ARM Memory Interface

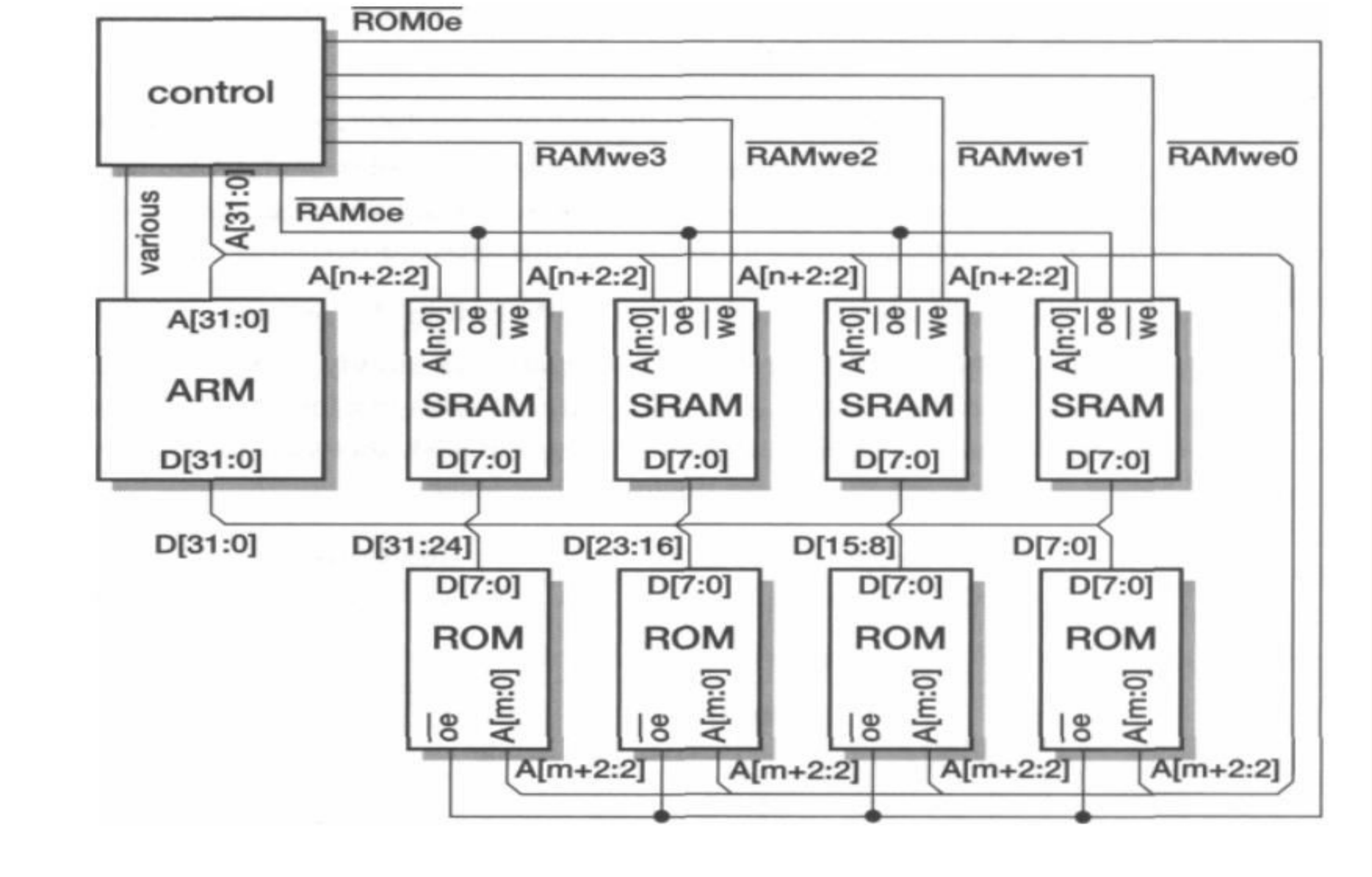
Arm Bus Signal, Single Memory Interface, Control signal and Different accessing ways

ARM Bus Signals

Memory bus interface signals include the following:

- A 32 bit address bus
- A 32 bit bidirectional bus
- Signals specifying what kind of memory requirement *(mreq, seq)*
- Signals specifying direction and size of memory
- Bus Timing and control signals *(abe, ale, ape, dbe, lock, bl[3:0]).*

A basic ARM Memory System(1/2)



A basic ARM Memory System(2/2)

- This figure illustrates the connection of 8-bit memory parts, which are a standard configuration for SRAMs and ROMs.
- The output enable signals *(RAMoe* and *ROMoe)* and the write enables *(RAMwe) is seen in figure*
- ROM can have a word access unlike RAM where separate enable for every 8 bit needed
- For ROM word access, [31:2] bits required, [1:0] can be used as control signals and for half word access [0] bit can be used for control signal.

