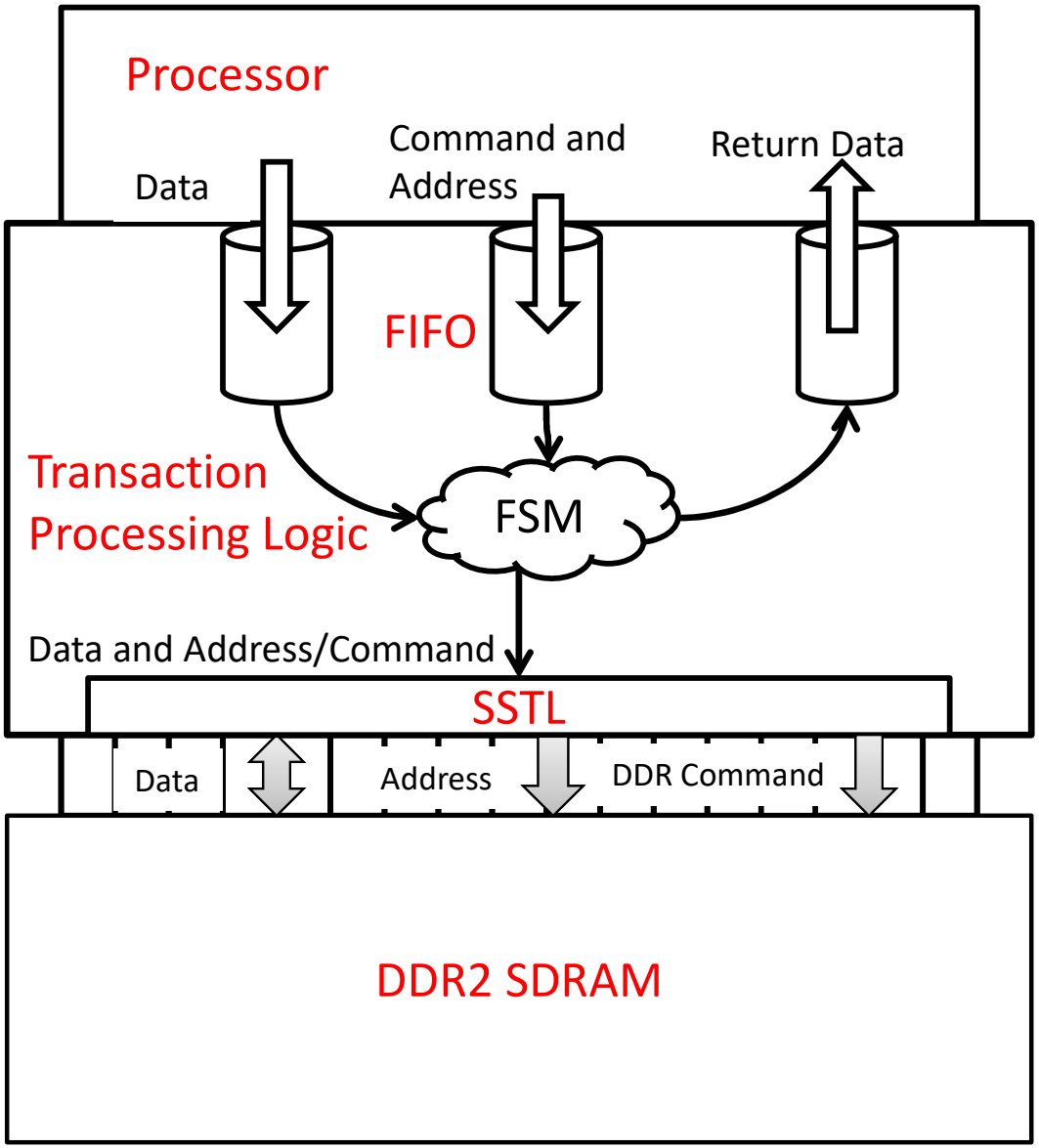
- The command generator provides commands that translates the accesses from the requestors to DDR bursts.
- Data path transports data to and from the memory.
- Arbiter picks the requests from the processor and arbitrates data and address.

