# 07-ram-notes

## ***RAM***

---

## Agenda

---

- 0. Re-Orienting
- 1. Latches and Flip-flops
- 2. RAM (from the top)
- 3. RAM (from the bottom)
- 4. Dynamic RAM
- 5. If time.... flashy numbers!

---

## 0. Re-Orienting

---

electricity to gates

---

representing numbers as binary

---

building circuits that calculate functions on binary numbers

---

## 1. Latches and Flip-flops

---

---

putting "feedback" into circuits lets them remember state

---

the physical reality of the circuits can affect the logic design. (The abstraction barrier has holes in it!)

---

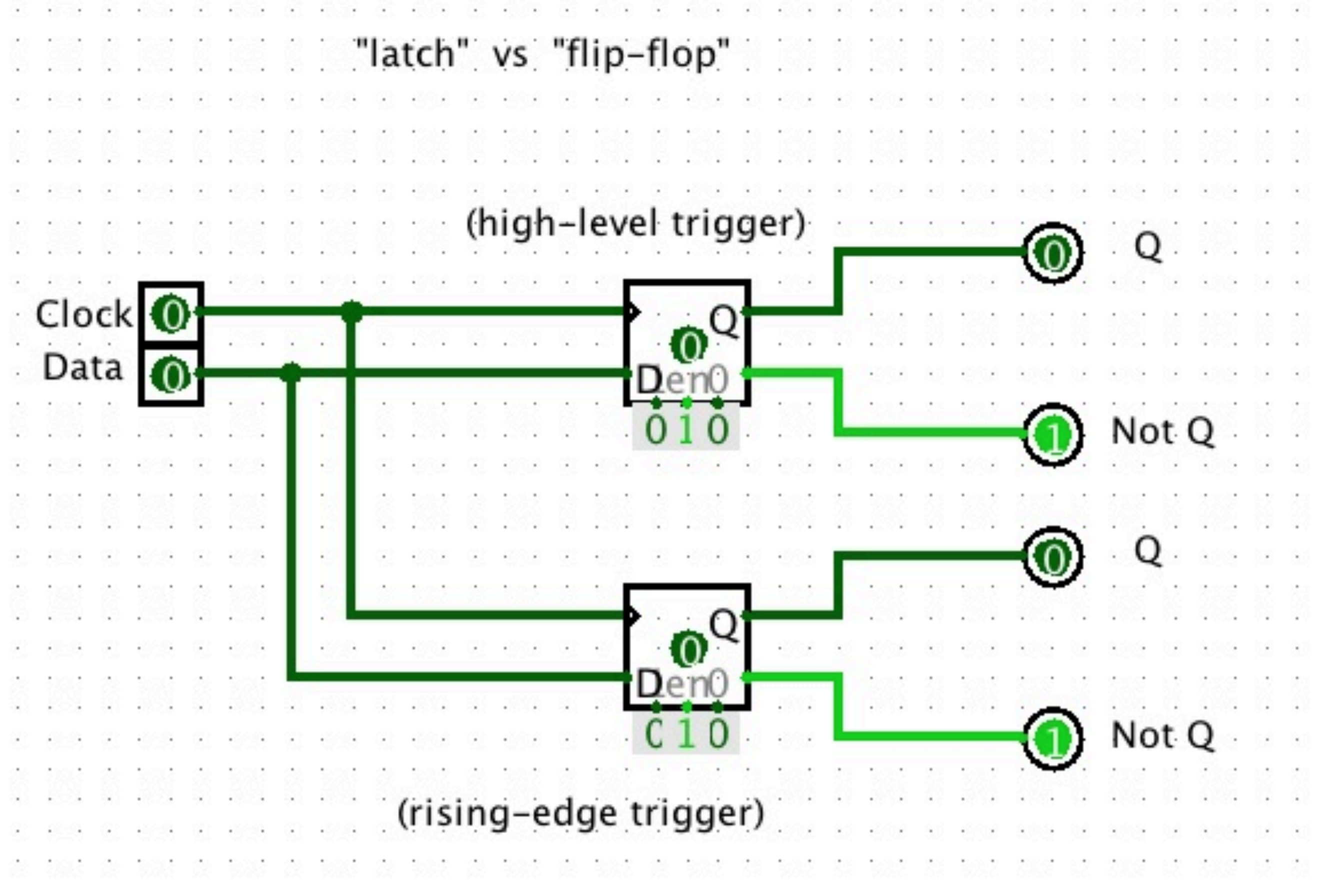
we can use this hole to turn edges into short pulses. (we didn't have time to get to this yesteday---but see the notes!)

