

C. Multibank DRAM Technology

The memory used in the MoM-PDA is a special kind of synchronous DRAM with support for burst data transfers. These memory components, called MDRAMs, have been designed by MoSys (see [MoS96]) and are fabricated by SIEMENS (see [Sie97]).

This chapter will give a brief description of the MDRAM operations, which is necessary to understand the functionality of memory controller. For a more detailed description, refer to the data sheets from MoSys or SIEMENS ([Sie96], [Sie97] and [MoS96]).

C.1 The Internal Structure and Interfaces

Generally, MDRAM consist of a number of DRAM matrices (called banks) which are connected to an internal bus. Each matrix contains 256 rows, each row 32 data words (columns), and each data word 32 bits. The number of banks in a MDRAM depends on the type. The type used in MoM-PDA contains 32 banks, which leads to a total capacity of 1 megabyte or 256k words each MDRAM device.

The internal structure of MDRAM is illustrated in figure C-1 at page 254. Address decoders are arranged locally to every bank. A set of two banks shares the same bank address, only differing in the least significant bit. The bank addresses are not hard wired but stored in bank ID registers. After power-up or reset, these address registers are initialized with default values, but the contents can be changed. With reprogramming the bank ID registers, it is possible to map up to 8 MDRAMs into the same address space without any external address decoding hardware. In the MoM-PDA, currently two MDRAMs are cascaded.

Even though data is stored in 32bit words, any data transfer is handled over a 16 bit wide bus. To increase the bandwidth, a 16 bit word is transferred each clock edge. Thus, a complete data word can be transferred within a clock cycle.

All banks are connected to an internal 16 bit wide bus. During write operation, external data is passed directly (via driver) on this bus, and the first 16 bit halfword is sampled with a rising clock edge. When reading, data from internal bus is first stored in a 6 word long FIFO. This enables to read data with a latency, which is useful e.g. for switching driver direction in high speed systems. This way, specifying the clock edge polarity (rising or falling) for the first read operation is possible as well.