Control Logic

It performs mainly three functions:

- ❖ It decides when to activate the RAM and when to activate the ROM.
- ❖ It controls the byte write enables during a write operation
- ❖ It ensures that the data is ready before the processor continues.

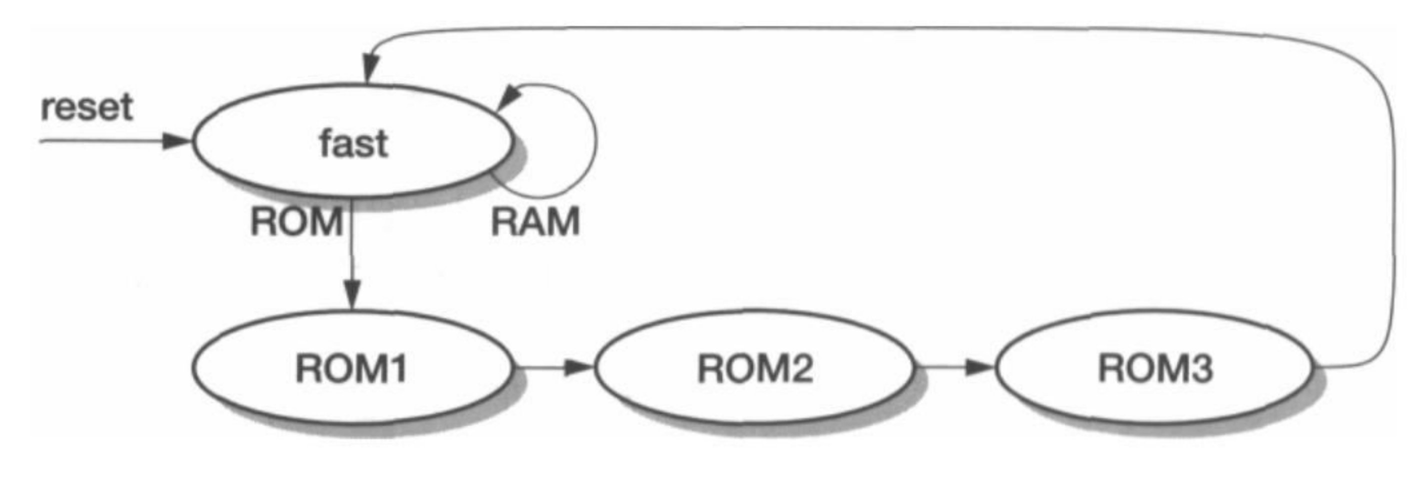
Stretching Access cycles

- The ARM7TDMI processor does not contain any dynamic logic that relies on regular clocking to maintain the internal state. Therefore, there is no limit upon the maximum period for which **MCLK** can be stretched, or **WAIT state** held LOW. There are two methods available to stretch access times as described in:
  - *Modulating MCLK*
  - *Use of WAIT state to control bus cycles*.

