Computer Organization and Architecture

Module 5 (Part 2)

Design of Memory Subsystems

Prof. Indranil Sengupta

Dr. Sarani Bhattacharya

Department of Computer Science and Engineering

IIT Kharagpur

1

Memory Interfacing and Addressing

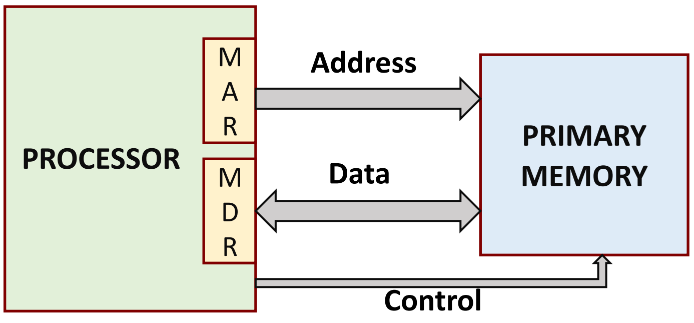
2

Memory Interfacing

- Basic problem:
  - Interfacing one of more memory modules to the processor.
  - We assume a single level memory at present (i.e. no cache memory).
- Questions to be answered:
  - How the processor address and data lines are connected to memory modules?
  - How are the addresses decoded?
  - How are the memory addresses distributed among the memory modules?
  - How to speed up data transfer rate between processor and memory?

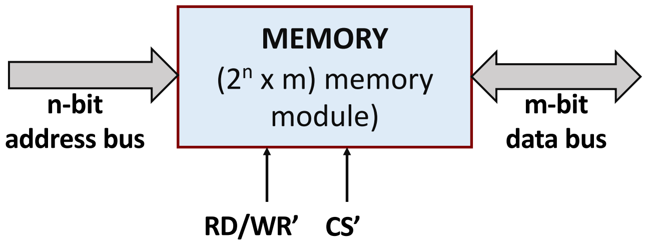
3

3



The processor's view of memory

- Typical interface of a memory module.
- Real chip may contain more signal lines (e.g. DRAM).



4

4

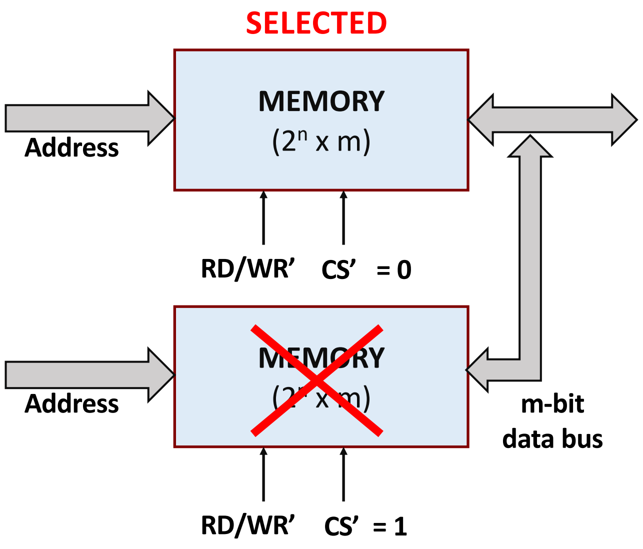
A Note About the Memory Interface Signals

- The data signals of a memory module (RAM) are typically bidirectional.
  - Some memory chips may have separate data in and data out lines.
- For memory *READ* operation:
  - Address of memory location is applied to *address lines*.
  - *RD/WR’* control signal is set to 1, and *CS’* is set to 0.
  - Data is read out through the *data lines* after memory access time delay.
- For memory *WRITE* operation:
  - Address of memory location is applied to *address lines*, and the data to be written to *data lines*.
  - *RD/WR’* control signal is set to 0, and *CS’* is set to 0.

5

5

- Why is *CS’* signal required?
  - To handle multiple memory modules interfacing problem.
  - We typically select only one out of several memory modules at a time.
- What happens when *CS’ = 1*?
  - When a memory module is *not selected*, the data lines are set to the *high impedance state* (i.e. electrically disconnected).
  - An example scenario is shown.



