LECTURE 28

Today…

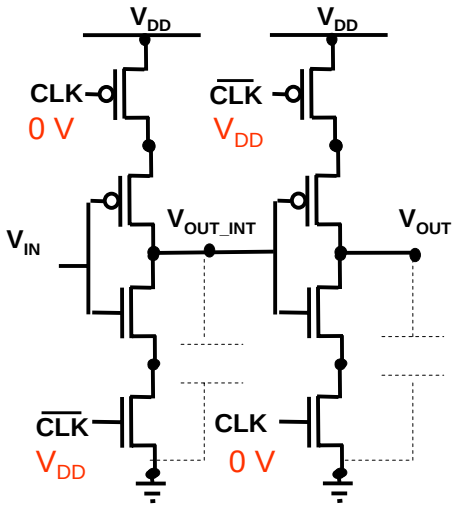
Analyzing digital computation at a very low level!

- The Latch
- Pipelined Datapath
- Control Signals
- Concept of State

Time permitting, RC circuits (where we **intentionally** put in resistance and capacitance to serve a useful purpose!)

- Relay
- Analog Counter
- Improved Integrator

THE LATCH



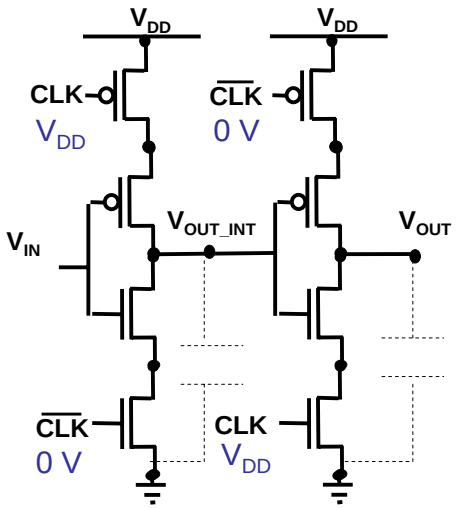
When CLK is low, the left-hand transistors conduct.

V_{OUT_INT} is charged to $\overline{V_{IN}}$.

When CLK is low, the right-hand transistors are open.

V_{OUT} remains the same; there is no charging path.

THE LATCH



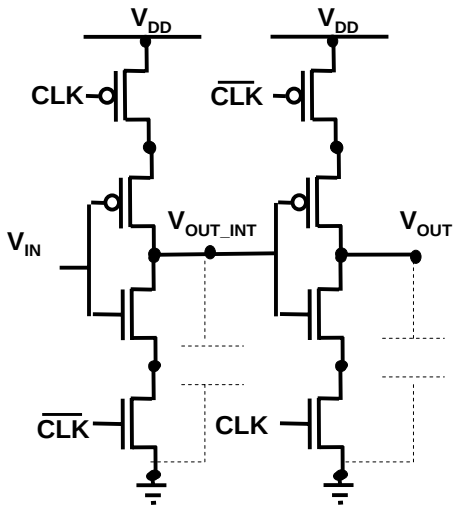
When CLK is high, the right-hand transistors conduct.

V_{OUT} is charged to $\overline{V_{OUT\_INT}}$.

When CLK is high, the left-hand transistors are open.

V_{OUT_INT} remains the same; there is no charging path.

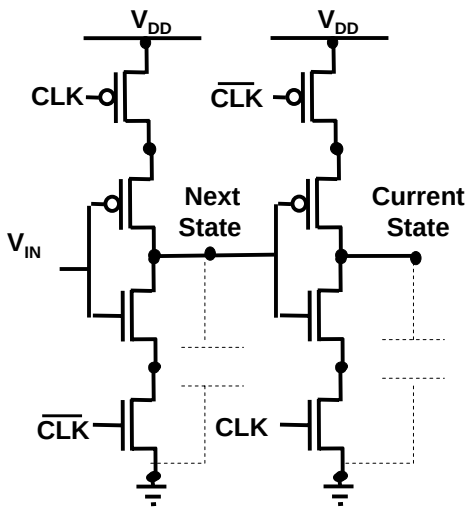
LATCH OPERATION



Data ($\overline{V_{IN}}$) is written to the internal node when the clock is low. The output remains frozen.