➢ Modulating MCLK

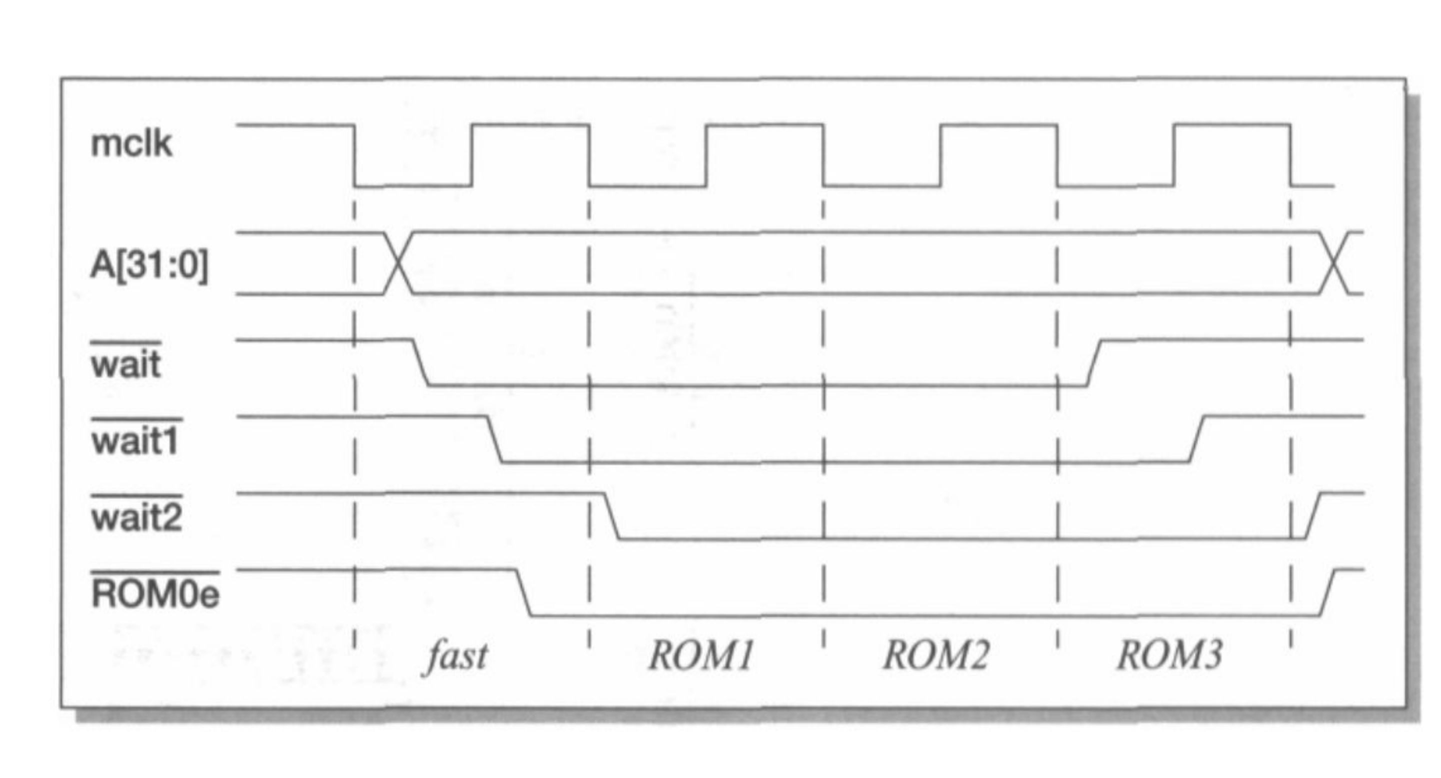
- All memory timing is defined by **MCLK**, and long access times can be accommodated by stretching this clock. It is usual to stretch the LOW period of **MCLK**, because this enables the memory manager to abort the operation if the access is eventually unsuccessful.
- **MCLK** can be stretched before being applied to the processor, or the **WAIT state** input can be used together with a free-running **MCLK**. Taking **WAIT state** LOW has the same effect as stretching the LOW period of **MCLK.**