6

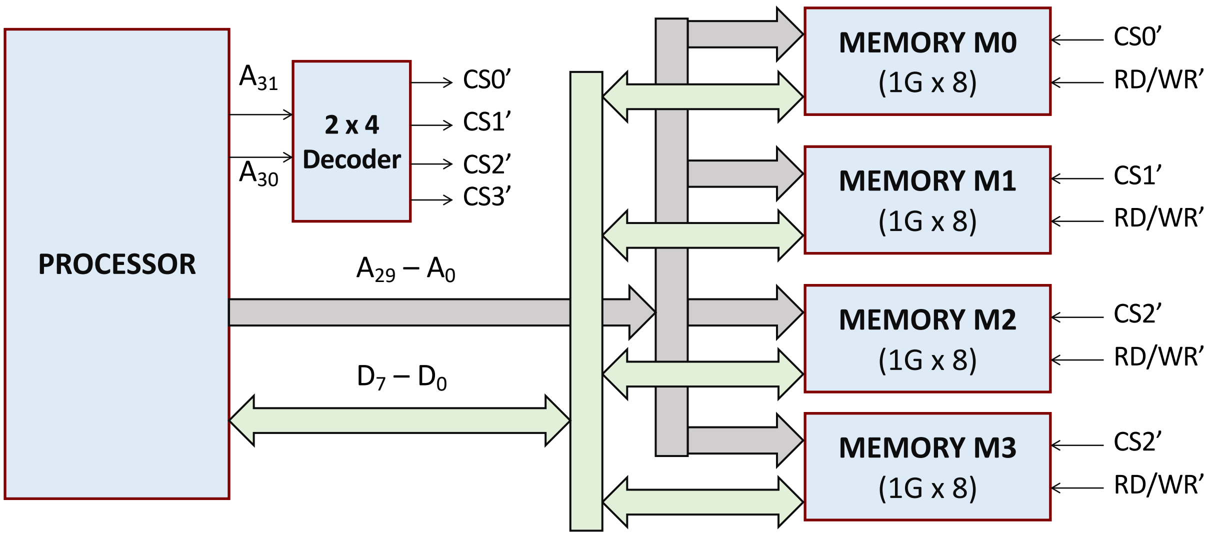
6

An Example Memory Interfacing Problem

- Consider a MIPS32 like processor with a 32-bit address.
  - Maximum memory that can be connected is 2^{32} = 4 Gbytes.
  - Assume that the processor data lines are 8 bits.
- Assume that memory chips (RAM) are available with *size 1 Gbyte*.
  - 30 address lines and 8 data lines.
  - Low-order 30 address lines (A_{29}-A_0) are connected to the memory modules.
- We want to interface *4 such chips* to the processor.
  - Total memory of 4 Gbytes.

7

7



8

8

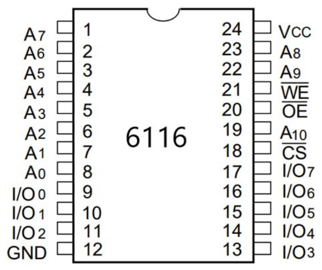
- High order address lines (A_{31} and A_{30}) select one of the memory modules.
- **When is M0 selected?**
  - Address is: 0 0 x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x
  - Range of addresses is: 0x00000000 to 0x3FFFFFFF
- **When is M1 selected?**
  - Address is: 0 1 x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x
  - Range of addresses is: 0x40000000 to 0x7FFFFFFF
- **When is M2 selected?**
  - Address is: 1 0 x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x
  - Range of addresses is: 0x80000000 to 0xBFFFFFFF
- **When is M3 selected?**
  - Address is: 1 1 x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x
  - Range of addresses is: 0xC0000000 to 0xFFFFFFFF

- **An observation:**
  - Consecutive block of bytes are mapped to the same memory module.
  - For MIPS32, we have to access 32 bits (4 bytes) of data in parallel, which requires four sequential memory accesses here.
  - We shall look at an alternate memory organization later that would make this possible.
    - Called *memory interleaving*.

Exercise 1

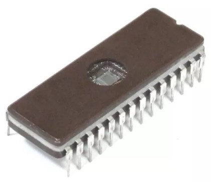
- 6116 is a 2K x 8 RAM chip. Build a 16K x 8 memory using 6116 chips. Show all the connections, and state how the addresses are distributed across memory modules.

