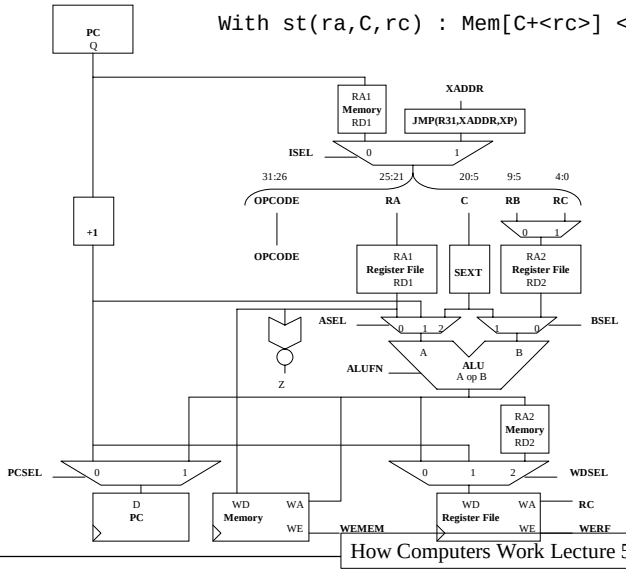
How Computers Work

Lecture 5

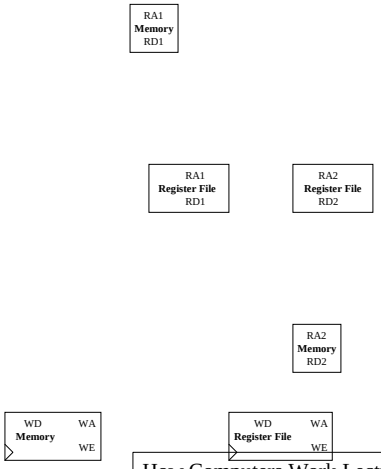
Memory Implementation

A Top-Down View of the Beta Architecture

With st(ra,C,rc) : Mem[C+<rc>] <- <ra>

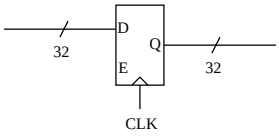


Today's Lecture: How do we build these?



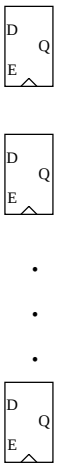
How Computers Work Lecture 5 Page 3

Recall the Enable-Controlled Register



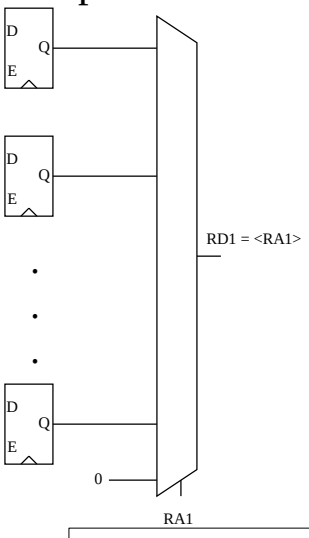
How Computers Work Lecture 5 Page 4

How do we select 1 of 31 registers to read?



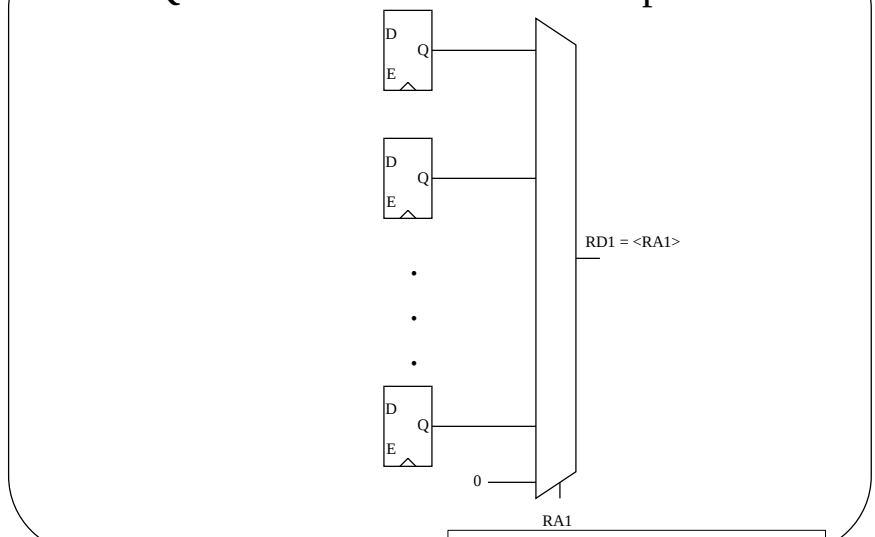
How Computers Work Lecture 5 Page 5

A: Add an output selector.