---

...and thus build

- *latches,* which store a bit on a given clock *level*
- *flip-flops,* which store a bit a given clock *edge*

(Although there are other ways to build in edge-sensitivity)




Recall... "latchflop" in [ram.circ (https://ssl.cs.dartmouth.edu/~sws/cs51-s15/07-ram/demo/ram.zip)](https://ssl.cs.dartmouth.edu/~sws/cs51-s15/07-ram/demo/ram.zip)

---

## 2. RAM (from the top)

---

Flip-flops/latches REMEMBER things. In the context of computer architecture, probably their most natural application is MEMORY.

---

Reviewing (or introducing) the concept of computer memory:

- an array of "slots"
- each slot can hold some unit of data (e.g., a byte)
- each slot has an index number, called an **address**
- in a **read** operation, we specify an address, and the memory spits out the contents of that slot
- in a **write** operation, we specify an address and some new slot contents, and the memory stores those contents in that sot

---

As we discussed already....

- This model implicitly makes it possible to touch the memory slots in **any** order
- Hence the term **random access**
- (But the world now uses the term RAM to mean specifically random access memory that is **also** readable and writable.)

---

But in the real world, we have some ambiguity: what is the "slot" size?

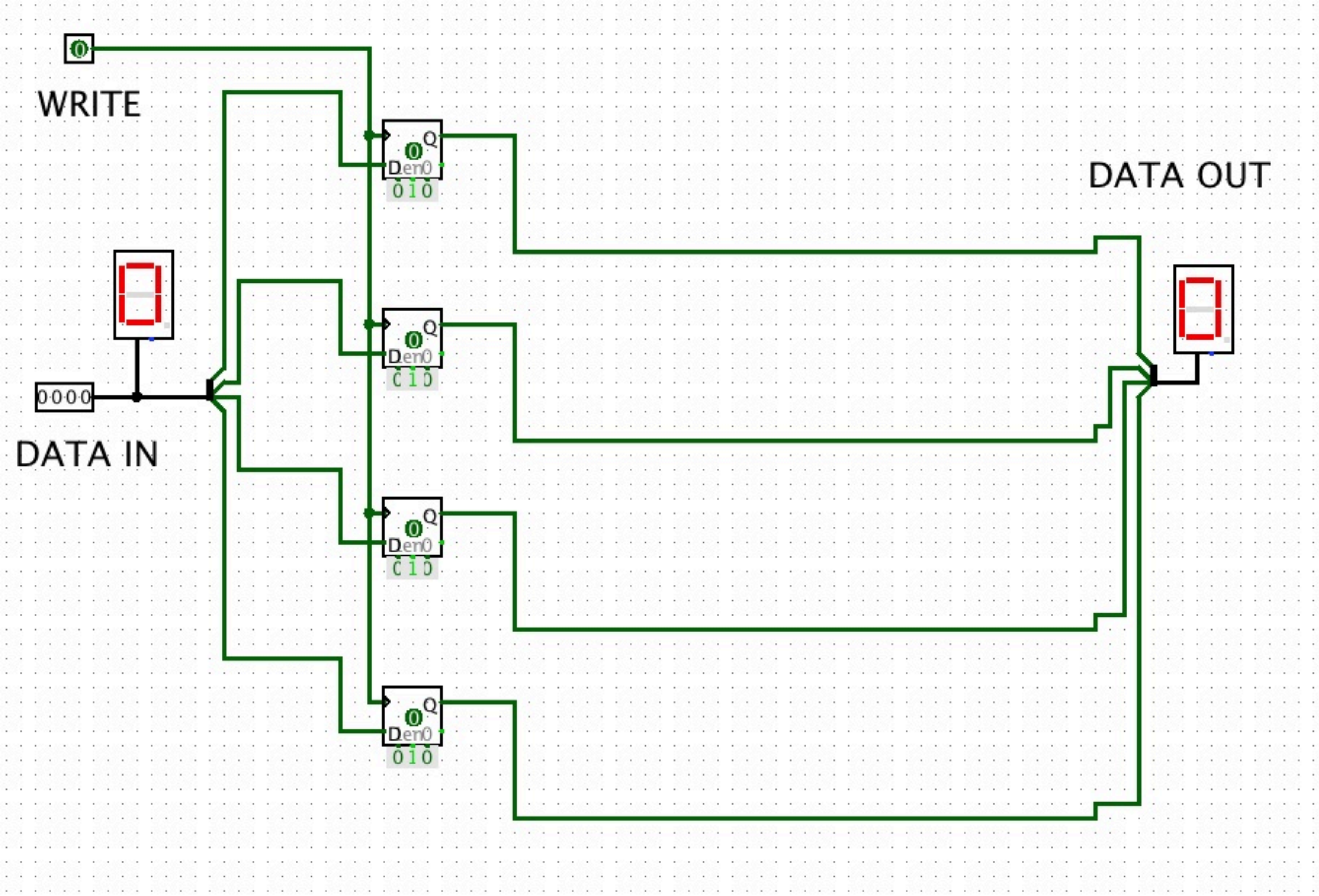
- generally, it's still a byte
- except the natural data size in standard machines is usually some larger **word**, such as 32 bits.
- so we still number the slots by bytes, but also permit operations to touch the **word** that lives at a specific address
  - that is... the word consisting of the bytes at address A, A+1, A+2, A+3
  - but what if this address is not a multiple of four?
- And as 32-bit machines give way to 64-bit machines, this will get even more fun.