➢ Wait States



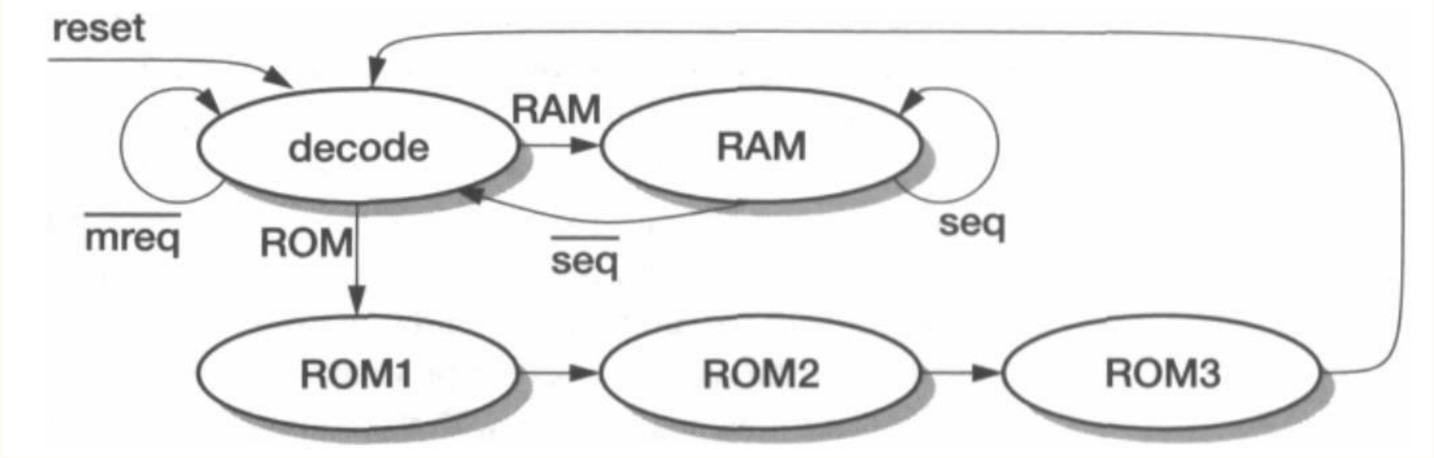
ROM Wait control state transition diagram

The timing diagram for the ROM wait state logic


mclk
A[31:0]
$\overline{\text{wait}}$
$\overline{\text{wait1}}$
$\overline{\text{wait2}}$
$\overline{\text{ROM0e}}$
fast
ROM1
ROM2
ROM3

Memory Access required or not?

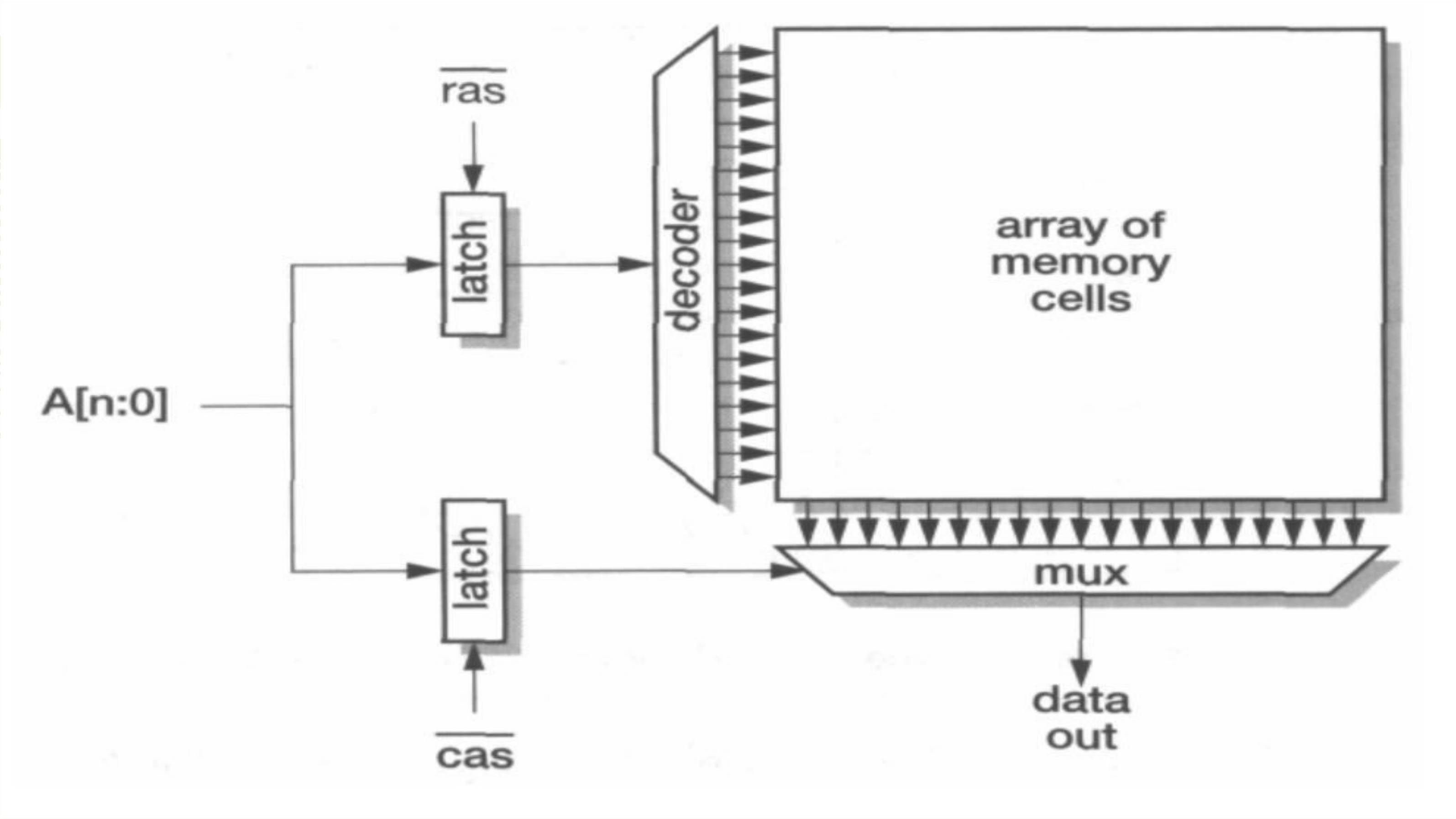
- State transition diagram with a wait state for address decoding:



Dynamic RAM (DRAM)

- Cheapest memory technology
- Stores information as electrical charge on capacitor
- Memory control logic refreshes the DRAM, so that information can be stored safely somewhere else.

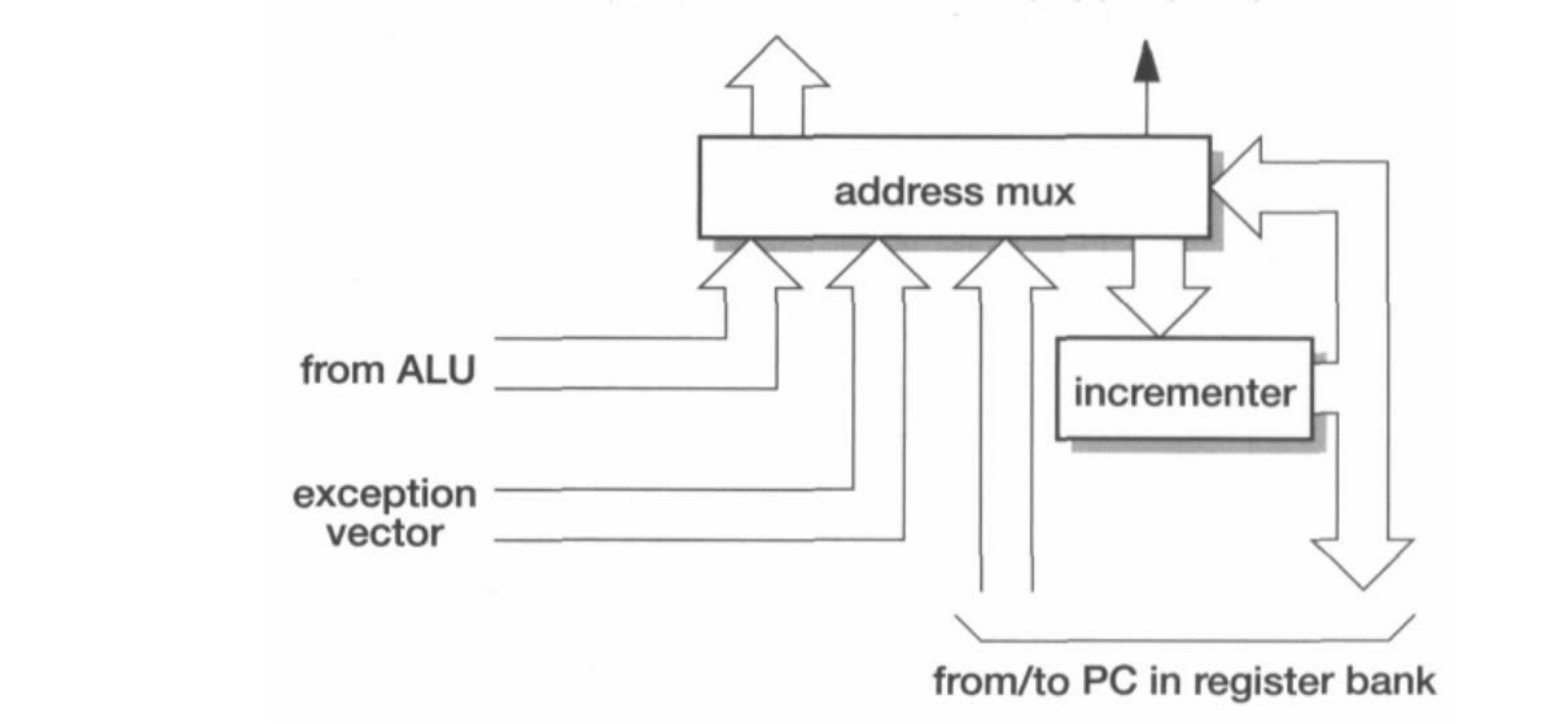
DRAM Memory organization(1/2)



