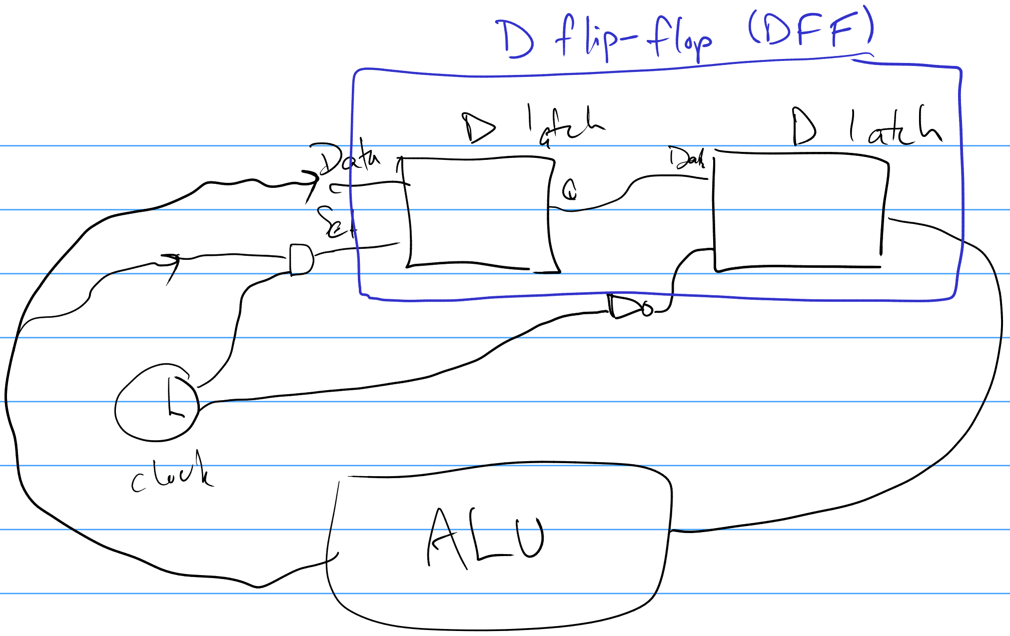


— stores 1 bit.

RAM = Random Access Memory

Memory address = integer (binary)

Our computer will have 15-bit addresses = 2^{15} memory locations.

Each memory location will store 16 bits.

- Amt of data a computer can process @ once is called <u>word size</u>.

How much memory is that?

$2^{15} \times 2^4 = 2^{19}$ bits. (1 byte = 8 bits)

$= 2^{16}$ bytes. (1 KB (properly 1 KiB = "kibibyte") $= 2^{10} = 1024$ bytes)

$= 2^6$ KB

$= 64$ KB

How much memory does 32-bit addresses give, if 1 byte per address?

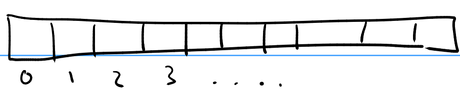
2^{32} B = 4 GB.

2^{10} Kilo
2^{20} Mega
2^{30} Giga
2^{40} Tera
2^{50} Peta
2^{60} Exa
2^{70} Zetta
2^{80} Yotta

<u>Random access</u> = we can access any location equally quickly.

Why? → Arrays!

Array of size n is n <u>consecutive</u> memory locations.



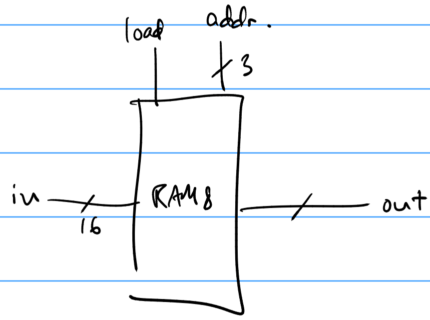
int[] arr = new int[10]   arr holds a memory address.

arr[3] → arr + 3*eltsize

Just math + a RAM lookup → O(1) time.

Project 3

- Bit = DFF + a 'load' bit
- Register = 16 bits. (still only one load bit input).
- RAM chips.
  - RAM8 – stores 8 registers.
    - input (16)
    - output (16)
    - load (1)
    - address (3)



  - RAM64

  etc.

- PC (program counter) – stores address of current instruction.