

Register File Design and Memory Design

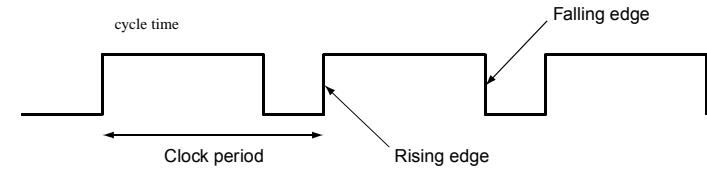
Presentation E

Slides by Gojko Babić

07/11/2005

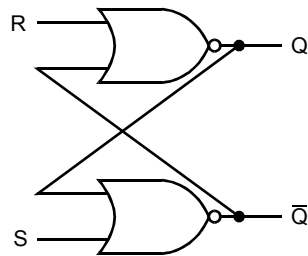
State Elements

- Unclocked vs. Clocked
- Clocks used in synchronous logic
 - when should an element that contains state be updated?



An unclocked state element

- The set-reset latch
 - output depends on present inputs and also on past inputs



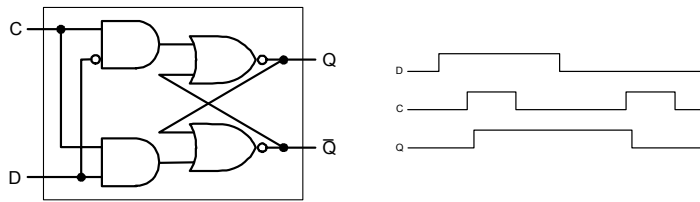
Latches and Flip-flops

- Output is equal to the stored value inside the element (don't need to ask for permission to look at the value)
- Change of state (value) is based on the clock
- Latches: whenever the inputs change, and the clock is asserted
- Flip-flop: state changes only on a clock edge (edge-triggered methodology)

A clocking methodology defines when signals can be read and written
– wouldn't want to read a signal at the same time it was being written

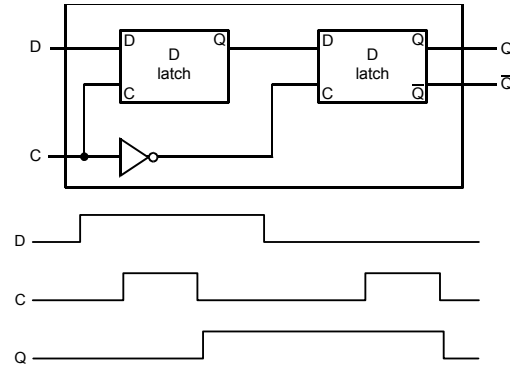
D-latch

- Two inputs:
 - the data value to be stored (D)
 - the clock signal (C) indicating when to read & store D
- Two outputs:
 - the value of the internal state (Q) and it's complement



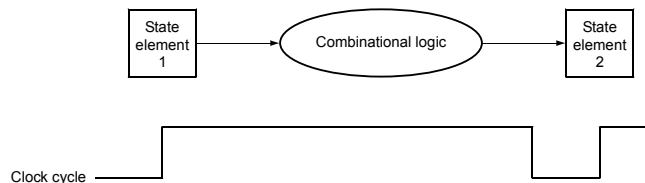
D flip-flop

- Output changes only on the clock edge



Our Implementation

- An edge triggered methodology
- Typical execution:
 - read contents of some state elements,
 - send values through some combinational logic
 - write results to one or more state elements



Register File

- The register file includes 32 32-bit registers, as it is needed for 32 general purpose registers of MIPS architecture

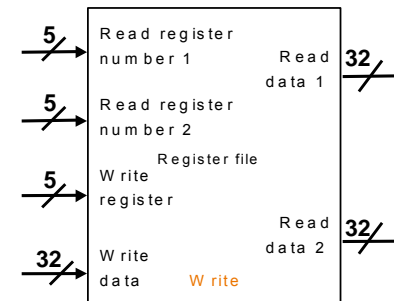


Figure B.8.7

- This register file makes possible to simultaneously read from two registers and write into one register as it is appropriate for MIPS processor.