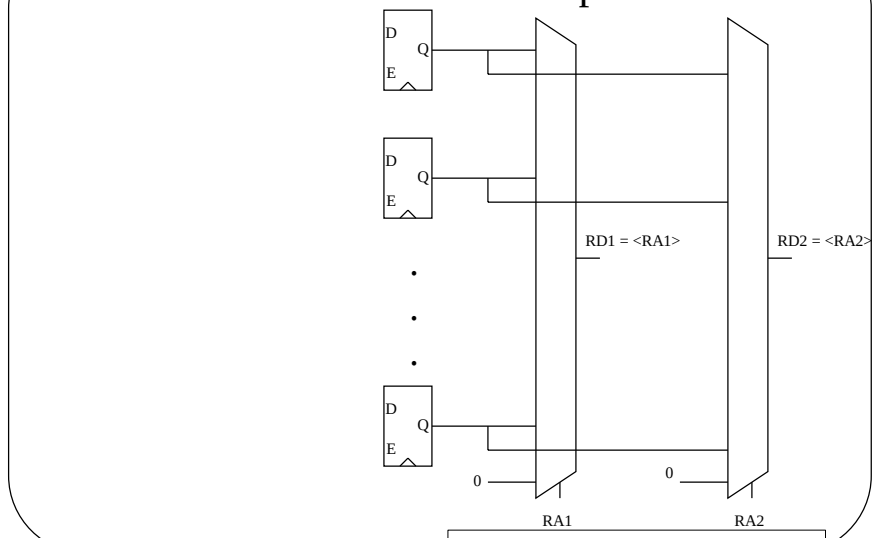
How Computers Work Lecture 5 Page 6

Q: How do we add a second port?