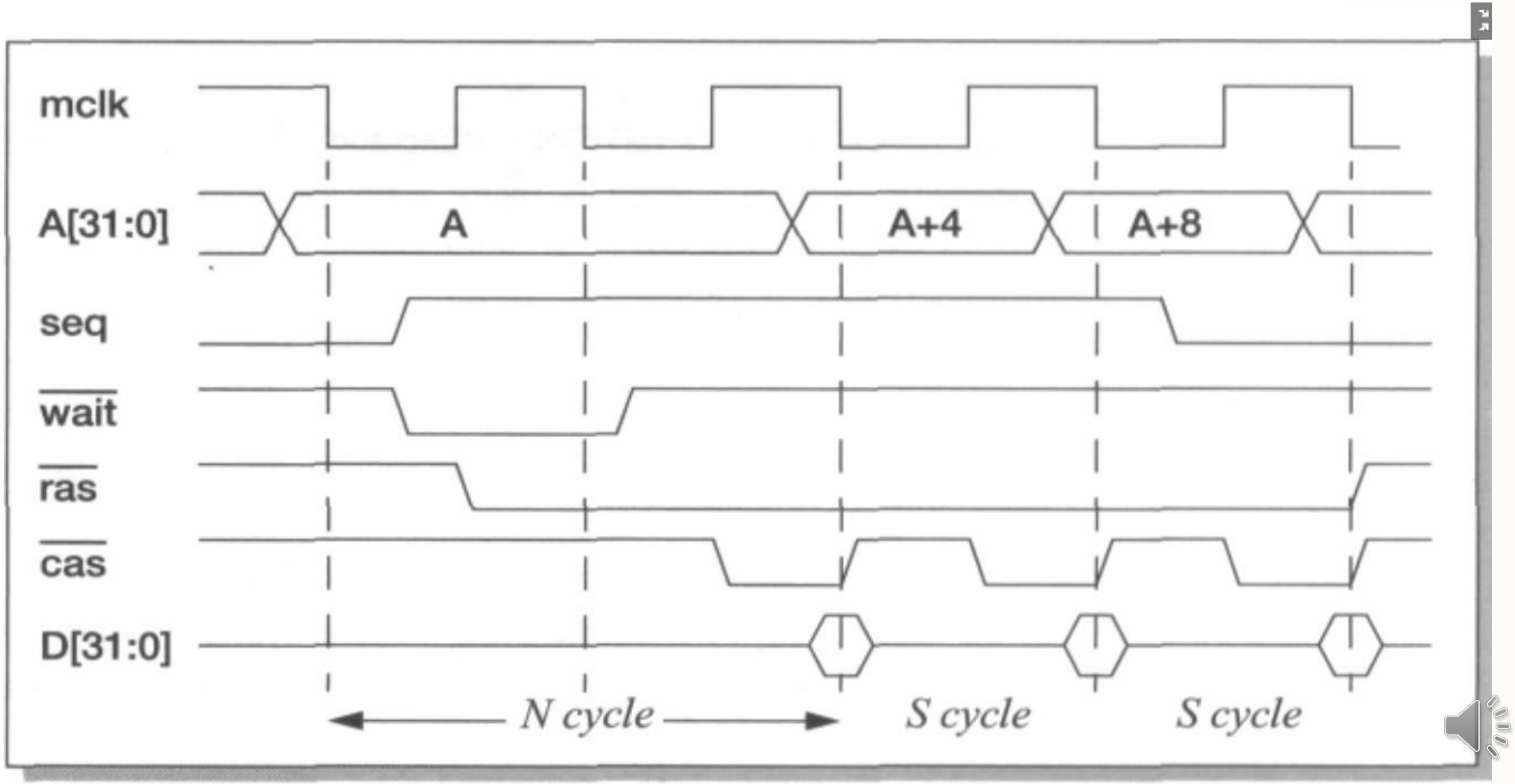
DRAM Memory organization(2/2)