The internal node remains frozen when the clock is high. The (inverted) internal node voltage is written to the output.

CONCEPT OF STATE



A latch stores a “1” or “0”.

The stored value is known as the “state”.

This is one of the basic elements needed to make a “state machine” (covered in EE 20 and CS 61C).

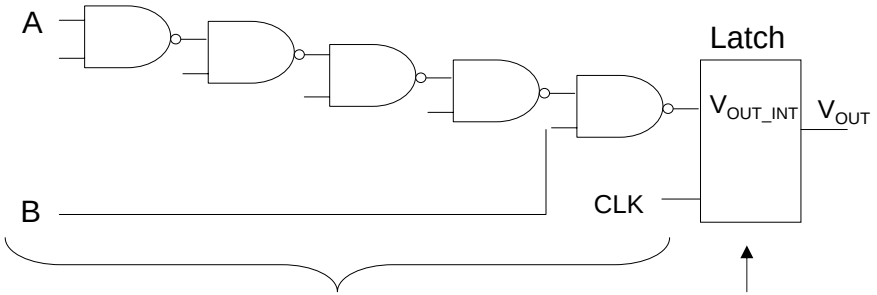
EECS 40 Spring 2003 Lecture 28 S. Ross

EECS 40 Spring 2003 Lecture 28 S. Ross

LATCH AS GATEKEEPER

A signal may have to go through a complex system of gates, with paths of different delays: possibility of false output!

Latch can prevent changing output until max delay has passed.



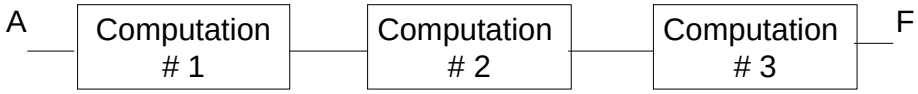
Combinatorial Logic
Signal propagates all the way through
Includes our logic gates: NAND, NOT, etc.

Sequential Element
Prevents changes in output until signaled

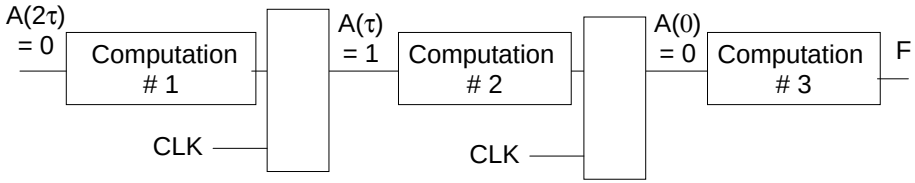
EECS 40 Spring 2003 Lecture 28 S. Ross

LATCH ENABLING PIPELINING

Why use latches? They add delay! Why not just wait for whole computation to complete? Eventually the final output will be correct, right? **Whole computation could take a long time…**



The insertion of latches allows **pipelining**. Parts of a long computation path can be used for the next computation before the first computation has finished. Increases computations per s!



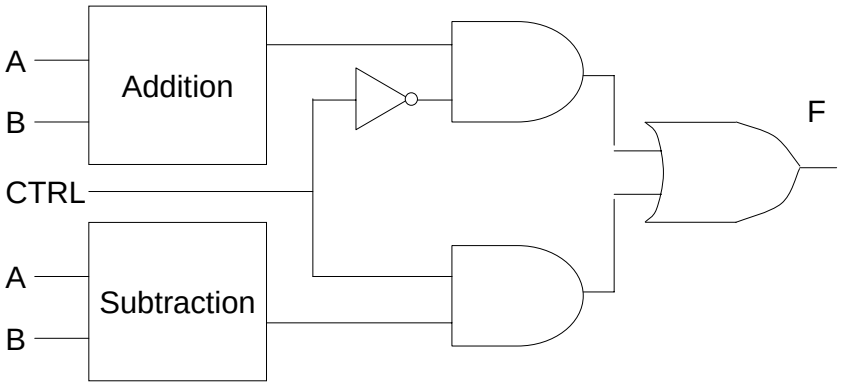
EECS 40 Spring 2003 Lecture 28 S. Ross

EECS 40 Spring 2003 Lecture 28 S. Ross

CONTROL SIGNALS

It would be pretty boring if a processor performed the same set of operations on each input.