---

## 3. RAM (from the bottom)

---

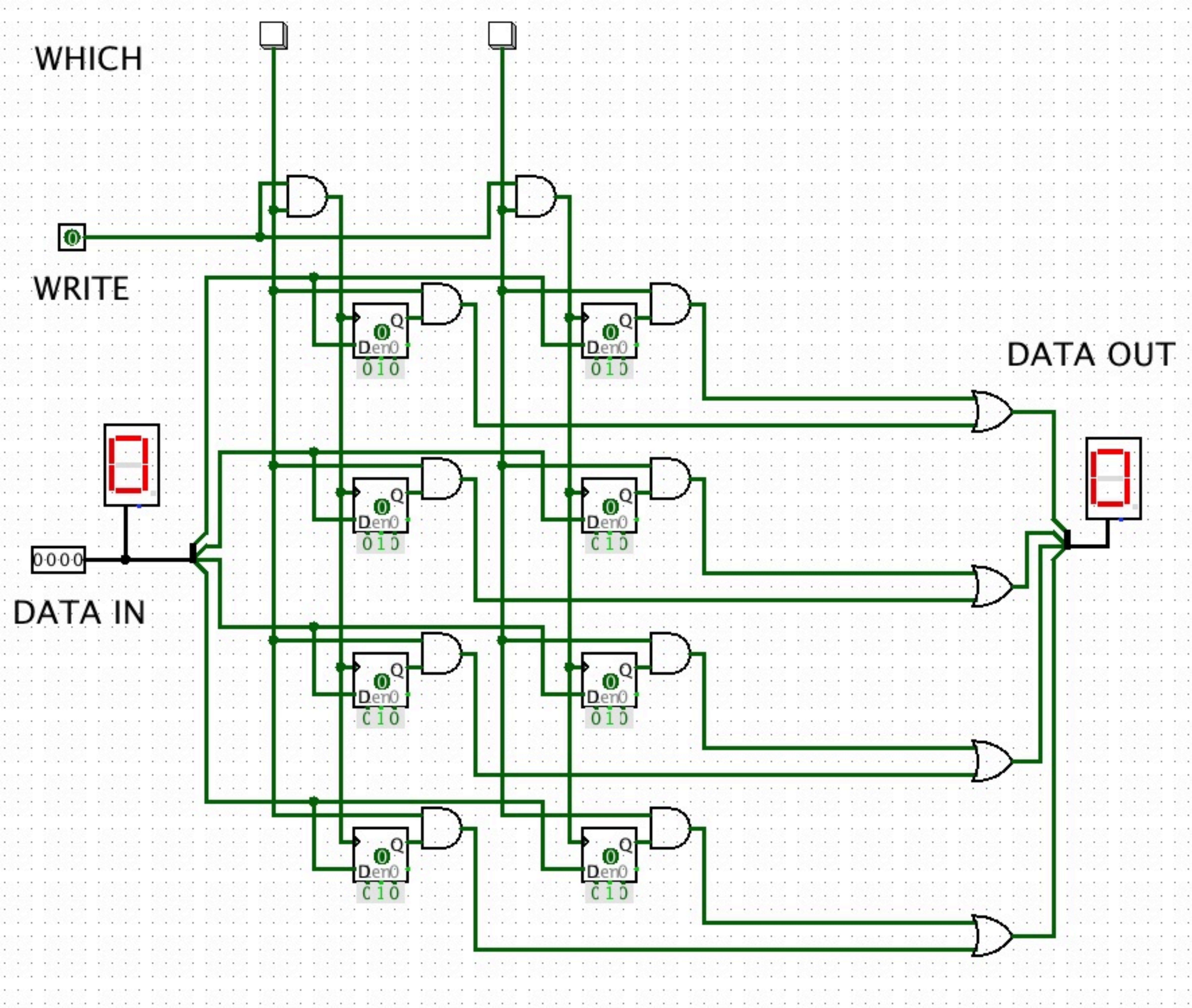
Four 1-bit D flip-flops:




("one-nibble" in ram.circ)

---

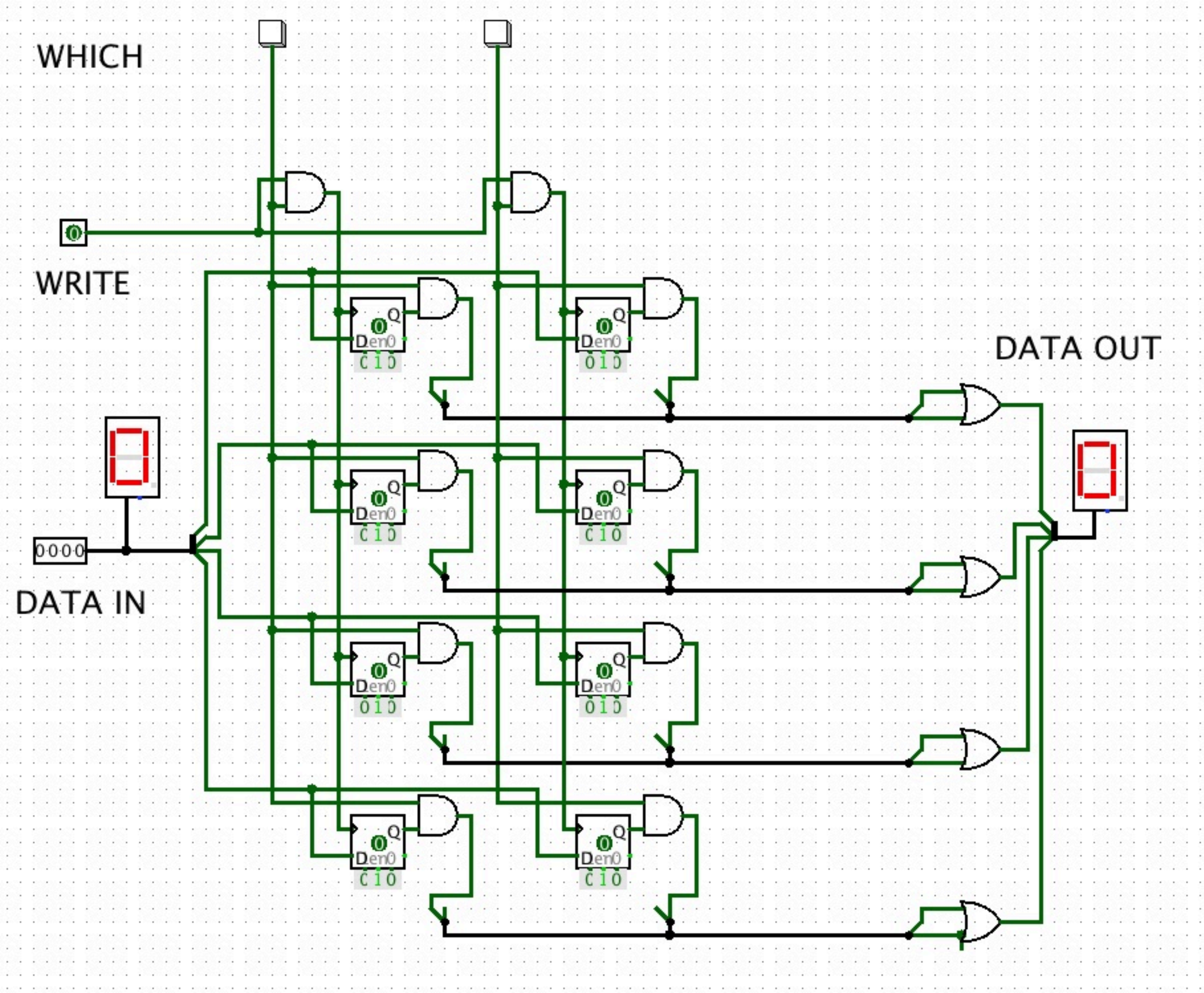
Let's add another, and the interesting one selectable