- The row address is presented and latched using the active-low row **address strobe** signal *(ras)*
- The column address is presented and latched using the active-low **column address strobe** *(cas)*
- Very advantageous to use *cas-only* accesses whenever possible

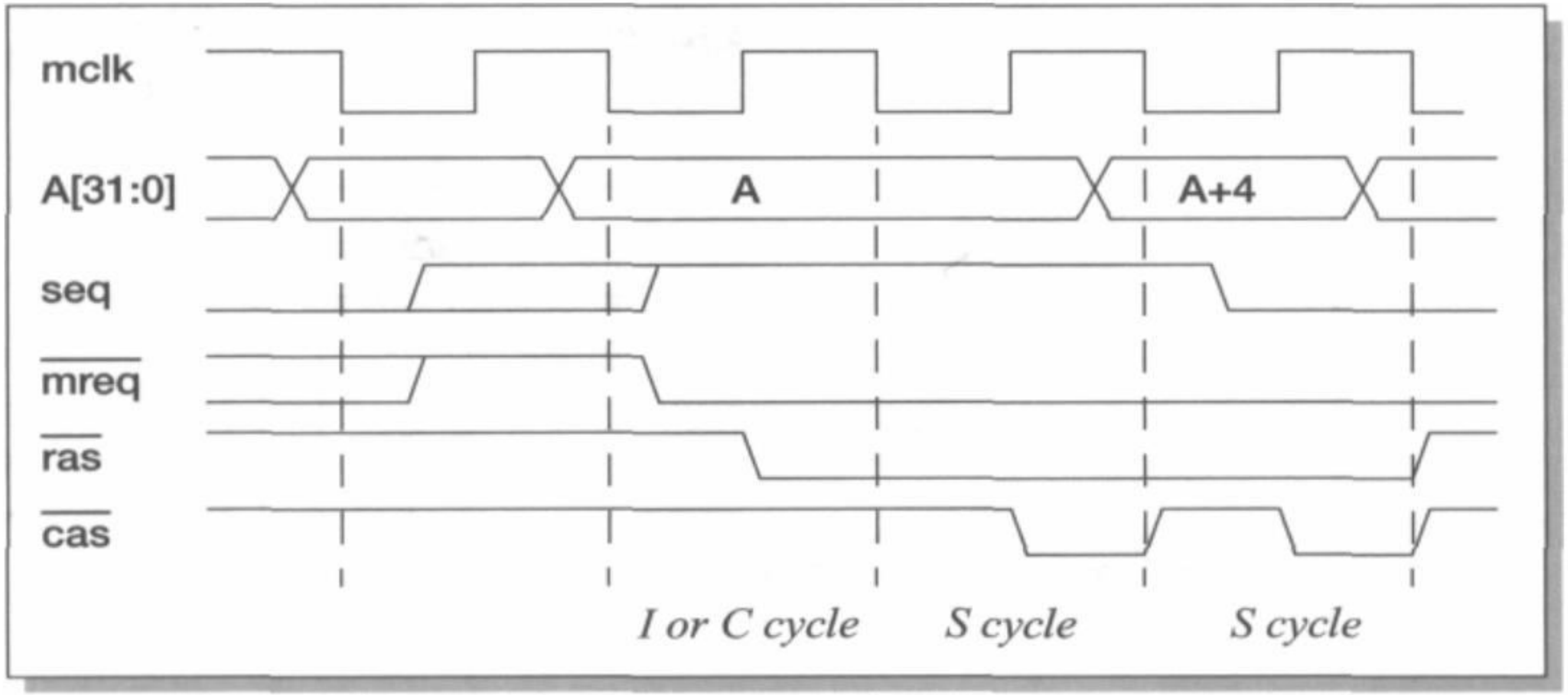
ARM Address Incrementor



DRAM Timing Illustration



DRAM timing after an internal cycle



Thank You…!