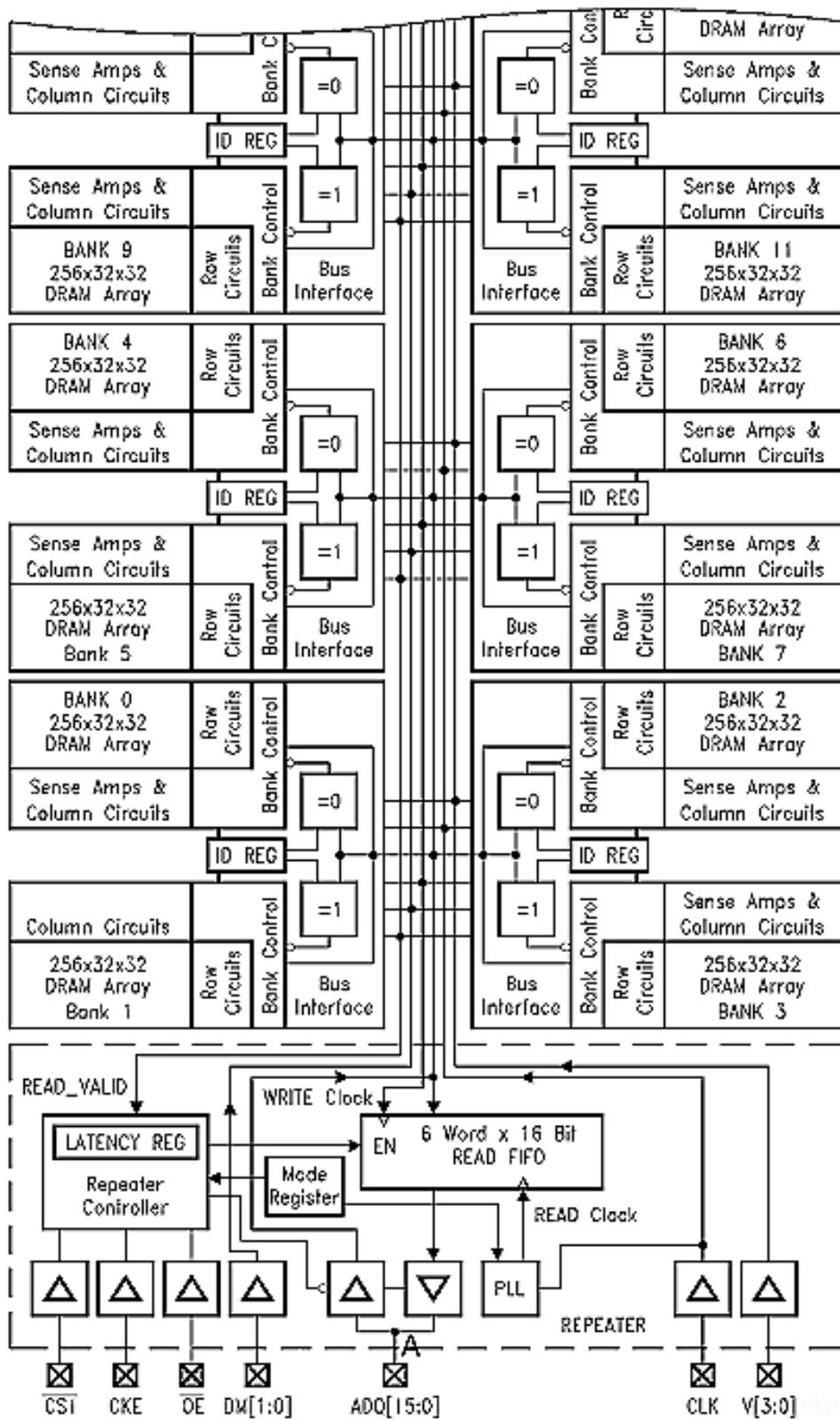


Figure C-1: MDRAM block diagram (reprinted from [Sie96]).

The MDRAM interface contains the following 26 active signal pins:

- ADQ (15:0)
Bidirectional multiplexed address/data bus. When initiating a burst operation, the start address is passed over these pins. During a burst, 16bit data halfwords are read/written over these pins.
- $V(3:0)$, also called command (3:0)
Unidirectional command bus. The four signals CRE (control register enable), \overline{RAS} (row address strobe), \overline{CAS} (column address strobe) and WE (write enable) are combined in a bus. MDRAM operation is controlled by these signals.
- $DM(1:0)$
Signals for data masking. Support for single byte write accesses. In that mode DM addresses a specific byte of a 32 bit word. (This option is not used in MoM-PDA.)
- $CLK, CKE, \overline{OE}, \overline{CS}$
Clock, clock enable, output enable and chip select. In systems consisting of more than one MDRAM Chip on a bus, chip select signal is used to disable all MDRAMs except one, when the bank ID registers are being reprogrammed. Output enable is normally not used, because ADQ pins remain in high impedance state except during a read burst.

All input signals except \overline{OE} and \overline{CS} are synchronous, i.e. they are sampled on the rising edge of the CLK signal. For the appropriate setup and hold times refer to the data sheet [Sie97].

C.2 The MDRAM Commands and Functionality

Each MDRAM operation is initiated by passing an opcode on the command bus ($V(3:0)$) and an address on the ADQ -bus. In most cases, an operation needs a command sequence rather than a single command (command sequences are described detailed in the data sheet, see [Sie97]). Table C-1 at page 256 shows the MDRAM commands and their semantics.

Reset and Setup

After power up, the MEMRESET command sequence must be executed to ensure that all internal registers and state machine are initialized correctly. The bank ID registers are set to their default values. This command sequence consists of the MEMRESET command followed by two STOP commands.

command	hexadecimal value	semantics
activate	02 _H	activate row of specified bank
precharge	03 _H	restore row of activated bank
read	04 _H	start read burst
write	05 _H	start write burst
stop	00 _H	stop running burst
memreset	06 _H	reset memory and IDregister
idregwrite	0D _H	change IDregister contents
moderegwrite	07 _H	write mode register

Table C-1: MDRAM commands and their semantics and opcodes.

In systems with more than one MDRAM on a bus, the bank ID registers should be reprogrammed now. (Addressing is also possible by using the \overline{CS} signals instead of reprogrammed bank IDs. However, addressing via bank IDs is more uncritical in terms of timing.)

When reprogramming bank IDs, all MDRAMs except one must be disabled with the \overline{CS} signals. The sequence IDREGWR-STOP-STOP must be passed onto the bus. The commands are sampled only by the enabled MDRAM. When passing the IDREGWR command, bits 15 to 9 of the *ADQ* bus must be set to the old bank ID value (i.e. the bank ID register which is about to be reprogrammed). On the next rising clock edge (when the first STOP command is being passed) bits 15 to 9 of the *ADQ* bus must be set to the new bank ID register content. On the next *falling* clock edge (when second STOP command is being passed) the ID register content will change. A timing diagram of the reprogramming sequence is shown in figure C-2.