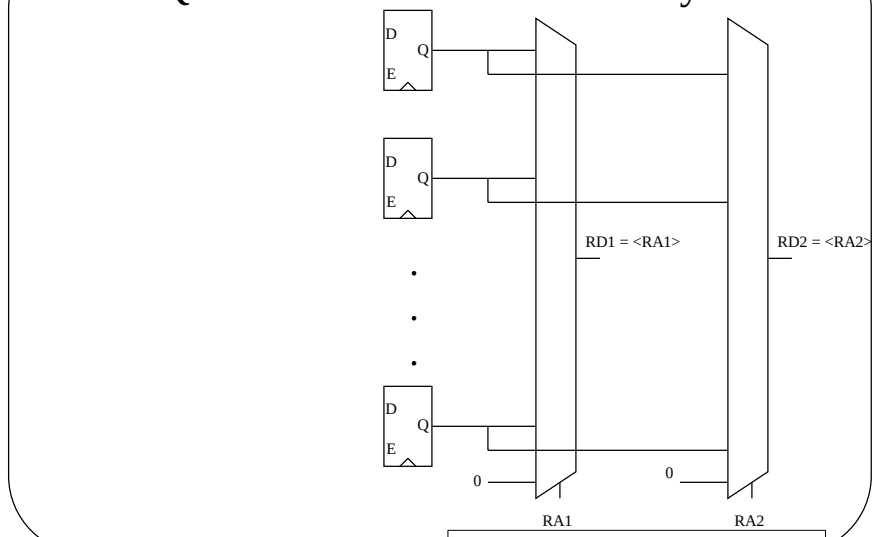
How Computers Work Lecture 5 Page 7

A: Add a second multiplexor



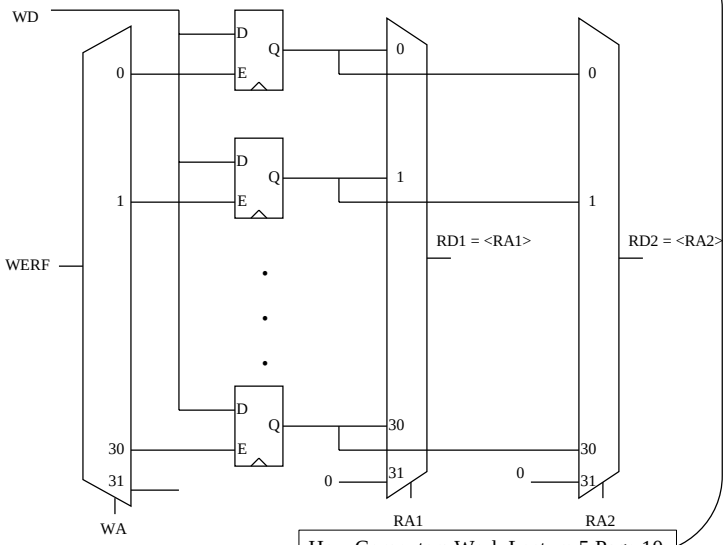
How Computers Work Lecture 5 Page 8

Q: How do we write selectively?



How Computers Work Lecture 5 Page 9

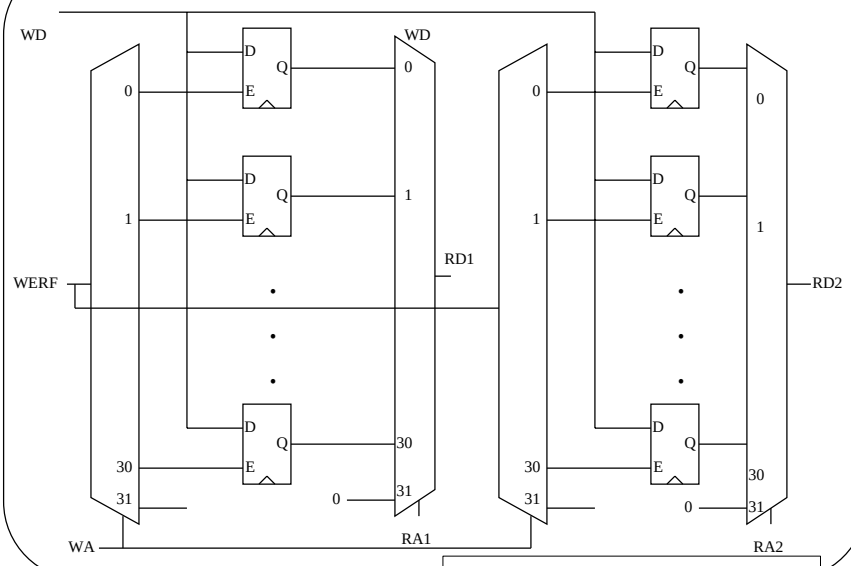
A: Use a decoder on the Enables



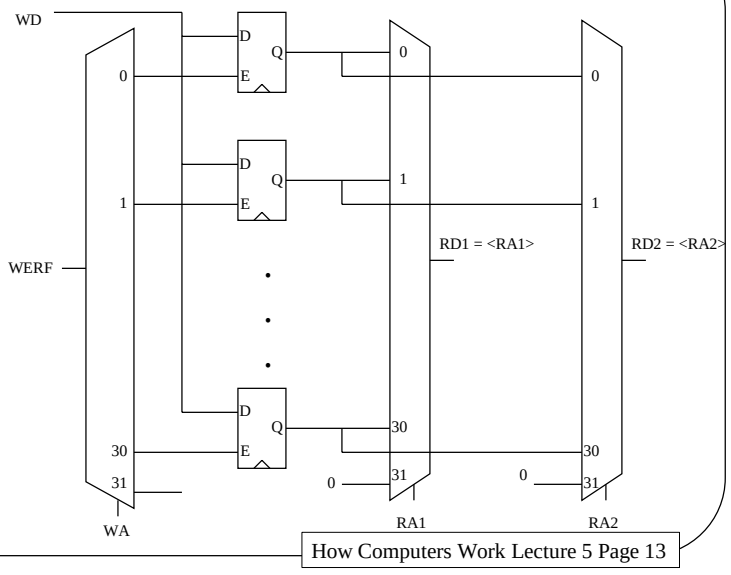
How Computers Work Lecture 5 Page 10

The Decoder / Demultiplexor

To minimize wires:

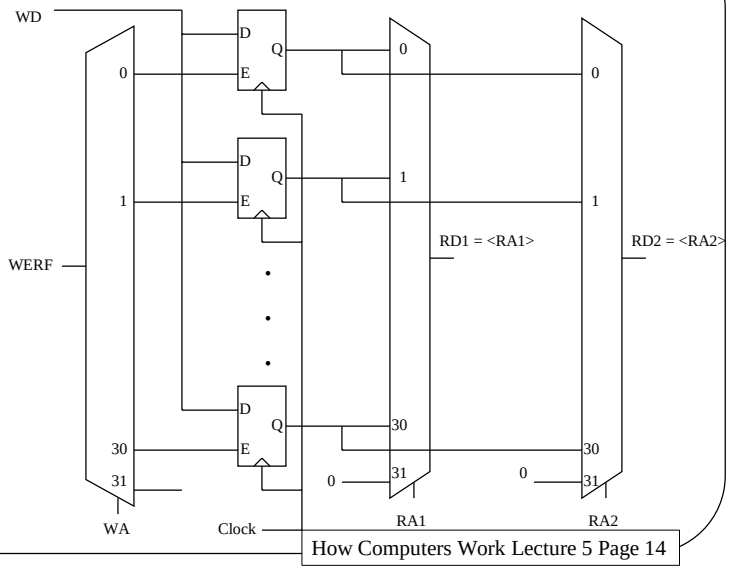


Q: What about the clocks?



How Computers Work Lecture 5 Page 13

A: Connect them all together.

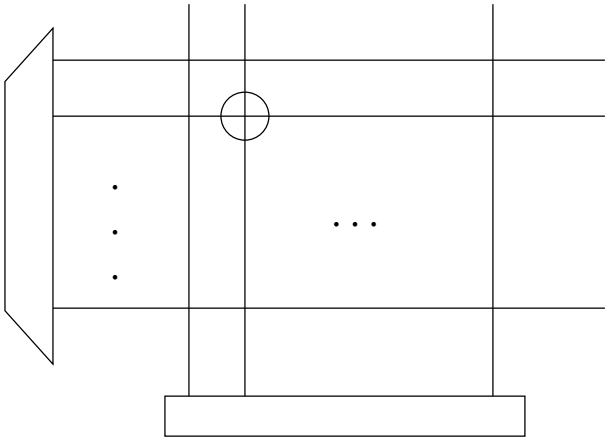


How Computers Work Lecture 5 Page 14

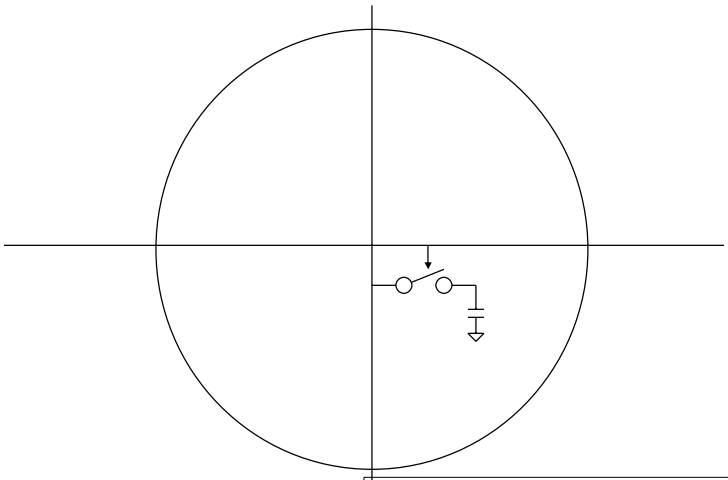
Q: Is it practical to do the big Memory this way?