With a little extra combinatorial logic, we can choose what operations to perform:

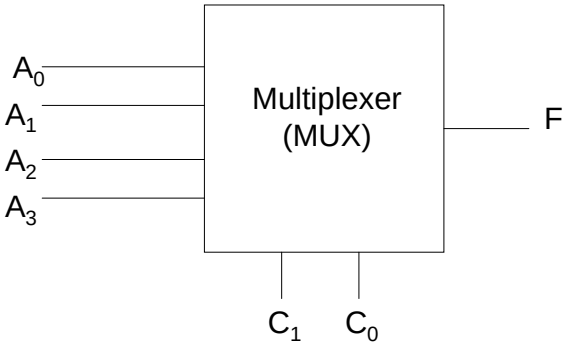


When will this circuit add? When will it subtract?

EECS 40 Spring 2003 Lecture 28 S. Ross

MULTIPLEXER

To choose between many operations, use a **multiplexer** instead of adding in lots of individual gates.



Choose which input goes through by setting C_1 and C_0.
Example: C_1 C_0 = 1 0 would let A_2 through.

EECS 40 Spring 2003 Lecture 28 S. Ross

EECS 40 Spring 2003 Lecture 28 S. Ross

YOUR COMPUTER AT THE TRANSISTOR LEVEL

Now we know how to represent and analyze logic states, computation, and control signals down to transistor level detail.

This is the electronic basis of many other EE and CS courses.

Designing processors and state machines, whether at the transistor-by-transistor level or at a much higher level, is explored in further EECS courses.

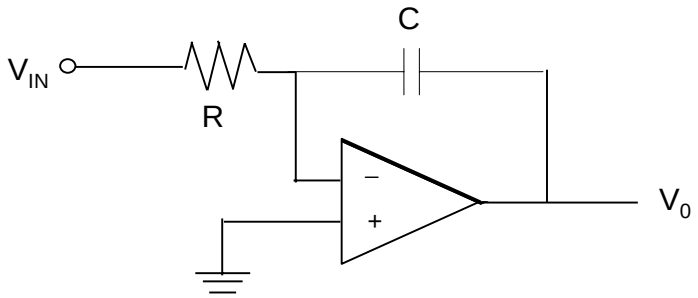
What are some digital design issues that already appear to be important through our analysis?

EECS 40 Spring 2003 Lecture 28 S. Ross

RC ON PURPOSE

RC analysis isn't just for computing gate delay.

The integrator uses R and C to accomplish its task:

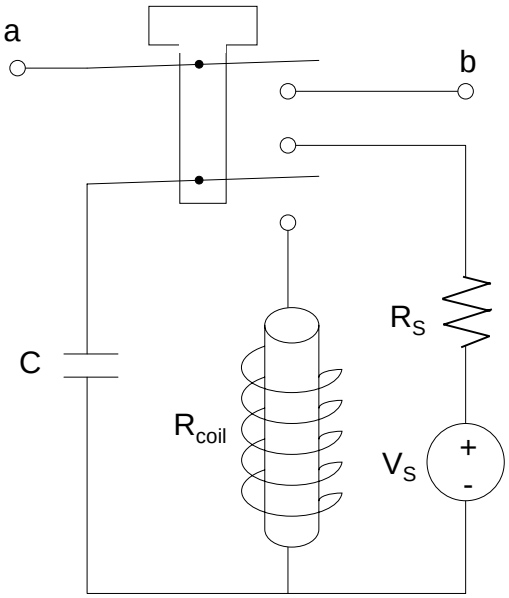


$$V_0(t) = -\frac{1}{RC}\int_0^t V_{IN}(T)dT + V_0(0)$$

EECS 40 Spring 2003 Lecture 28 S. Ross

EECS 40 Spring 2003 Lecture 28 S. Ross

THE RELAY



Start with button fully "up", capacitor fully charged.