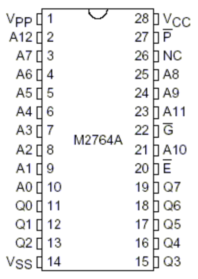
6116 Pinout



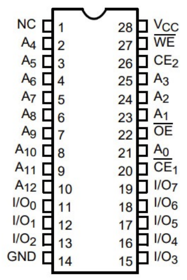
Exercise 2

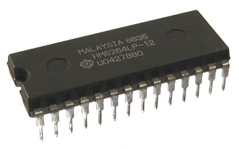
- 6264 is an 8K x 8 RAM chip, and 2764 is an 8K x 8 EPROM chip. Build a 32K x 8 memory system with 8 KB EPROM and 24 KB RAM. Show all the connections, and state how the addresses are distributed across memory modules.





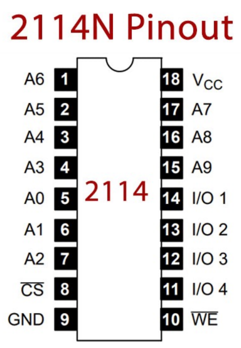
6264 Pinout





Exercise 3

- 2114 is a 1K x 4 RAM chip. Build a 4K x 8 memory using 2114 chips. Show all the connections, and state how the addresses are distributed across memory modules.



13

Memory Interleaving

14

Improved Memory Interface for MIPS32

- We make small changes in the organization so that 32-bits of data can be fetched in a single memory access cycle.
  - Exploit the concept of *memory interleaving*.
  - Consecutive bytes are mapped to different memory modules.
- The main changes:
  - High order 30 address lines (A_{31}-A_2) are connected to memory modules.
  - Low order two address lines (A_1 and A_0) are used to select one of the modules.

15

15

- How are the addresses mapped to memory modules?
  - ***Module M0***: 0, 4, 8, 12, 16, 20, 24, …
  - ***Module M1***: 1, 5, 9, 13, 17, 21, 25, …
  - ***Module M2***: 2, 6, 10, 14, 18, 22, 26, …
  - ***Module M3***: 3, 7, 11, 15, 19, 23, 27, …
- Memory addresses are *interleaved* across memory modules.
- What we can gain from this mapping?
  - Consecutive addresses are mapped to consecutive modules.
  - Possible to access four consecutive words in the same cycle, if all four modules are enabled simultaneously.