A: NO

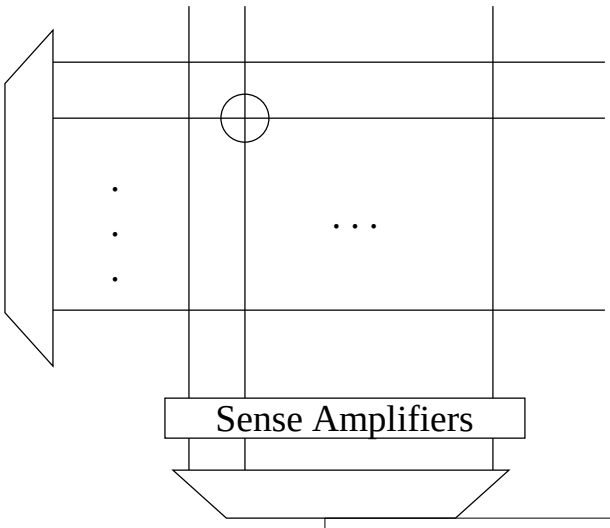
Minimize per-bit circuitry



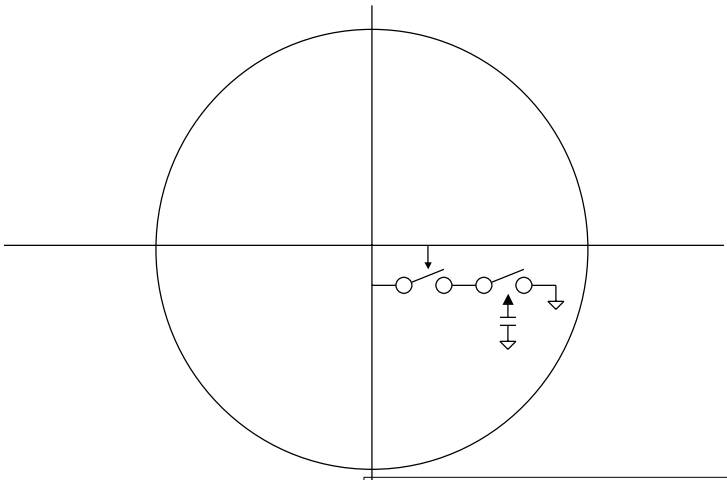
1 Bit Cell



Minimizing per-bit circuitry



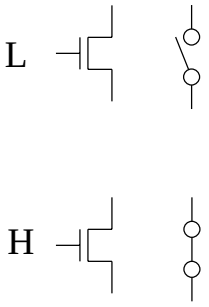
How about ROMs?



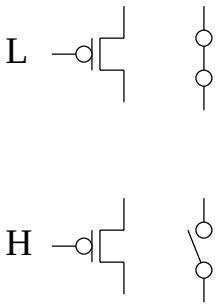
Q: What can we use for a switch?

A: The Field-Effect Transistor

The N-Channel FET (NFET)



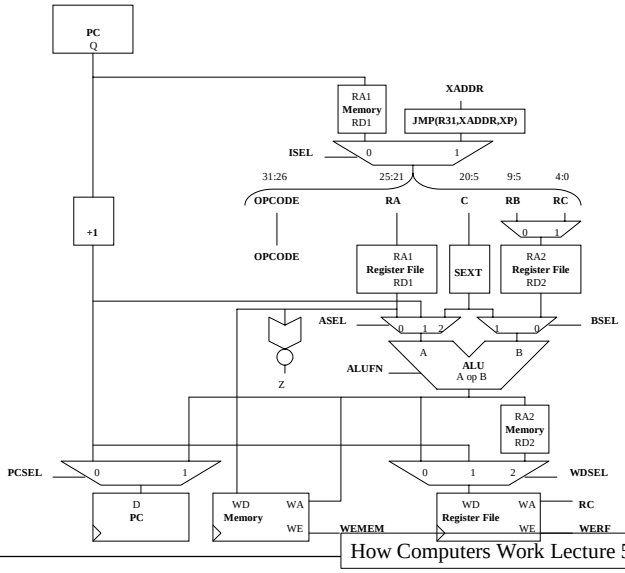
The P-Channel FET (PFET)



How do we implement multiple ports?

- 2 Read and 1 Write Ports
  - For now, LD and ST instructions are mutually exclusive.
    - 1 RD + 1 RD/WR port needed
- LD and ST are don't happen that often
  - Most of the time only 1RD port necessary
- Easy answer : Do them sequentially
  - Need a way to "stall" machine waiting for Mem

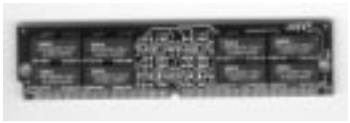
Q: How do we stall this machine?

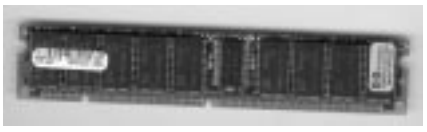


A: Stalls are done by:

- Disabling WERF
- Disabling Memory Write
- Disabling PC write

Another Approach - Increasing Memory *Bandwidth*





- Make memory twice as wide
  - 64 Bits Instead of 32
- Should work out in the long run, as 2 words are read per machine cycle, but
  - Words read are next to each other in address space
  - Need a place to stash the extra word
  - Sometimes, the stashed word isn't used.

Summary

- What Did we learn today?
  - How to Implement Registers + Big Memory
  - Multi-Port Big Memories aren't easy
    - Sequential Access (stalls + extra logic)
    - Wide Access + Some sort of *cache* + extra logic
- Recitation
  - Review of today's lecture