Memory Controller



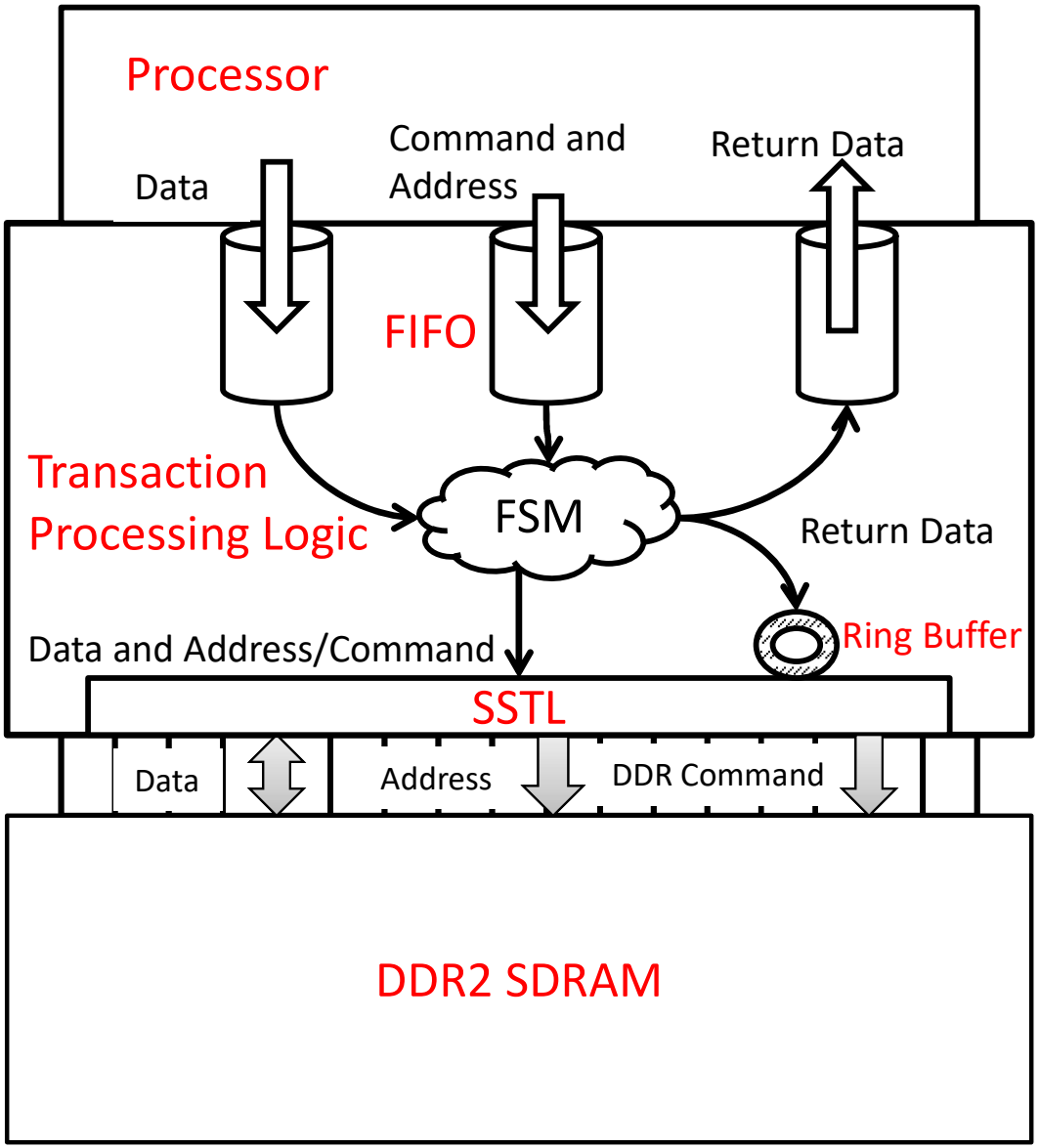
- FIFOs are designed to decouple flow between Processor and Memory Controller.
- Input FIFOs accept Data and Command. Output FIFO sends data to the processor with the correct generated address.
- SSTL Driver/ Receivers with ODT are provided as an IP core by Denali.

