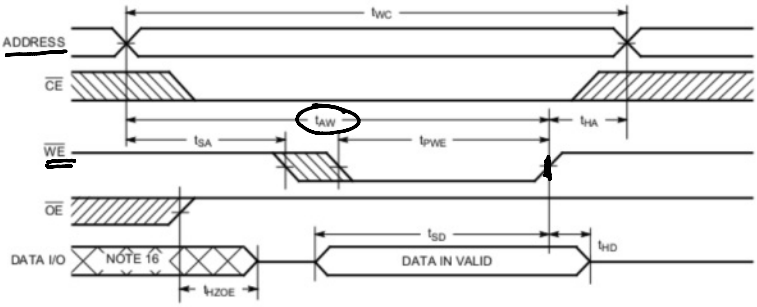
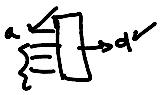
Memory System Design

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

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

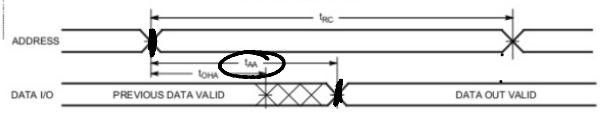




t_{AW} is requirement

t_{AA} is guaranteed response

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



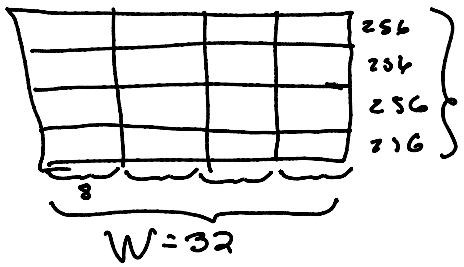
Exercise 2: How many 256 k×8 memory IC's would be required to build a 1 M ×32 memory? What is the width of the data bus? How many address bus bits would be required from the CPU? Which of these would connect to the memory ICs? What address values could be placed on the address bus? How many chip-select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

$$\frac{W}{w} \cdot \frac{2^{m+n}}{2^n}$$

$$= \frac{32}{8} \cdot \frac{1M}{256}$$

$$= 4 \cdot 4$$

$$= 16$$



$1M = 2^{20}$

$2^M = 4$

$m = 2$

$n = \log_2(256k)$

$= 18$ bits

need $1M \times 32 = 1M \times 4$ bytes $= 4$ MBytes.

so need total of $\log_2(4M)$ address bit

$= \log_2(4M) = 22$ bits

but i.s. 2 bits (A_0 & A_1) are used to select a byte from the 32 bits

so only need 20 bits ($A_2 \cdots A_{21}$) to select a word.

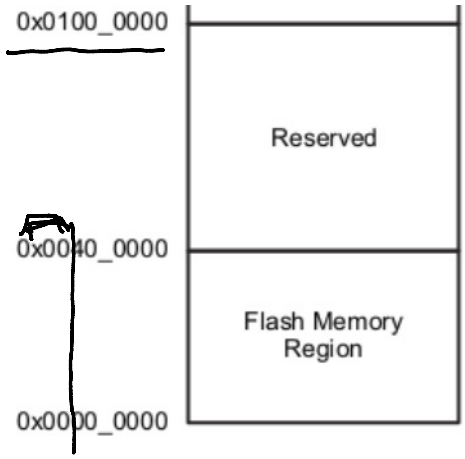
for each IC need to select 1 of 256 k words so need $\log_2(256k) = 18$ bits.

$256\,k = 2^8 \cdot 2^{10} = 2^{18}$

need $m = 18 + n = 2 + 2$ (for byte select)

$= 22$ bits

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



4
3
2
1
0

4 - 0 = 4

3 - 0 + 1 = 3+1

40 0000
- 00 0000
0x 40 00 00 ← number of bytes.

= 4'194'304.10 bytes

= 4096 kBytes.

= 4 MBytes.

0x 100 - 0000
- 0x 40 _ 0000

0x C00000 = 12'582'912.10

= 12 MBytes.

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?

$1\ GByte = 2^{30}$ bytes.

2^{32} bytes = 4 GBytes can be addressed.

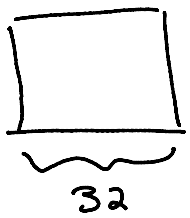
$0 - (64k-1)$ or 0000 → 65535

0x0000 → 0xFFFF

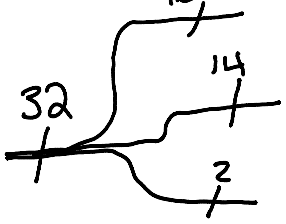
32 bits words. is 32/8 = 4 bytes per word.

need $\log_2(4) = 2$ bits to select a byte.

$n = \log_2(16k)$ → = 14 bits



$\frac{64k\ Bytes}{4\ bytes/word} = 16k$ words.

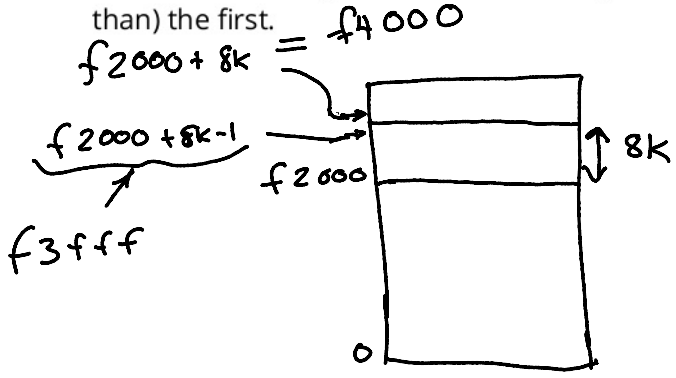


select the memory region where the memory appears in the memory map.

selects word

select a byte

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 signal, 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.



4k×16 = 4k × 16/8 = 8k bytes

→ assign cs0 = a >= 20'hf2000 && a<=20'hf3fff;

≥ f2000 = 1111 0010 0000 0000

≤ f3fff = 1111 0011 1111 1111

don't care

assign cs0 = a[19:13] == 7'b1111_001;

or

assign cs0 = a ==? {7'b1111_001, {13{1'b?}}}

```
module decoder
  ( input logic [19:0] a,
    output logic cs0, cs1, cs2 );

   assign cs0 = a >= 20'hf2000 && a <= 20'hf3fff ;
   assign cs1 = a[19:13] == 7'b1111_001 ;
   assign cs2 = a ==? { 7'b1111_001, {13{1'b?}} } ;

endmodule
```