Register File Functioning

- The given register file functions as follows:
 - any value provided on 5-line Read register number 1 port makes that content of corresponding register is provided on 32-line Read data 1 port
 - any value provided on 5-line Read register number 2 port makes that content of corresponding register is provided on 32-line Read data 2 port
 - on the falling edge of write line, values that appear on 32-bit Write data port are written into the register with the number on 5-line Write register port.
- Note that requirements for set-up time (and hold time) also apply here.

Register File Design: Read Part

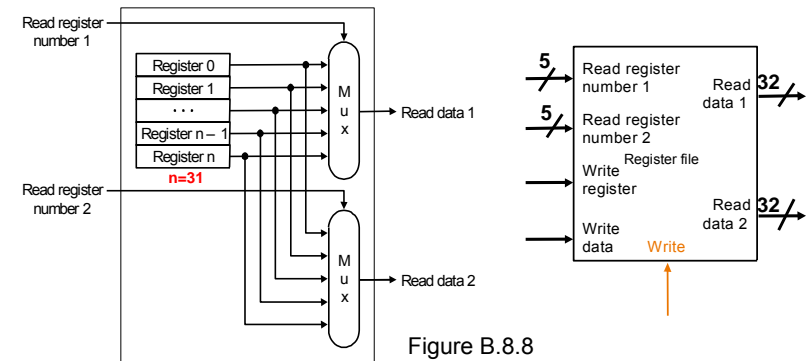


Figure B.8.8

This is a design at a level of register and complex multiplexer.

Register File Design: Write Part

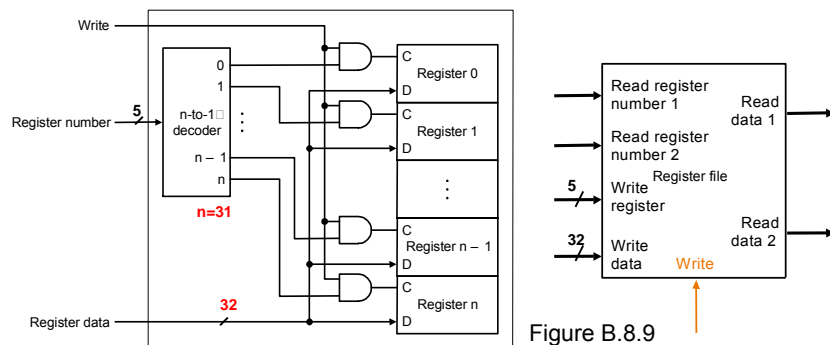


Figure B.8.9

This is a design at a level of register and decoder.

We are now going to design the MIPS register file at a level of flip-flop, basic multiplexer and decoder.

Introduction to Memory Design

- Main memory is built in one of two technologies:
 - SRAM - Static Random Access Memory
 - DRAM - Dynamic Random Access Memory
- A memory is normally built using a number of memory chips.
- Memory chips have specific configurations given as a product of two number, e.g.
 - 128K*1** - 128K addressable locations with 1 bit in each location, i.e. width of read/write operations is 1 bit
 - 16K*8** - 16K addressable locations with 8 bits in each location, i.e. width of read/write operations is 8 bits
- Notice that two chips above accommodate identical number of bits (128K bits).
- Both memories are *volatile*.

SRAM and DRAM: 1 Bit Memory Cell

- In SRAM technology, a three-state D-latch is the basic building block, i.e. basic memory cell. Internally, a D-latch can have a state corresponding to 0 or 1.
- In DRAM technology, the basic memory cell is built around one capacitor coupled with one transistor. The value in the cell is stored as a charge. A charge can not be stored indefinitely and DRAM chips must be periodically refreshed. Since charges can be kept for several msec, 1-2% of time is used for refreshing.

DRAM Technology Characteristics

Since 1975, the main memory is composed of semiconductor DRAM's (Dynamic Random Access Memory). DRAM chip capacity had been growing at rate of about 4 times every three years, while lately growth slowed down to 2 times every two years. Currently, maximum DRAM chip capacity is 512M bits with an access time in the range 40-60 nsec and a cycle time of about 80 nsec. Access time & cycle time are two measures of memory latency:

- access time – the time between a read is requested and when the desired content arrives,
- cycle time – the minimum time between two memory requests.

For DRAM technology cycle time is longer than access time. For SRAM technology access time and cycle time are identical.

