Memory System Design

The lecture describes simple memory systems.
After this lecture you should be able to: answer short questions about different memory types and memory terminology; identify read/write timing specifications from a timing diagram (as for flip-flops); draw a schematic showing how memory ICs are connected to increase the data bus size and the total amount of memory; design (combinational) address decoding logic to place memory at specific address ranges.

Memory Terminology

memory cell the structure that stores a single bit within a memory. Typically a capacitor (for DRAM or flash) or a latch (for SRAM).

word the bits read or written to a memory in parallel (at the same time).

byte eight bits.

rows and columns a memory can be considered to be an array of bits arranged in rows and columns. Each row is an address, each column is one bit of a word.

address the location of data stored in a memory. The address bus is the set of signals used to select the location in the memory to read or write from/to. An n-bit address can select one of 2^n words.

read retrieval of a word from a memory.

write storage of a word in a memory.

data bus the signals used to read or write the value of one word from/to the memory.

address bus the signals used to select the word to be read/written from/to the memory.

volatile memory memory whose contents are lost if power is removed.

non-volatile memory memory whose contents are *not* lost if power is removed.

RAM (random-access memory). Another name for volatile memory.

ROM (read-only memory). Another name for non-volatile memory.

PROM, EPROM, EEPROM types of ROM can be programmed once (PROM), erased with UV light (EPROM), or electrically erased one word at a time (EEPROM).

"flash" memory a type of EEPROM where many words must be electrically erased simultaneously before individual words can be reprogrammed.

static RAM (SRAM) RAM that stores data in a latch. The contents are lost when power is removed.

dynamic RAM (DRAM) RAM that stores data in capacitors. The contents are lost if the charge on the capacitor is not "refreshed" periodically.

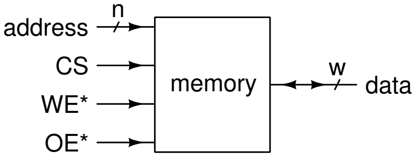
cache memory fast memory used as temporary storage between the main memory and CPU.

mass storage non-volatile memory, typically large and slow. Examples are hard drives and flash memory.

kB, MB, GB for memories, a kilobyte means 2^{10} (1024) bytes instead of 10^3 (1000). Similarly a MB is 2^{20} bytes and a GB is 2^{30} bytes. Often (but not always) a lower-case 'b' is used to denote bits (e.g. kb is used for a kilobit).

Memory Interfaces

The interface of a memory IC has the following signals:



address bus this n-bit bus selects one of 2^n "rows" in the memory array

data bus this w-bit (word size) bus has the value of the data read from, or to be written to, the memory. This bus can be bidirectional (controlled by OE), or there can be separate read and write data buses.

CS "Chip Select." This signal must be asserted to enable either a read or a write. Sometimes called CE (chip enable), particularly on ICs that are not memories.

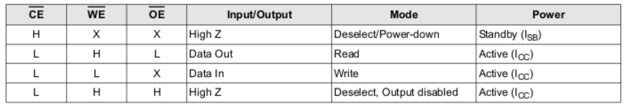
OE "Output Enable." When asserted, the data bus acts as an output; otherwise the data bus is an input.

WE "Write Enable." When asserted, the value on the data bus is written to the memory location present on the address bus. The write operation completes when the signal is de-asserted. This signal is similar to the clock signal for a flip-flop.

OE and WE are often active-low ($\overline{\text{CS}}$, $\overline{\text{OE}}$, or $\overline{\text{WE}}$). This enables multiple devices to drive these buses using open-collector (open drain) outputs.

The following truth table[1] explains the different states of the memory:

Truth Table



$\overline{\text{CE}}$	$\overline{\text{WE}}$	$\overline{\text{OE}}$	Input/Output	Mode	Power
H	X	X	High Z	Deselect/Power-down	Standby (I_{SB})
L	H	L	Data Out	Read	Active (I_{CC})
L	L	X	Data In	Write	Active (I_{CC})
L	H	H	High Z	Deselect, Output disabled	Active (I_{CC})

Shared Buses

Connecting more than one logic output together is not possible because this would create contention (different outputs driving the same net with different logic levels).

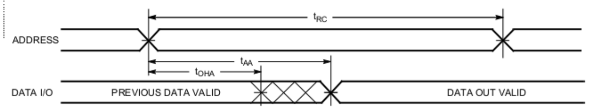
To connect the data buses of multiple memory devices we must ensure that only one device is enabled at a time. This is accomplished by using tri-state outputs.

Memory Read/Write Cycle Timing

The following SRAM timing diagrams are also taken from the Cypress CY7C1399 data sheet.

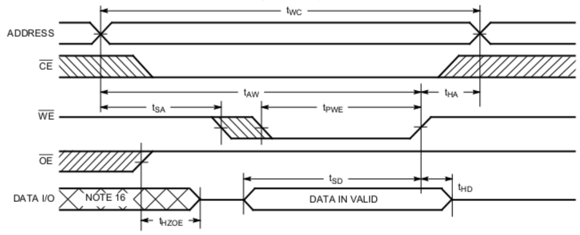
The first timing diagram is for a read cycle. The most important specification is the access times (t_{AA}). This is measured from when the address input changes to when the corresponding data is available on the data bus output.

