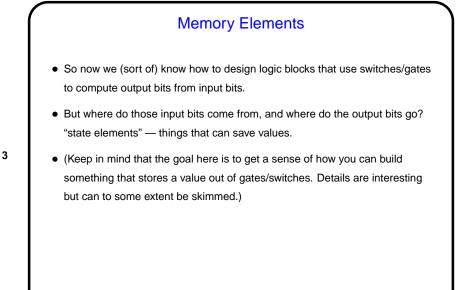
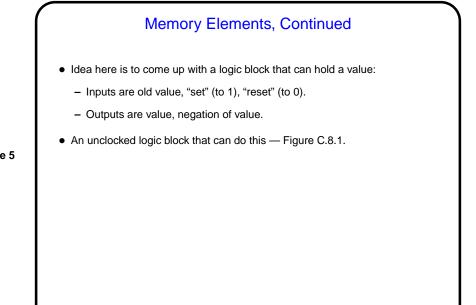


Minute Essay From Last Lecture

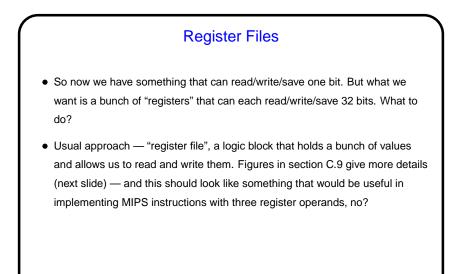
• Many people guessed right. Figures C.6.1 and C.6.2 provide another analogy that may be helpful in understanding the overall idea of data flowing through logic blocks.



A Very Little Bit About Clocking Many (most, currently?) hardware designs are based on the idea of a "clock" – something that generates regular signal changes and can be used to control when updates to state elements happen. As sketched in section C.7 — inputs/outputs to combinational logic block are connected to state elements. Input values are "sampled" at one point in the clock cycle and written out at a different point in the cycle — "synchronous" circuit. (So does that mean "asynchronous" circuits are also possible? yes, but well beyond the scope of this course.) Why do this? as a way to avoid race conditions. One implication, though, is that the clock cycle has to be long enough for the slowest combinational logic block!



Memory Elements, Continued Can then extend this to something that only samples (data) input when clock input is 1 ("D latch", Figure C.8.2) and further to something whose output only changes when clock input is 0 ("D flip-flop", Figure C.8.4). Notice how we're starting with simple things and using them to construct more complicated things — much as you do in writing software. "Hm!" ?



Register Files, Continued
Inputs:

Two (multi-bit) register numbers saying which registers we want to "read" (use as input to some operation).
One (multi-bit) register number saying which register we (might) want to "write" (change the value of).
One (32-bit) value to (maybe) save in a register.
A "yes do a write" bit.

Outputs:

Two (32-bit) values representing the contents of the two registers selected by the "read register" numbers used as input