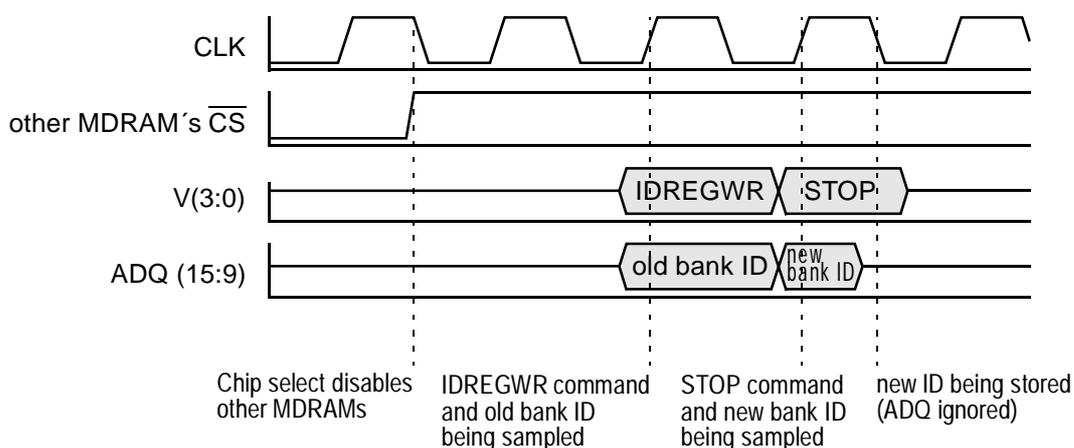


Figure C-2: Timing of ID register reprogramming command sequence.

Except the first MDRAM, which can use the default ID values, this procedure must be repeated for every bank ID register in every MDRAM.

When all bank IDs are properly set, a system of up to 8 MDRAMs connected to the same bus can be handled as a single MDRAM component without any extra address decoding hardware.

Read and Write Operations

Figure C-3 illustrates the infrastructure needed to perform read or write bursts. Before a stored data word can be accessed, the line that contains the data word is read out from the appropriate DRAM matrix and stored in a register (sense amplifier register, senseamp). This process is called *activation* of the bank. For activating a bank, the ACTIVATE command must be passed on the command bus and the ADQ bus must be set to the bank address (bits 15 to 8) and the row address (bits 7 to 0).

A bank only can be activated once a time, because every bank contains only one senseamp register. Due to the fact that the bank matrix consists of dynamic memory cells, the content of the activated line in the matrix is destroyed when activated. Before another line of the same bank can be activated, first the senseamp contents must be restored to the bank matrix (precharge command, see below).

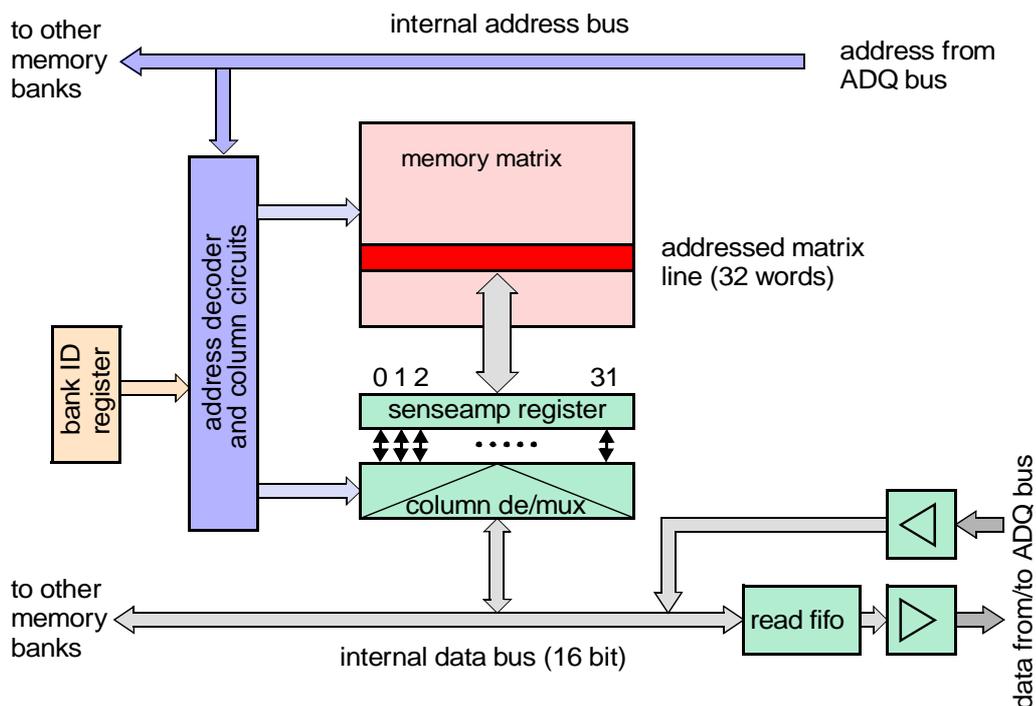


Figure C-3: Address and data paths for read/write operations in the MDRAM.