Memory Controller



- Arbiter and the command generator are merged into a Finite State Machine: Transaction Processing Logic.
- Processing Logic
  - fetches the command/data queued in the input FIFOs
  - sends the data received from DRAM to the output FIFO with the correct return address

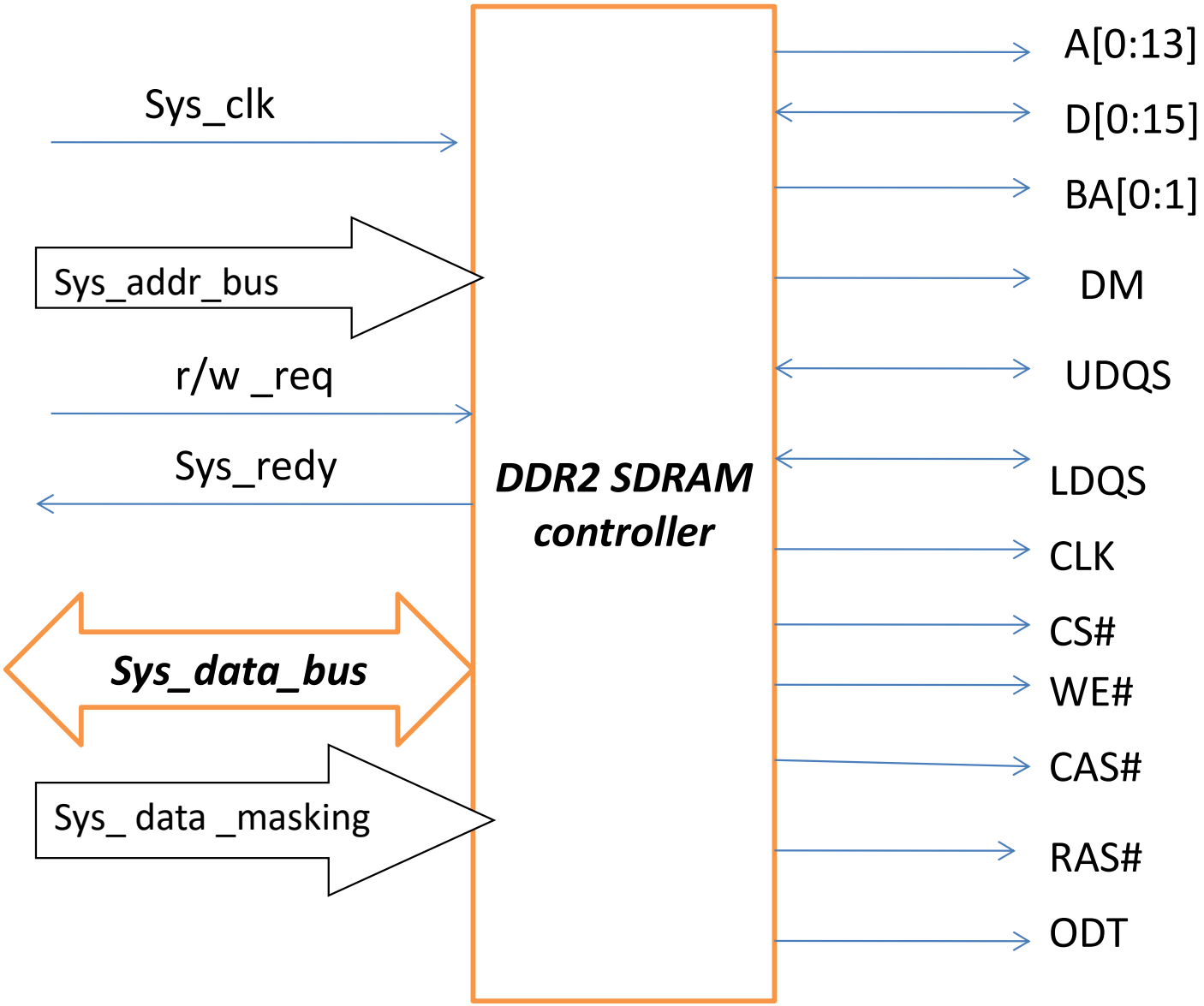
Memory Controller