16

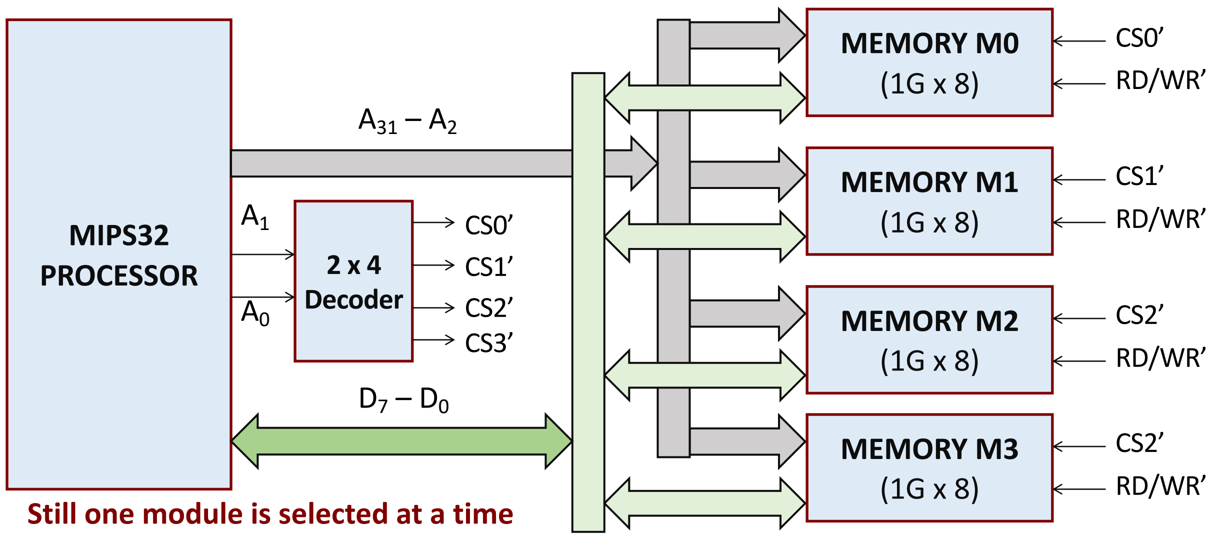
16

- Motivation for word alignment in MIPS32 data words.
  - 32-bit words start from a memory address that is divisible by 4.
    - Corresponding byte addresses are (0, 1, 2, 3), (4, 5, 6, 7), (8, 9, 10, 11), (12, 13, 14, 15), etc. → *last two bits of the addresses are 00, 01, 10 and 11*.
    - Possible to transfer all the four bytes of a word in a *single memory cycle*.
  - What happens if a word is not aligned?
    - Say: (1, 2, 3, 4) or (2, 3, 4, 5) or (3, 4, 5, 6).
    - Two of the bytes will be mapped to the same memory module.
    - Hence the word cannot be transferred in a single memory cycle.

2 memory cycles required

17

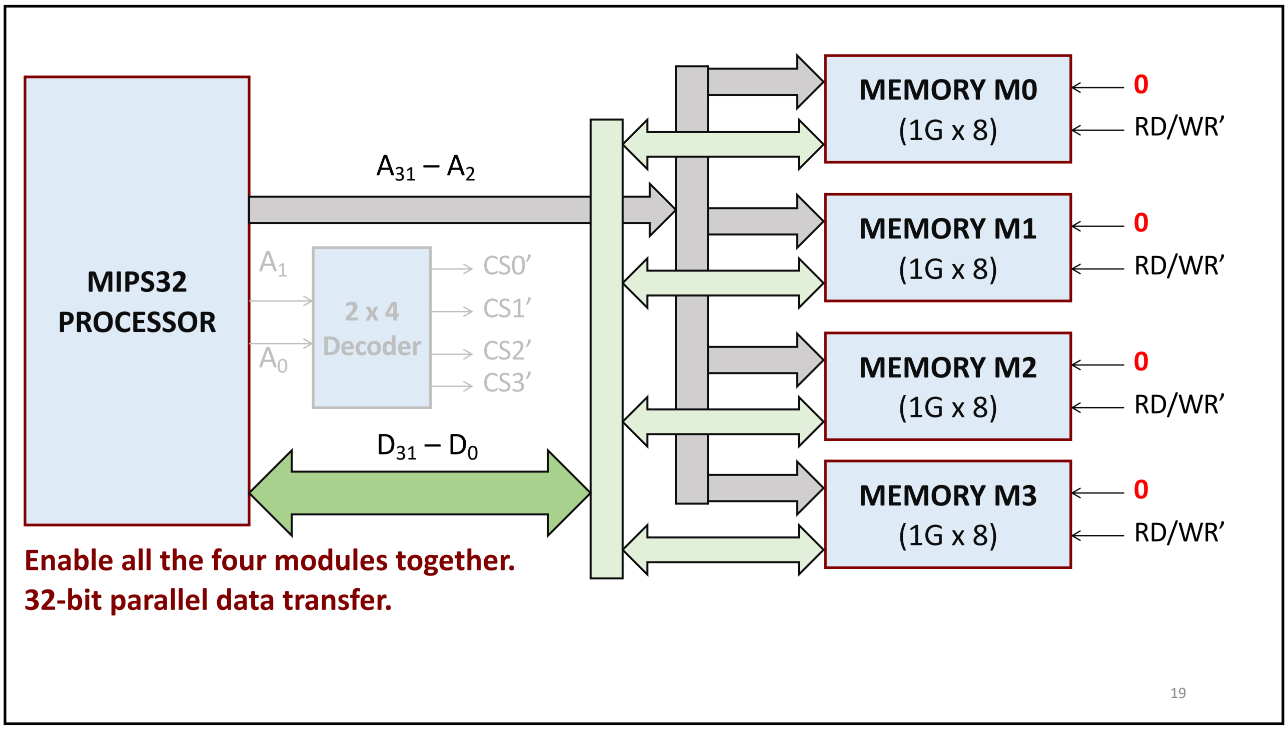
17



Still one module is selected at a time :: 8 bits data transfer per cycle.

