

Memory Hierarchy, Address Decoding, and RAM Expansion

Memory Structure

- Memory Composition: Memory is built from the smallest unit (memory cell) up to larger structures.
 - A memory cell stores 1 bit
 - 8 cells = 1 byte
 - 4 bytes = 1 word (in MIPS)
 - Words can be grouped into blocks
- Access Pattern:
 - Building: from cell \rightarrow byte \rightarrow word \rightarrow block
 - Accessing: from block \rightarrow word \rightarrow byte \rightarrow cell (big to small)

Read: uses MUX

Write: In order to get to a specific location: uses decoders

Memory Unit: Size and Structure

- Represented as $2^k \times m$:
 - k = address size (number of bits needed for addressing)
 - m = word size (number of bits per word)
- A memory unit includes:
 - Read/write control line
 - Input and output data lines
 - select line (for chip selection in multi-chip setups)
- Addressing Modes
 - Byte Addressing (eg. in MIPS): each address points to a byte
 - Word Addressing: each address points to a word

RAM: Types and Characteristics

- Static RAM (SRAM)
 - Faster, more expensive
 - Used in cache memory
 - Doesn't need refreshing
 - Stores data using flip-flops
 - More reliable for frequent and fast access
- Dynamic RAM (DRAM)
 - Slower, cheaper
 - Used in most main memory systems
 - Stores data using capacitors that leak charge
 - Needs periodic refreshing (read then re-write the value to restore it)
 - Less expensive and denser than SRAM, allowing for higher capacity

Problem statement: Expanding RAM with an Array of RAM Chips

Given a RAM configuration $1K \times 16$, we want to:

- Increase the RAM capacity 10 times
- Access word 3079 (draw the circuit and indicate the settings to accommodate this task)

To increase the capacity 10 times, we'll put 10 chips together $\rightarrow C_0$ to C_9

To access word 3079, we consider that each RAM chip holds 1024 words, so to access word 0-1023, we need to access C_0 . To access word 1024, we need to access C_1

$\hookrightarrow 3079 = 1024 \times 3 + 7 \Rightarrow$ our target word is on C_3 with a word offset of 7

$$3079 = 3 \times 1024 + 7 = \underset{\substack{\downarrow \\ 2^9}}{2} \times 1024 + \underset{\substack{\downarrow \\ 2^{10}}}{1024} + \underset{\substack{\downarrow \\ 7}}{7}$$

Absolute Address:

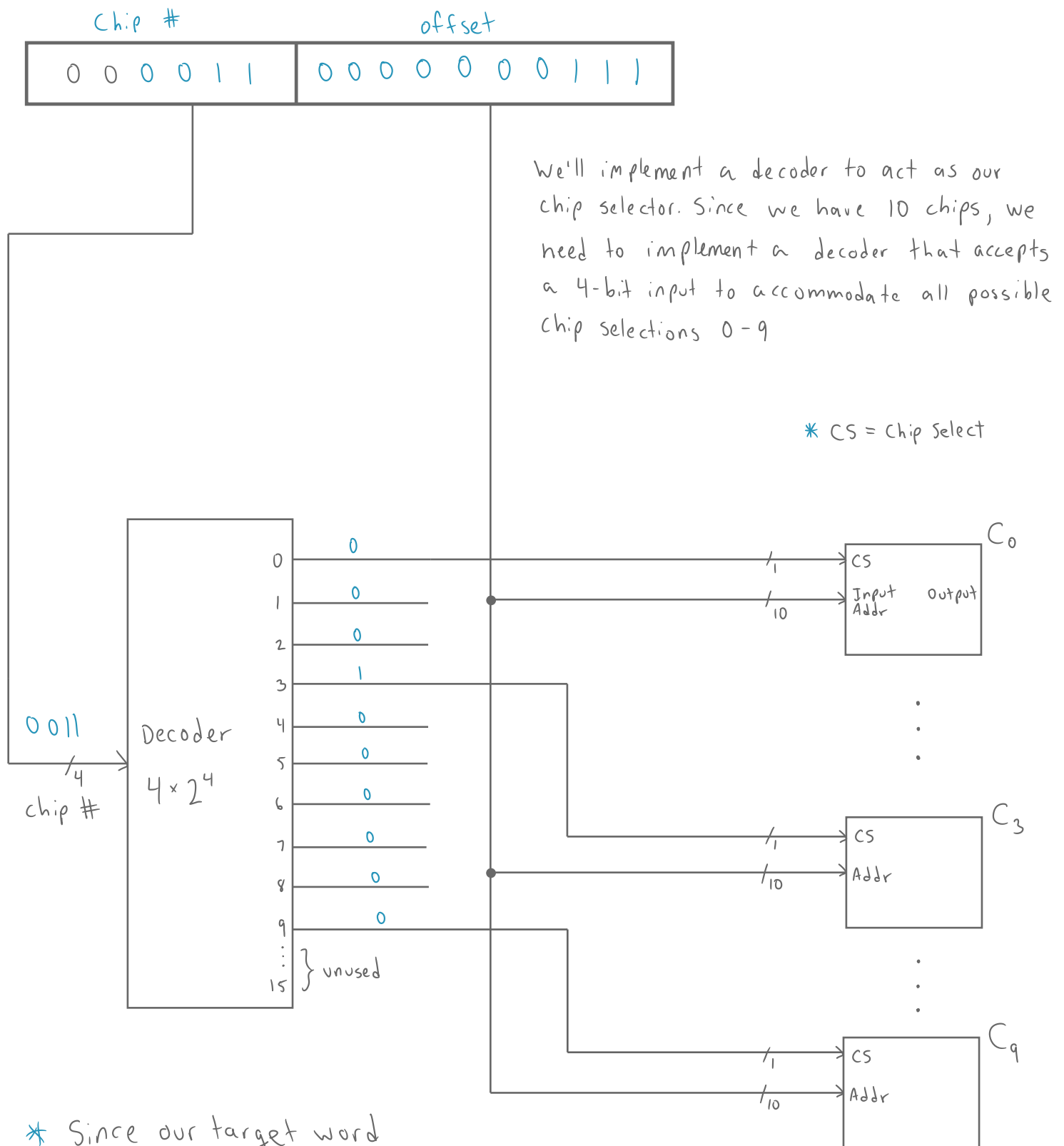
1	1	0	0	0	0	0	0	0	1	1	1
11	10	9	8	7	6	5	4	3	2	1	0

By examining our RAM config, $1K \times 16 = 2^{10} \times 16$, we can see that we need 10 bits to accommodate the offset. Since we have 16-bit words, we can assume a 16-bit address

Address Structure:



Our Address to access word 3079:



* Since our target word resides in C_3 , we'll use the decoder to select chip # 3, meaning the only decoder output that will be 1 will be line 3, which leads to C_3