("two-nibbles" in ram.circ)

---

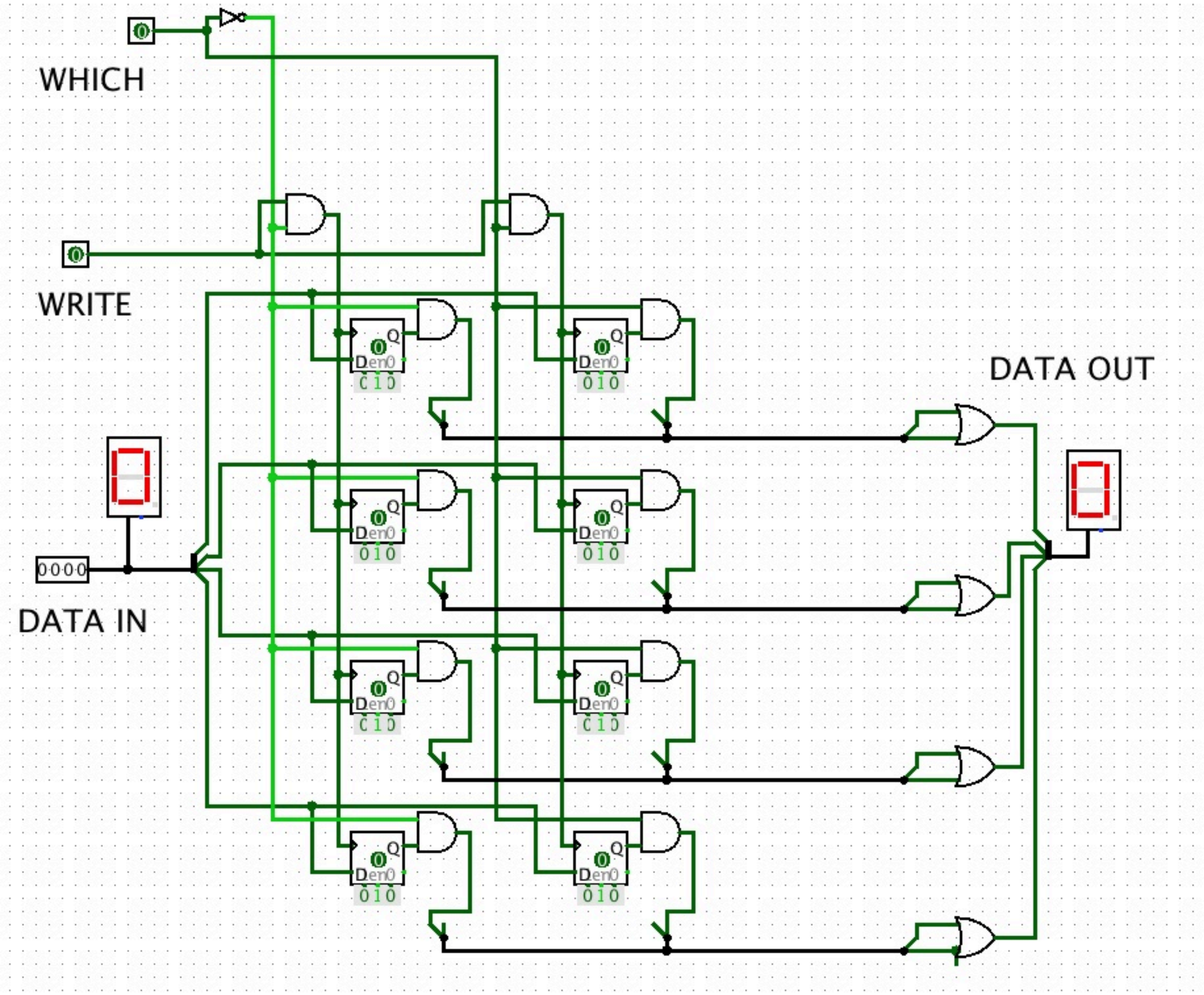
Let's make these parallel lines into a multi-bit wide line