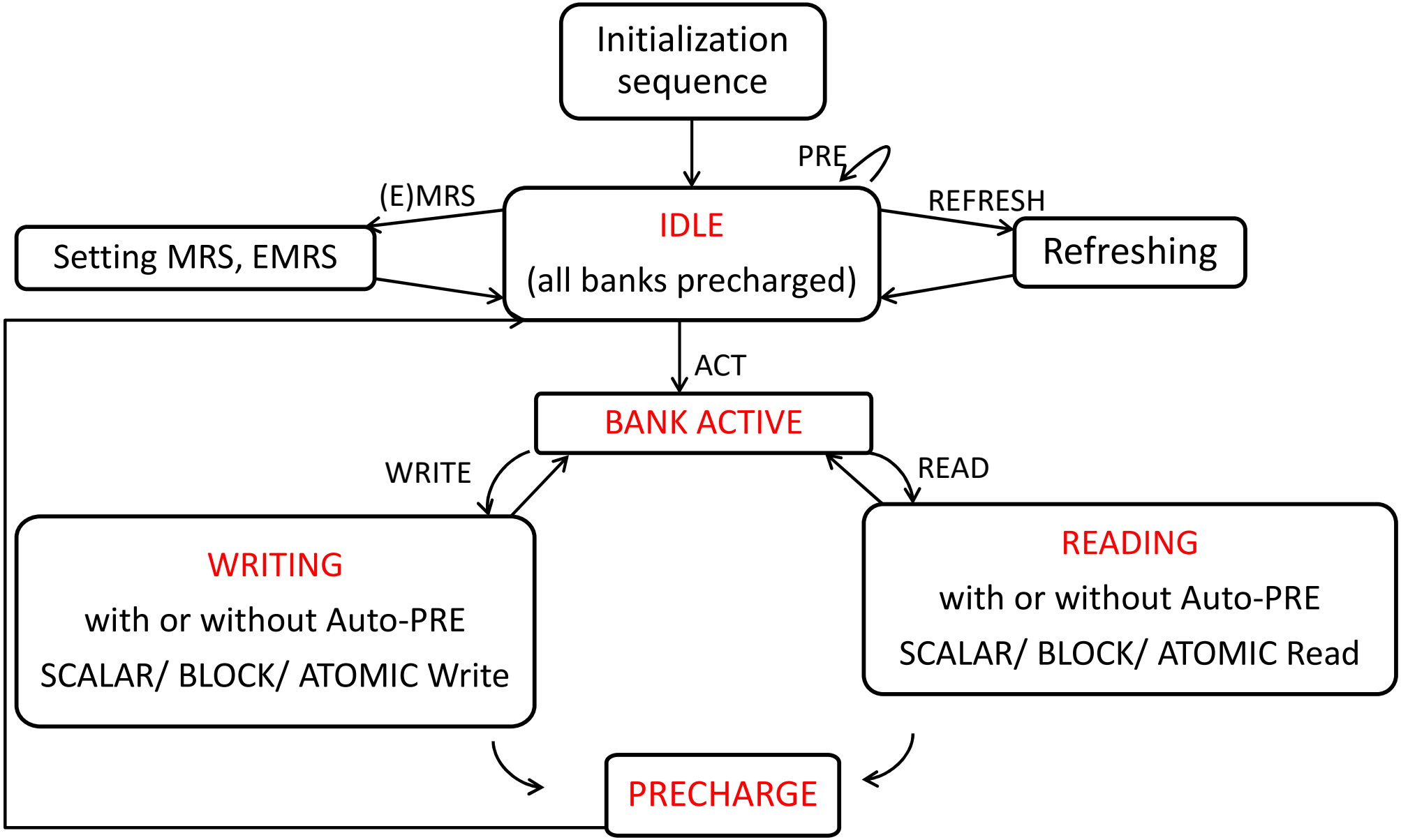
- Data returned from the DDR2 memory during a read access and its strobe signal are asynchronous and not aligned with the DDR2 controller or memory clocks.
- So the Ring buffer temporarily stores data coming from the DDR2 memory and then allows the DDR2 controller to read the data at the controller clock edge.