After activation read/write operations can be performed. The *ADQ* bus must be set to the bank ID (bits 15 to 8) of the activated bank and the column address of the first desired word (bits 4 to 0). The read or write command must be passed on the command bus.

A write burst will start at the next rising clock edge after sampling the write command. The first half word is sampled at the rising clock edge and the second half word at the subsequent falling clock edge. The next data words will be sampled in the same way on the following clock cycles. A write burst reaching the end of the senseamp register will wrap to the beginning, i.e. a burst that is longer than then 32 words (= length of the senseamp) will overwrite the first written data. There is actually no limit for the burst length, a burst will run until a *STOP* command is passed.

Read operations work in the same way. A complete data word is transferred from the senseamp register every clock cycle and a read burst reaching the end of the senseamp will wrap to the beginning. Unlike a write burst, the read data can be stored in a 6 word read FIFO before passing on the *ADQ* bus for realizing a read latency. This feature is not used in MoM-PDA. (At low frequencies, the MDRAM works without PLL setting, thus the read FIFO is not available. See data sheet [Sie97].)

When passing a *STOP* command, the currently running burst will terminate, but the transfer of the current data word will complete. Refer to figure C-4 for a sample timing.

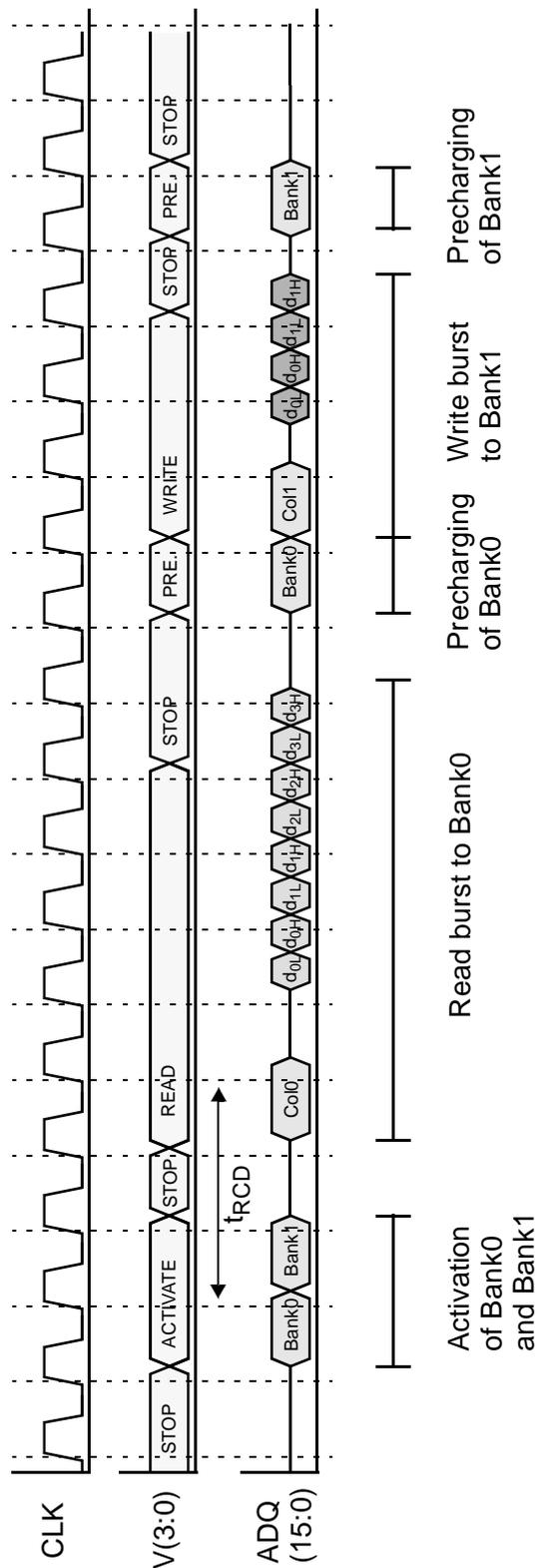


Figure C-4: Sample timing of a burst operation.