("two-nibbles-bus" in ram.circ)

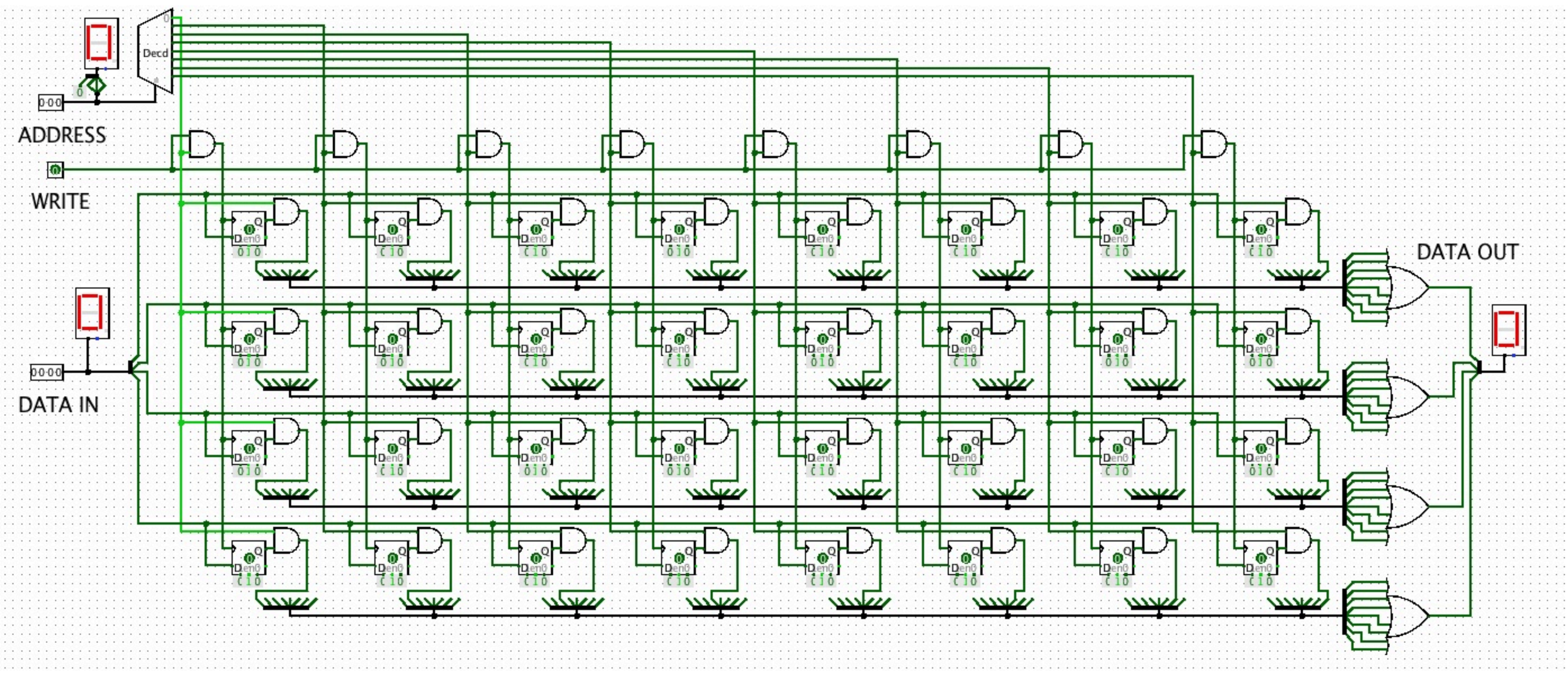
---

Let's get rid of the pushbuttons, and make it a 1-bit decoder




("two-nibbles-addr-bus" in ram.circ)

And generalize to three bits of address!




("main" in ram.circ)

Gosh, darn: RANDOM ACCESS MEMORY!

