

CS 42—Sequential Logic

Thursday, September 20, 2018

Summary

Last time, we talked about combinational logic—how to turn truth tables into circuits using minterm expansion. We ended with a question: “As a computational model, how powerful are circuits?” For example, can they solve the same problems as DFAs? Can they solve the same problems as Turing Machines?

Today, we’ll build on combinational logic and talk about **sequential logic**. Sequential logic adds an element of time to circuits. We’ll ask: “How does adding time to combinational logic affect the computational model?”

We’ll also see how an important idea in computer science—the idea of self-reference—can help us use gates to build bits of memory.

The handout for today is unusual: there’s nothing on it other than this summary. That’s because we’ll be doing a lot of thought-experiments and exercises today where we draw circuits. The slides from today might be your best reference material.

Next time: Stored-program machines + assembly code