CORE I/O DIAGRAM

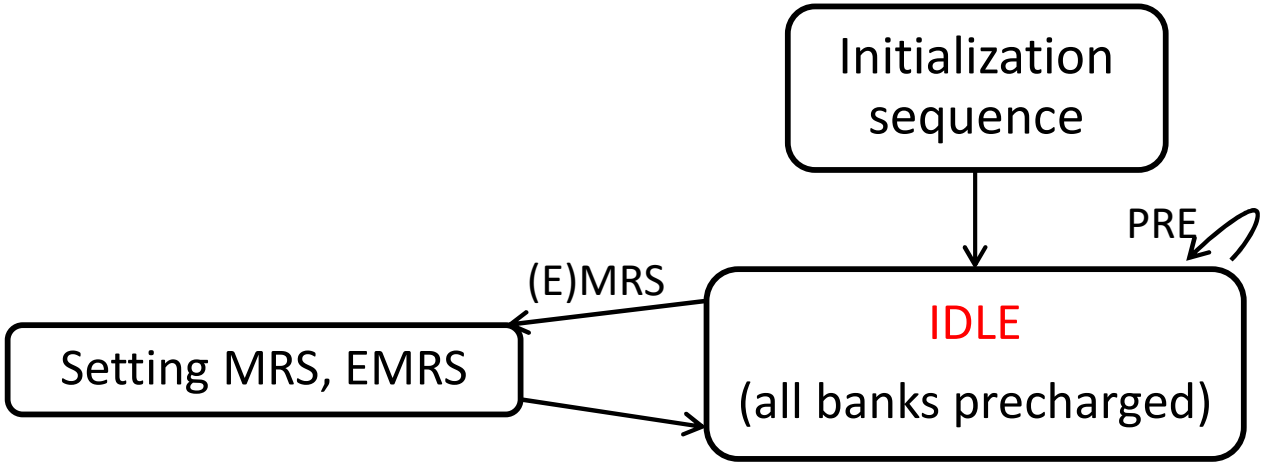
DDR2 CONTROLLER



Transaction Processing Logic



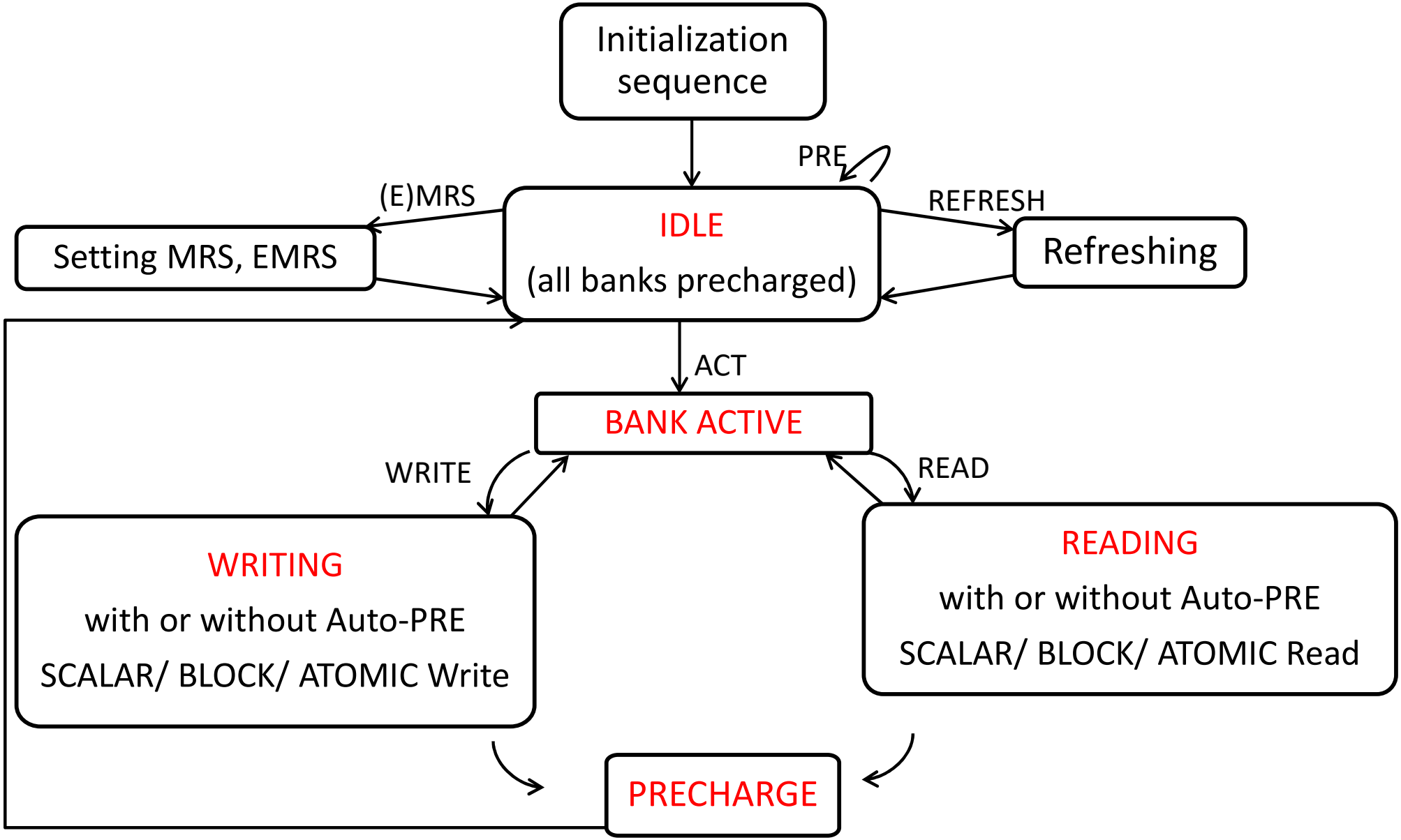
Transaction Processing Logic



Initialization

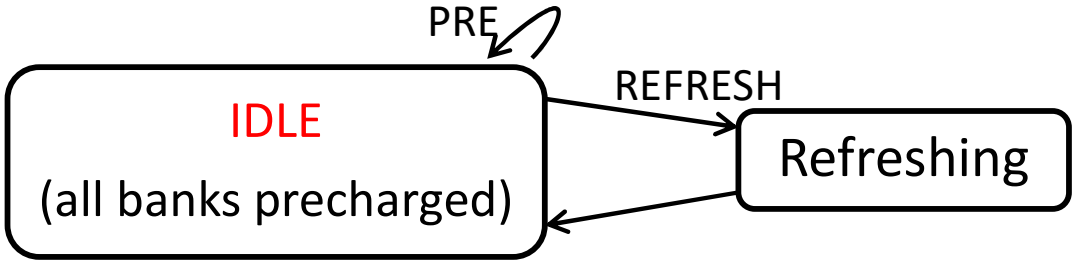
- Before starting any operation, DDR2 SDRAM must be powered up and initialized in a predefined sequence. So we have a separate module, INITIALIZATION ENGINE, with its own state machine.
- When INITDDR input is asserted the Initialization Engine would initialize the DDR2.
- Following this, DDR2 controller memory clock starts, all banks are readied for operation and the various modes registers are configured.
- When initialization sequence is completed, it asserts an output signal READY.