Growth of Capacity per DRAM chip

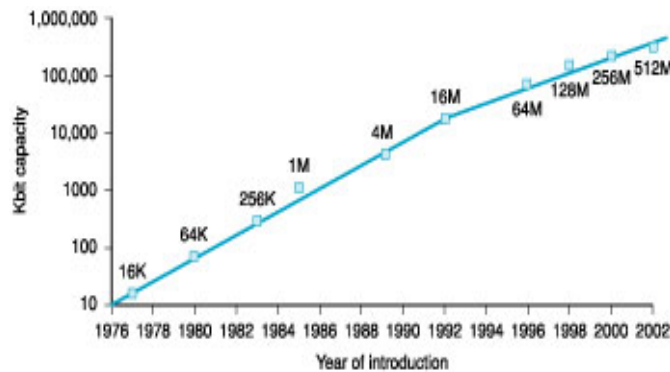
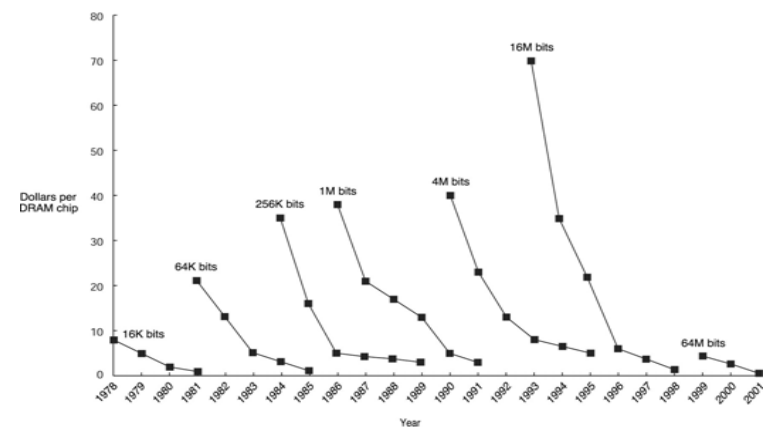


Figure 1.13

Prices of Six Generations of DRAMs



© 2003 Elsevier Science (USA). All rights reserved.

SRAM Technology Characteristics

- **SRAM** – **S**tatic **R**andom **A**ccess **M**emory technology is normally used for caches.
- In comparable technologies, SRAM cycle time is about 8 to 16 times faster than DRAM, e.g. currently 0.5-5 nsec.
- But, SRAM chip capacity (as well as density) is roughly 4 to 8 times less than that of DRAM
- Also SRAM is more expensive, e.g. 1GB in 2004 \$4,000 – \$10,000 for SRAM and \$100 – \$200 for DRAM.
- In addition, SRAM chips have higher power consumption and power dissipation than DRAM chips.
- Thus SRAM designs are concerned with speed, while in DRAM designs the emphasis is on cost per bit and capacity.

g. babic

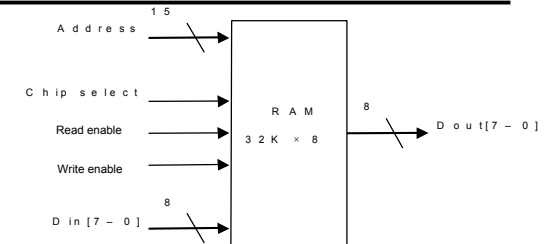
Presentation E

17

Memory Chip Functioning

Example: 32K×8 chip

- read and write operations are 8 bits wide
- there are 32K addressable locations



- Functioning of memory chip:
 - CS (**Chip select**) has to be set for either reading or writing
 - R (**Read enable**)=0 & W (**Write enable**)=0 → chip is not being accessed
 - R=0 and W=1 → write values at **Din** lines into the chip address at **Address** lines
 - R=1 and W=0 → read into **Dout** lines values from the chip address at **Address** lines
 - R=1 and W=1 → not allowed

Two designs of memory chip:

- **Basic structure design**
- **Typical organization design**

g. babic

Presentation E

18

Basic Structure Design of 4×2 SRAM Chip

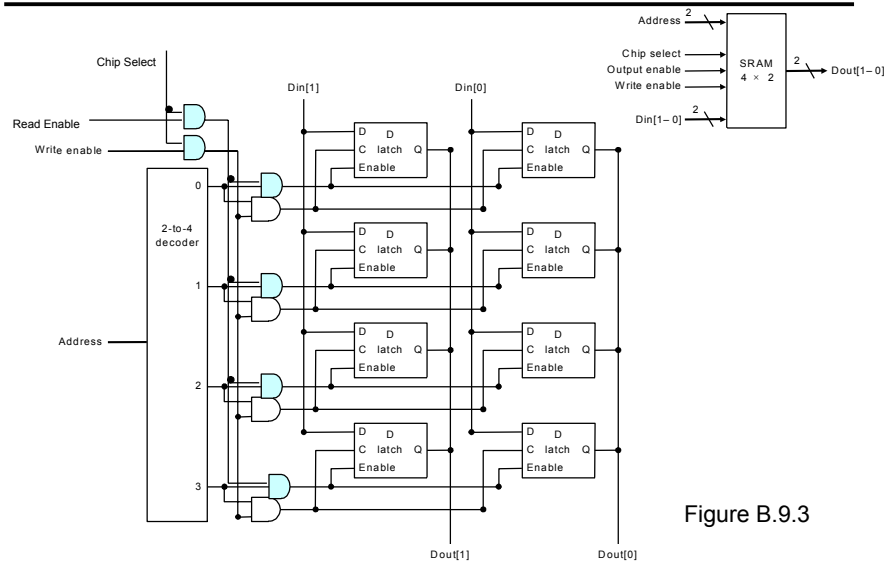


Figure B.9.3

g. babic

Presentation E

19

Design of Memory Chip Basic Structure

- The design of the basic structure of SRAM chip uses some ideas from the register file design e.g. the write parts in two designs are identical. The main differences are in read part design. In the memory chip with the usage of three-state D-latches a multiplexer is eliminated. E.g. for 32K×8 SRAM chip, a multiplexer with 32K inputs each input having 8 lines would be needed.
- But for design of the basic structure of SRAM chip, we still need a very large decoder. E.g. for 32K×8 SRAM chip, a decoder with 15 input lines (that is not so bad) and 32K output lines (that is bad) is required. Typical organization of SRAM uses two level decoding that eliminates need for that very large decoder.

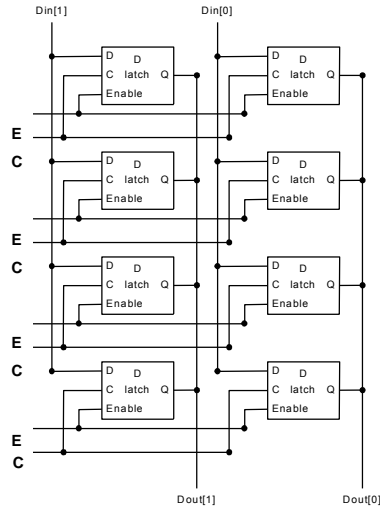
g. babic

Presentation E

20

4x2 Array of D- Latches

The next example will be using 512x64 array of D-latches.

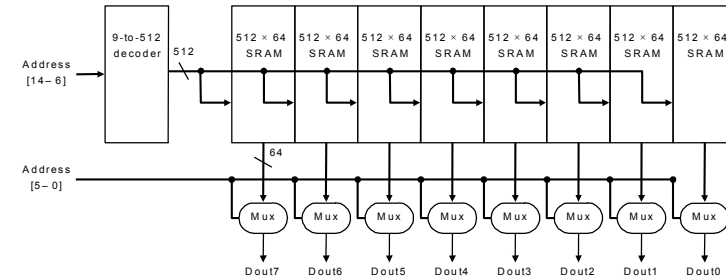


g. babic

21

Typical Organization Design

- Example:** Design (read part only) a typical organization (i.e. two level decoding design) of 32Kx8 SRAM chip that uses 512*64 arrays of D-latches.
- Note:** Arrays used have to have a bit capacity equal to a number of addressable locations in the chip, e.g. in this example that condition is satisfied since $512 \times 64 = 32K$. A number of arrays used should be equal to the number of bits in each memory location.