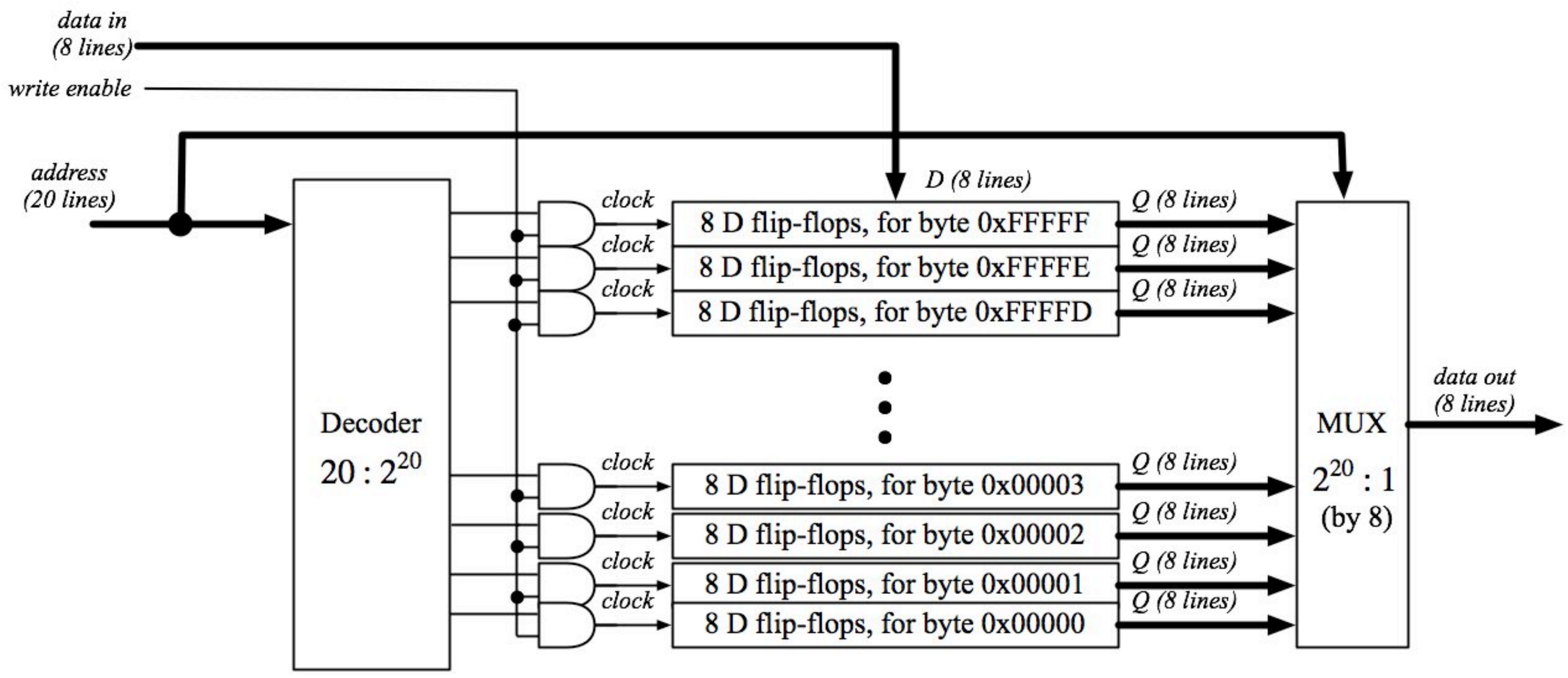
(This example uses the "column = word" approach, although I usually like to think of ROWS as words.)

do you understand all the pieces that are in there?

- the decoder
- the gating of the write-enable lines into the clock
- (although note that one may often want to have a master clock-in as well as a master enable)
- the marshaling of the data into the flip-flops
- gating the data out into the giant ORs
- the roles of the splitters in the collection of data out
- the role of the splitters taking the address lines into the address display