Transaction Processing Logic



Transaction Processing Logic

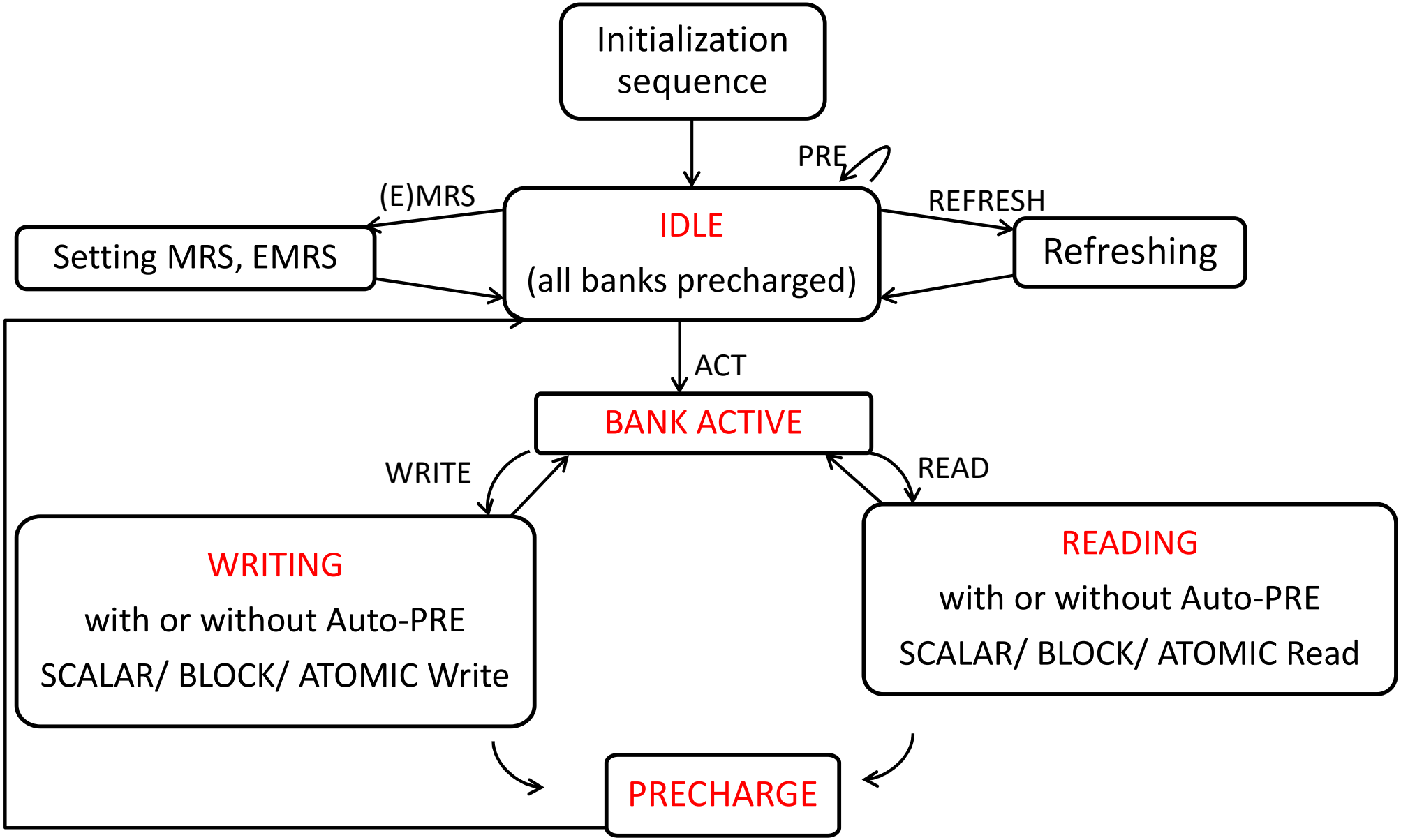
Refresh



- The DDR2 Initialization module is responsible for the refresh. It executes 2 Autorefresh aka Refresh cycles according to these timings:
  - Refresh to active/Refresh command time (tRFC) = 105 ns.
  - Exit self refresh to a non-read command (tXSNR) = tRFC + 10 = 115 ns.
  - Exit self refresh to a read command (tXSRD) = 200 cycles = 800 ns.