Figure 5. Read Cycle No. 1 [10, 11]



The second timing diagram is for a write cycle. The important specifications here are the address setup time to write end, t_{AW}, address setup time to write start, t_{SA}, and data setup to write end t_{SD}, and the corresponding hold times t_{HD} and t_{HA} (which are both zero).

Figure 7. Write Cycle No. 1 ($\overline{\text{WE}}$ Controlled) [13, 14, 15]



For both types of cycles there will also be minimum cycle times, for example, t_{RC} and t_{WC} as well as minimum pulse widths such as t_{PWE}.

There are two timing diagrams because the write cycle can be terminated by either the $\overline{\text{WE}}$ or $\overline{\text{CE}}$ signals.

Exercise 1: Is t_{AW} a requirement or a guaranteed specification for this memory? How about the t_{AA} ?

Memory Arrays

We can combine memory ICs to increase the:

word width by using combining ICs in parallel. This allows us to increase the word width to any multiple of the IC word width. For example a 16-bit memory would require two 8-bit memories in parallel or four 4-bit memories. All of the IC's in such a "bank" are connected to the same control bus signals (CS, OE, WE).

[1] From the Cypress CY7C1399 SRAM data sheet.

amount of memory by combining these banks of memories we can increase the total amount of memory available. To avoid conflict, each bank must be enabled so it appears at non-overlapping memory locations. This means each bank has to have it's own CS signal. This is the function of address decoding, described below.

You can follow these rules for memory design:

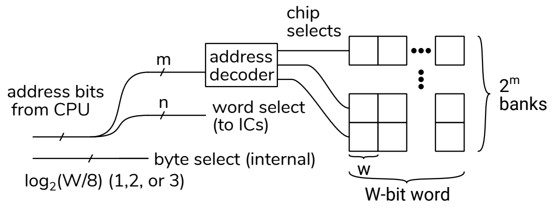
- The number of ICs in each bank is the memory word size (W) divided by the IC word size (w). For example, if we need $W = 32$ bits per word and have $w = 4$-bit memory ICs then we need $W/w = 32/4 = 8$ ICs per bank.

- The number of banks equal to the number of words in the memory divided by the number of words in each IC (2^n). For example, if we need 16k words using ICs with 4k words then we need 16k/4k = 4 banks.

- Addresses are always byte addresses so $\log_2(W/8) - 1$ of the least-significant address bits select a byte from the word and are not output by the CPU (1, 2 or 3 bits for 16-, 32- and 64-bit buses respectively).

- The next least-significant n bits are connected in parallel to each memory IC.

- m additional address bits select from 2^m contiguous memory banks.



Exercise 2: How many 256 k×8 memory IC's would be required to build a 1 M ×32 memory? What is the width of the data bus? How many address bus bits would be required from the CPU? Which of these would connect to the memory ICs? What address values could be placed on the address bus? How many chip-select lines would be required?

Memory Map

A memory map is a diagram showing the addresses of memory devices. The addresses increase (or decrease[2]) from top to bottom and the range is divided up into blocks. Each block is labelled with a range of addresses as well as the name of the device (memory or I/O port) that is enabled for that range of addresses.

Here is an example of a memory map from the MSP432 microcontroller datasheet:

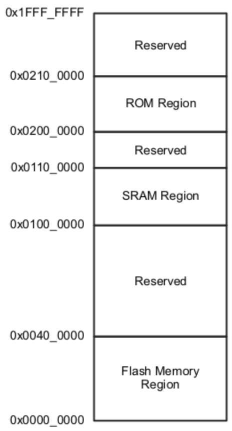


Figure 6-2. Code Zone Memory Map

Exercise 3: How large are the two lowest memory regions in the memory map above?

Address Decoding

The remaining bits of the address bus are used to select a specific bank of memory by enabling only one of the CS signals. The circuit that does this is called an address decoder. The design of the address decoder is what determines the location of a memory bank in the memory map.

Memories are placed at multiples of their size to simplify address decoding.

Exercise 4: If a CPU has a 32-bit address bus, how many bytes can it address? What range of addresses would correspond to the first 64 k Bytes? If this range of memory was to be implemented with 32-bit words, how many address bits would be required to select a byte within each word? How many bits would be required to select a 32-bit word within the 64 k range? How many bits are not directly connected to the memory ICs? What would they be used for?

[2] Unfortunately, there are different convention for the direction in which to draw the memory map (addresses can be shown increasing up or down).

Exercise 5: A 4k×16 memory is to be used in a system with a 20-bit address bus. This memory is to respond to addresses starting at 20’hf2000. Draw the memory map. Assuming the address signal is defined as `logic [19:0] a;` and the chip-select as `logic cs0 ;`, write the Verilog that would implement the chip-select signals cs0. Write the expression for a second chip-select, `cs1` that would enable a second 8 kBytes bank immediately above (at a higher address than) the first.