DRAM memory chip would have similar design.

g. babic

Presentation E

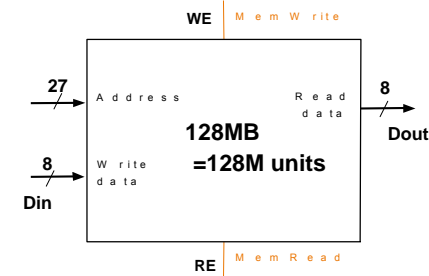
22

Main Memory Specification

- A memory has identical inputs and outputs as memory chips, except that CS does not exist. But the specification of a memory should include:
 - memory capacity (usually in bytes),
 - memory addressability, i.e. smallest unit that has its address,
 - width of read/write operations, i.e. a number of bits that can be read or written from/to memory.
- Operations on memory: reading from memory and writing into memory;
 - RE=0 and WE=0 → memory is not being accesses
 - RE=0 and WE=1 → writing into memory
 - RE=1 and WE=0 → reading from memory
 - RE=1 and WE=1 → not allowed

Main Memory Specification: Example 1

- Provide inputs and outputs of 128MByte memory with 8-bit read/write operations and byte addressability.
Note that 8-bit read/write operations requires byte addressability.



g. babic

Presentation E

23

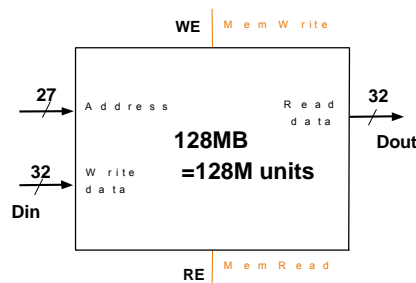
g. babic

Presentation E

24

Main Memory Specification: Example 2

- Provide inputs and outputs of 128MByte memory with 32-bit read/write operations and byte addressability.



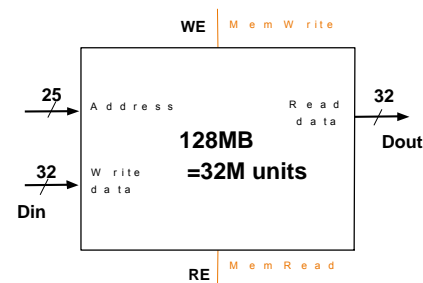
g. babic

Presentation E

25

Main Memory Specification: Example 3

- Provide inputs and outputs of 128MByte memory with 32-bit read/write operations and 32-bit addressability.



g. babic

Presentation E

26

Steps in Memory Design

- determine inputs and outputs for a memory to be design and memory chips that are being used;
- determine number of memory chips needed;
- determine number of memory chips in each set; a number of Dout and/or Din lines in the set should be identical to number of Dout and/or Din lines in the memory;
- determine number of sets;
- allocate sufficient number of memory address lines to select each of sets: those are the most significant address lines
- allocate next set of memory address lines as inputs to all memory chip address lines;
- If the number of bits in read/write operations equals the number of bits in addressability, then all memory address lines are used up in steps 5 and 6.
- When condition in 7 is not satisfied → go to slide 26
- Connect Din and Dout lines of memory and chips

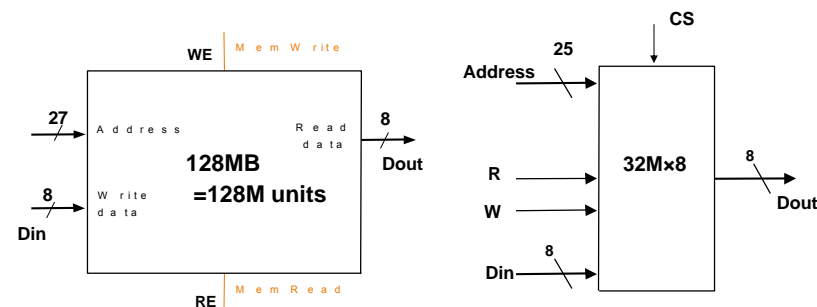
g. babic

Presentation E

27

Memory Design: Example 1

- Design 128MByte memory using 32M*8 chips, with 8-bit read/write operations and byte addressability.