When a burst operation is completed, the contents of the senseamp register must be restored to the DRAM matrix. Restoring the senseamp is called *precharging* the bank. A bank must be precharged subsequently to every burst operation, even if senseamp contents are unchanged (e.g. after a read burst). As mentioned above, the content of the addressed DRAM line has been destroyed through activation. Therefore, a precharge must be performed in any case. Precharging is done by passing the PRECHARGE command on the command bus and the bank ID on the *ADQ* bus (bits 15 to 8). Unlike activation, the row address is not needed, bits 7 to 0 of *ADQ* are not evaluated.

The Activation of Multiple Banks

Each bank has a local senseamp register, therefore multiple banks can be active simultaneously. In a former version of MDRAMs, group activation/precharge commands were supported, which could be used to ACTIVATE/PRECHARGE four banks simultaneously, but these commands are not implemented any longer. However, it is still possible to activate any number of banks before a burst is started. When MDRAMs operate at high frequencies, between activation of a bank and starting a burst in the same bank, a number of clock cycles (up to 3) must be paused (T_{RCD} , activate to read/write delay, see data sheet). Using this small period to activate or precharge another bank is called "hidden RAS".

The sample shown in figure C-4 at page 258 illustrates the execution of read/write commands using hidden RAS. First, bank 0 and bank 1 are activated immediately after each other. *ADQ* is set to the appropriate bank ID (bits 15 to 8) and row address (bits 7 to 0). It is assumed, that T_{RCD} reaches over 3 clock cycles, so a STOP command must be performed for one clock cycle before bank 0 can be accessed. Now, a read burst for bank 0 is initiated by passing the READ command and the column address for the first data word to be read. The half words (d_{xL} and d_{xH}) are passed on the *ADQ* bus on consecutive clock edges. *ADQ* bus must not be driven by any device except the MDRAM. After 3 words are read, a STOP command is passed. The burst will terminate after reading another word. Then, the senseamp content of bank 0 is restored by passing the PRECHARGE command and the bank 0 ID. Bank 1 is still activated, a write burst can be started to bank 1 without any delay. The write burst is terminated after writing 2 words by passing a STOP command. At least, bank 1 is precharged.

C.3 The Memory Refresh

The MDRAMs memory cells must be refreshed periodically, otherwise the stored information will get lost. MDRAMs do not support a special command for refreshing. A DRAM matrix is simply refreshed by activating and precharging all its lines. All lines of all banks in all MDRAMs must be refreshed within a refresh period t_{Ref} . The refresh period is reported to be 16 ms (see data sheet [Sie97]), but MDRAMs are proved to work also at a refresh period of 32 ms. According to a statement from

SIEMENS, the chip die is manufactured using the same process like standard DRAM cells, which are specified for a refresh period of 64 ms. Thus, MDRAMs should also work with $t_{\text{Ref}} = 64\text{ms}$.

Due to the rather low clock frequency used in the MoM-PDA prototype, it might be necessary to interrupt a running burst operation only for refreshing a DRAM line. In highly clocked systems, a refresh operation may be delayed until the current burst has come to an end. The average time between two refresh operations will be almost constant, so that a t_{Ref} violation can be ruled out. At low clock frequencies, this is not possible because delaying a refresh could lead to a collision with the next refresh operation.

An ACTIVATE command cannot be followed by a PRECHARGE command immediately but after a small period (t_{RAS} , activate to precharge delay, see data sheet [Sie97]). Thus, it makes no sense to perform single refresh operations at high clock frequencies, because t_{RAS} would be wasted. When refreshing, hidden RAS may also be used for multiple refresh operations. That is, a number of banks is activated subsequently and then precharged in the same order. Depending to the clock frequency, t_{RAS} can be up to 4 clock cycles. Therefore it is useful to perform quadruple refreshing instead of single refresh operations.

In the MoM-PDA, t_{RAS} is shorter than one clock period, so that single refresh operations could be done without any waste of time. But for a later high-speed implementation, a quadruple refreshing scheme is already implemented. The advantage of this scheme is a 4 times higher number of clock cycles between two refresh operations.

The refreshing process is growing quite complex, when refresh operation must be done to a line of an already activated bank. In this case, two methods are common:

- The bank is not refreshed now but in the next refresh operation. This might be a great risk, because the bank could be activated for a long time, so that no refresh is performed at all for this bank.
- The activated bank is being precharged and reactivated after refresh. This method is sure but will lead to a complex refresh logic especially for multiple refreshing. In worst case, all 4 banks must be precharged and the addresses of the precharged rows must be stored.

The MoM-PDA Burst Control Unit implements the second refreshing method with the disadvantage that refresh logic is one of the most complex components of the memory controller.

More information on MDRAMs and the MoM-PDA Burst Control Unit can be found in [Bed98].