The instant the button is pressed down, capacitor still has voltage—turns on coil magnet. Path from a to b stays connected until capacitor voltage decays enough to weaken magnet.

Switch springs back up.

EECS 40 Spring 2003 Lecture 28 S. Ross

THE RELAY

The relay holds the button down for a fixed amount of time.

How long? Say R_{coil} = 25 kΩ, R = 4 kΩ, V_S = 10 V, C = 2 μF. Magnet will release when coil voltage drops to 5 V.

The question: how long does it take fully charged capacitor (10 V) in parallel with coil to discharge to 5 V?

Answer:

EECS 40 Spring 2003 Lecture 28 S. Ross

EECS 40 Spring 2003 Lecture 28 S. Ross

ANALOG COUNTER

Let's design a circuit that increases V_{out} by 1 V every time a button is pressed.

Storing voltage and adding to it over time -> integrator

With the relay from the example, I can create a voltage pulse of duration 34.5 ms when a button is pressed.

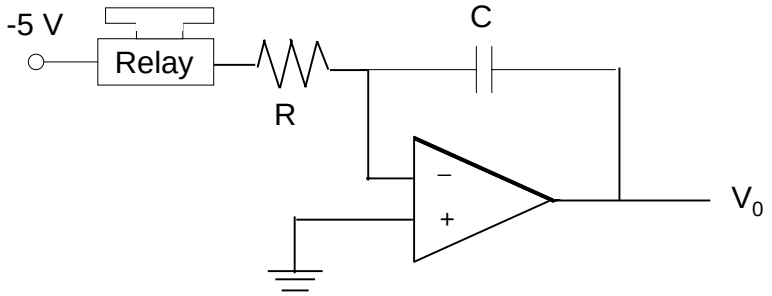
Each pulse can be fed to an integrator, to increment the output voltage by 1 V, by adding in the area under the pulse (scaled to make the increment 1 V).

Say that the voltage input is 5 V. What is the area under each pulse?

Answer:

EECS 40 Spring 2003 Lecture 28 S. Ross

ANALOG COUNTER



The area under each pulse is 0.1725 V s.

The scaling factor, 1/RC, should therefore be $(0.1725\ \text{V s})^{-1}$ to make the V_{out} increments equal to 1 V.

EECS 40 Spring 2003 Lecture 28 S. Ross

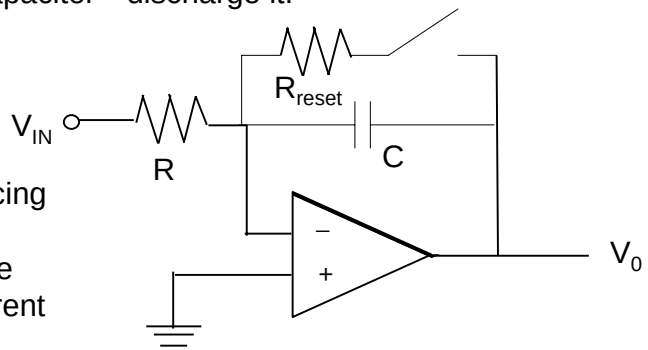
EECS 40 Spring 2003 Lecture 28 S. Ross

INTEGRATOR RESET

What if we want to restart the counting? What if we want to reset an integrator?

What part of the integrator "stores" information, and how do we "erase" it?

Answer: The capacitor—discharge it!



Factors influencing choice of R_{reset}:
- Discharge time
- Maximum current

EECS 40 Spring 2003 Lecture 28 S. Ross