Number of chips needed: 4

Number of chips per set: 1

Number of sets: 4

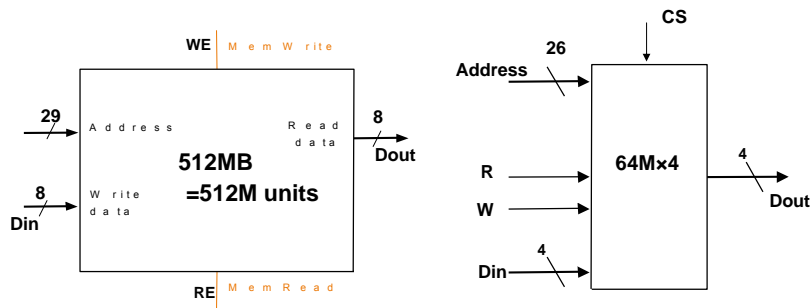
g. babic

Presentation E

28

Memory Design: Example 2

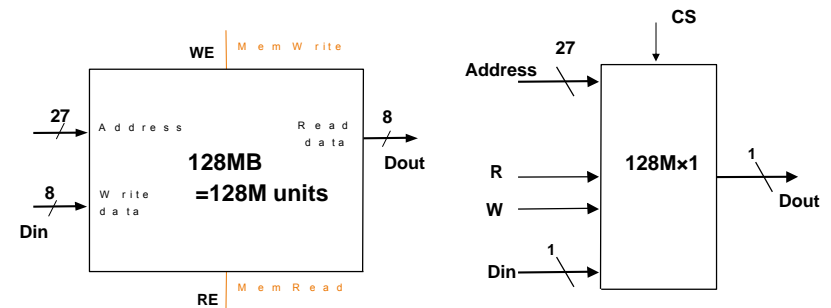
- Design 512MByte memory using 64M*4 chips, with 8-bit read/write operations and byte addressability.



Number of chips needed: 16
 Number of chips per set: 2
 Number of sets: 8

Memory Design: Example 3

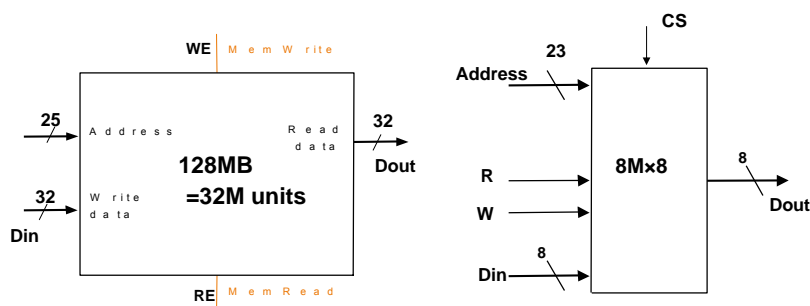
- Design 128MByte memory using 128M*1 chips, with 8-bit read/write operations and byte addressability.



Number of chips needed: 8
 Number of chips per set: 8
 Number of sets: 1

Memory Design: Example 4

- Design 128MByte memory using 8M*8 chips, with 32-bit read/write operations and 32-bit addressability.



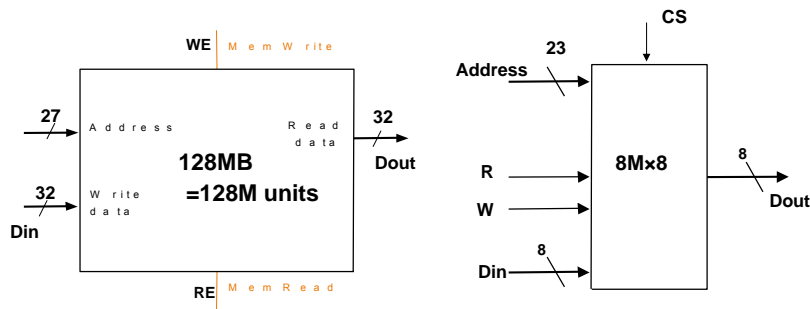
Number of chips needed: 16
 Number of chips per set: 4
 Number of sets: 4

Steps in Memory Design (continued)

- This is the second part of step 7 in “Steps in Memory Design” slide:
 - when the number of bits in read/write operations is greater than the number of bits in addressability, then some lowest order memory address lines are not used
 - if the width of read/write operations doubles that of addressability then the least significant memory line is unused,
 - if the width of read/read operations is 4 times greater than the number of bits in addressability then the two least significant memory lines are unused, etc.
- Note, it doesn't make sense to have the width of read/write operations smaller than addressability.

Memory Design: Example 5

- Design 128MByte memory using 8M*8 chips, with 32-bit read/write operations and 8-bit (byte) addressability.



Number of chips needed: 16

Number of chips per set: 4

Number of sets: 4