---

A megabyte of RAM




***1 megabyte of static RAM***

---

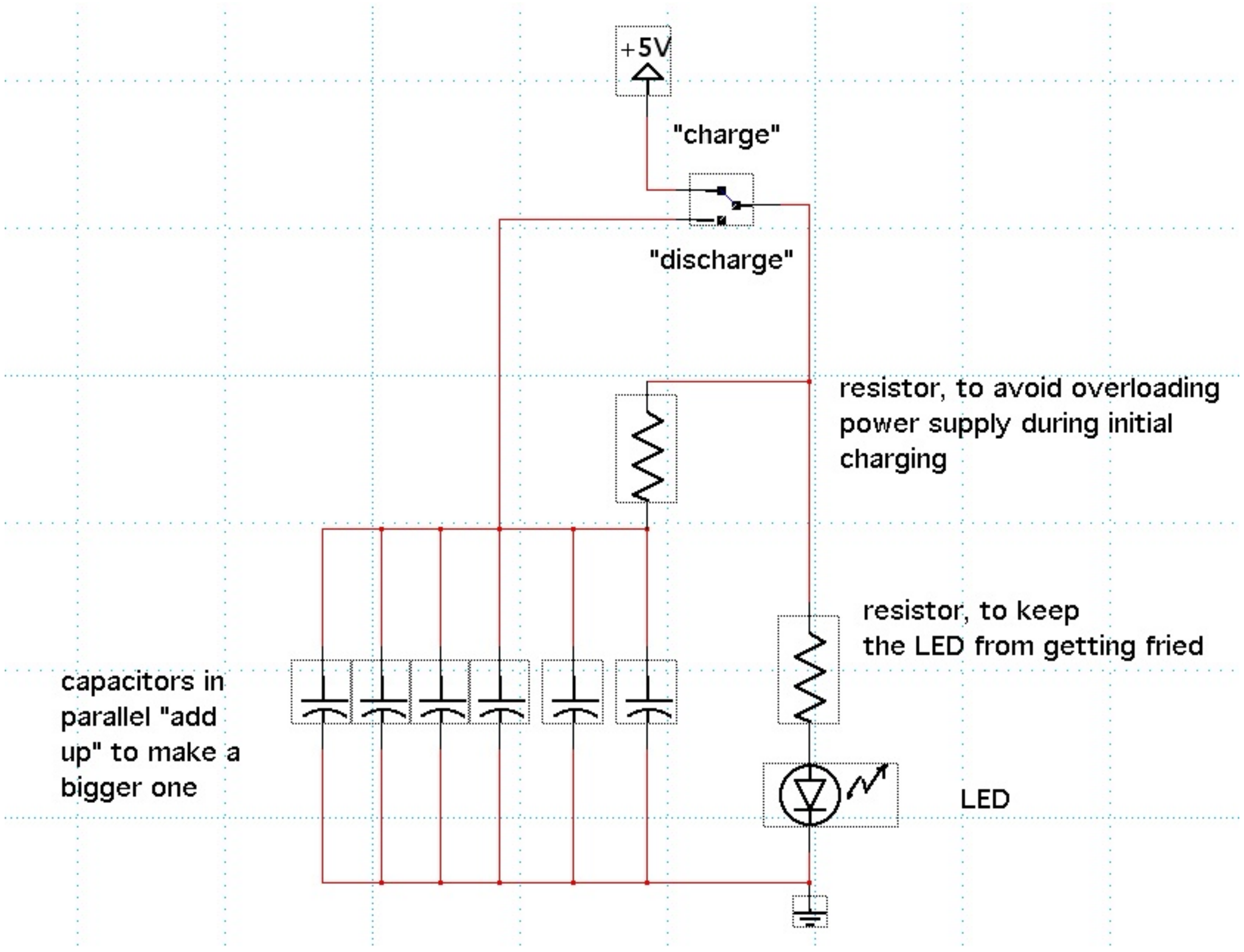

---

# 4. Dynamic RAM

---

Reality intervenes: using capacitors to build trickier but "sparser" DYNAMIC RAM (DRAM)... enabling much greater densities [http://micro.magnet.fsu.edu/electromag/java/capacitor/](http://micro.magnet.fsu.edu/electromag/java/capacitor/) (http://micro.magnet.fsu.edu/electromag/java/capacitor/)

(real demo)




Why don't we just use a capacitor to store 1 bit?

(There's a catch...and a workaround...)

---

What are the problems with building RAM out of capacitors?

---

DRAM: with capacitors (dense, higher power consumption)

SRAM: with flip-flops (less dense, but lower power)

(Yes, the textbook claims DRAM has lower power consumption, but I believe that's incorrect, particularly for long-term storage in battery-powered devices. The textbook probably does NOT take into account the cost of DRAM refresh)

---

See more discussion in section 6.1 of the textbook.

---