This discussion of design tradeoffs completes Part 1 of the course.

We've covered a lot of ground in the last eight lectures.

We started by looking at the mathematics underlying information theory and used it to help evaluate various alternative ways of effectively using sequences of bits to encode information content.

Then we turned our attention to adding carefully-chosen redundancies to our encoding to ensure that we could detect and even correct errors that corrupted our bit-level encodings.

Next we learned how analog signaling accumulates errors as we added processing elements to our system.

We solved the problem by using voltages "digitally" choosing two ranges of voltages to encode the bit values 0 and 1.

We had different signaling specifications for our outputs and inputs, adding noise margins to make our signaling more robust.

Then we developed the static discipline for combinational devices and were led to the conclusion that our devices had to be non-linear and exhibit gains > 1.

In our study of combinational logic, we fist learned about the MOSFET, a voltage-controlled switch.

We developed a technique for using MOSFETS to build CMOS combinational logic gates, which met all the criteria of the static discipline.

Then we discussed systematic ways of synthesizing larger combinational circuits that could implement any functionality we could express in the form a truth table.

To be able to perform sequences of operations, we first developed a reliable bistable storage element based on a positive feedback loop.

To ensure the storage elements worked correctly we imposed the dynamic discipline which required inputs to the storage elements to be stable just before and after the time the storage element was transitioned to "memory mode".

We introduced finite-state machines as a useful abstraction for designing sequential logic.

And then we figured out how to deal with asynchronous inputs in way that minimized the chance of incorrect operation due to metastability.

In the last two lectures we developed latency and throughput as performance measures for digital systems and discussed ways of achieving maximum throughput under various constraints.

We discussed how it's possible to make tradeoffs to achieve goals of minimizing power dissipation and increasing performance through decreased latency or increased throughput.

Whew!

That's a lot of information in a short amount of time.

Mr. Blue and the rest of the 6.004x staff hope you've found the course useful in increasing your skills in designing digital systems and analyzing their operation.

You've completed several actual designs and you're well on your way to designing a complete computer using our standard cell library.

That's quite an accomplishment.

If you'd like to continue the journey, please join us for Part 2 of the course where we'll discuss programmable architectures and work out the design of a modern 32-bit processor.

See you then!