Memory Basics

- We have already built registers using a series of Gated D latches joined by a WE (write enable) line
- How can we turn these into a memory?
- Start with the simple stuff: **address size** or **address width** is the number of bits in an address.
- **Address space** is the set of “addressable” locations, determined by the address width. It must be $\leq 2^k$ where $k$ is address width
- For example, a **k-bit** address means an address width of $k$ bits
- The **addressability** of a machine is the size of the most basic unit that can be named by an address
- A **Byte-addressable** machine is one where you can reference memory in 8-bit units (no smaller!)
- A **Word-addressable** machine can address memory at the unit of **word size**. This is typically the size of an integer. For example, a machine with a 64-bit word size (common today) that is word addressable would be able to access memory in units of 64-bits
- The chips in most of your computers are 64-bit (they have a 64-bit address width) and are byte addressable
- Byte addressability is the most common. This has historical ties to 8-bit ASCII (text) data
- If we have a k-bit address width and m-bit addressability, then we can have $m \times 2^k$ bits of memory (or $m \times 2^{(k-3)}$ bytes)
- **Important:** addressability and address width are independent!
- For word addressable machines, we’ll want to have an address width that is **smaller** than the addressability. Otherwise, we couldn’t store a memory address at a memory address!
- Alignment: If we have a byte addressable memory, we typically restrict addresses to be multiples of the addressability. For example, if we have 4-byte words, aligned accesses could be at 0, 4, 8, 12, etc. These are **word-aligned** addresses (they are aligned to the word size)
- Some chips will not let you do this, others will split things up for you, but it will be slow

**DRAM**

- See the drawings
Basic Memory

- Decoders
- Storage Cells
- Auxiliary Storage
- Muxes

\[ 2^2 \times 3 \text{- bit mem} \]

Addressability
Addr. Width

Row Decode

\[ I[2] \]
\[ I[1] \]
\[ I[0] \]

Data[2:0]

MUX!
DRAM

"Dynamic Random Access Memory"

The Cell

Word Line

Bit Line

Pass Transistor

Capacitor "Bucket o' Charge"

"Destructive" Reads

Needs Refresh Due to "Leaky Bucket" Capacitor

Memory Array

DRAM "Row"