18

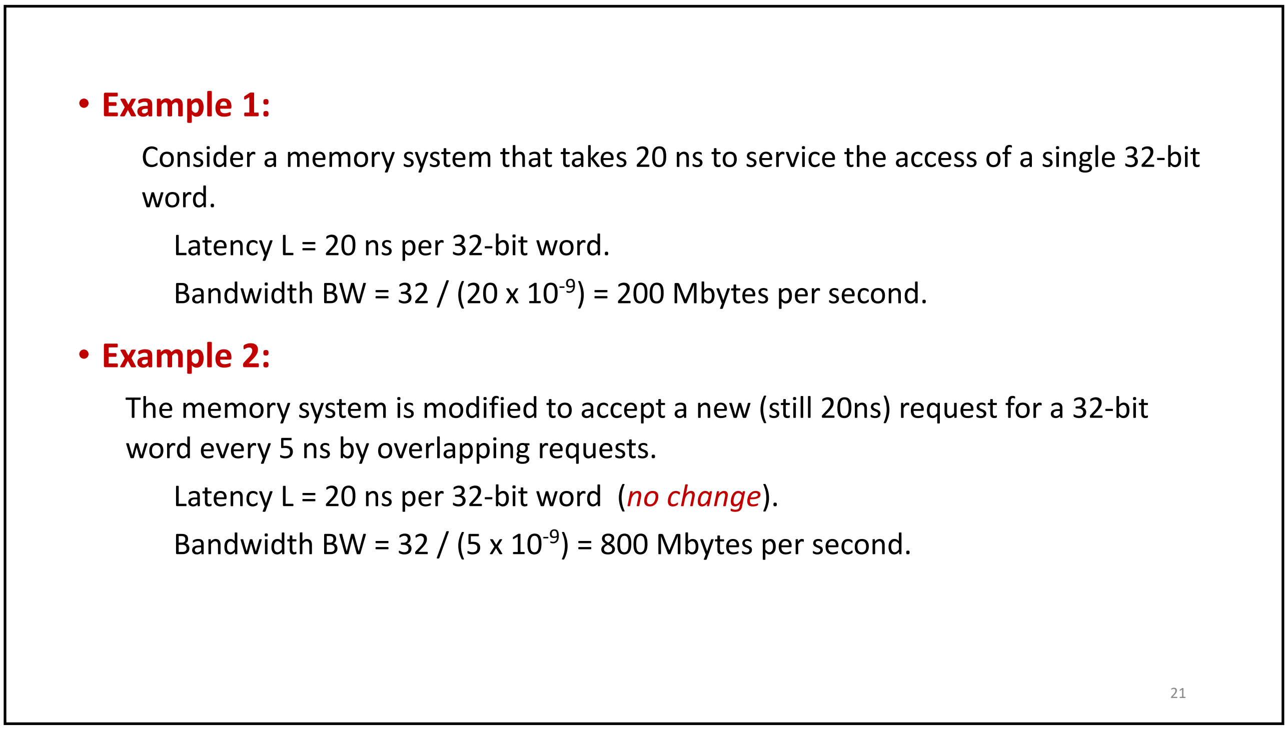
18



Enable all the four modules together.
32-bit parallel data transfer.

Memory Latency and Bandwidth

- **Memory Latency:**
  - The delay from the issue of a memory read request to the first byte of data becoming available.
- **Memory Bandwidth:**
  - The maximum number of bytes that can be transferred between the processor and the memory system per unit time.



- **Example 1:**

  Consider a memory system that takes 20 ns to service the access of a single 32-bit word.

  Latency L = 20 ns per 32-bit word.

  Bandwidth BW = 32 / (20 x 10^{-9}) = 200 Mbytes per second.

- **Example 2:**

  The memory system is modified to accept a new (still 20ns) request for a 32-bit word every 5 ns by overlapping requests.

  Latency L = 20 ns per 32-bit word (*no change*).

  Bandwidth BW = 32 / (5 x 10^{-9}) = 800 Mbytes per second.

21

21