- To prevent other states in the controller from violating the maximum refresh delay, the value at which the refresh counter should begin refresh = $(\text{delay between refreshes})_{max}$ - longest operation in the controller

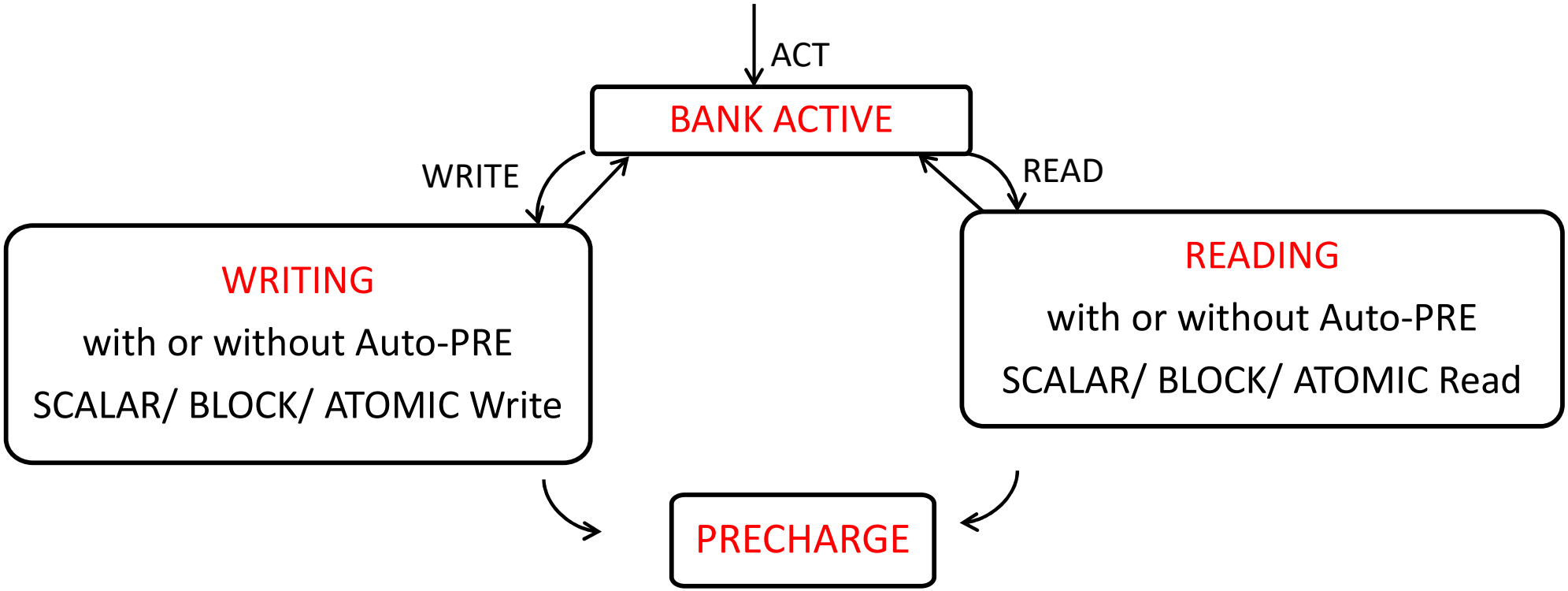
Transaction Processing Logic



Transaction Processing Logic

Operations

- Scalar Read/Write
- Block Read/Write
- Atomic Read/Write



Operations

- Scalar Read/Write
  - The scalar read and write operation operates on a single word out of the accessed burst length number of words. For scalar read, the controller stores the very first value read to the output data FIFO and ignores others. For the scalar write command, data mask is applied to prevent subsequent words from being overwritten.
- Block Read/Write
  - The block read and write commands operates on a large set of words successively. The number of words to be operated on are determined by SZ parameter, which is a multiple of the burst length.
- Atomic Read/Write
  - For an atomic read operation, the data word that is read from the DDR2 memory is operated upon and then stored back to the address specified by the instruction. Simultaneously, the data originally read is sent out.
  - For an atomic write operation, specified operation is performed on the supplied data word, and then written back to the original DDR2 memory address.