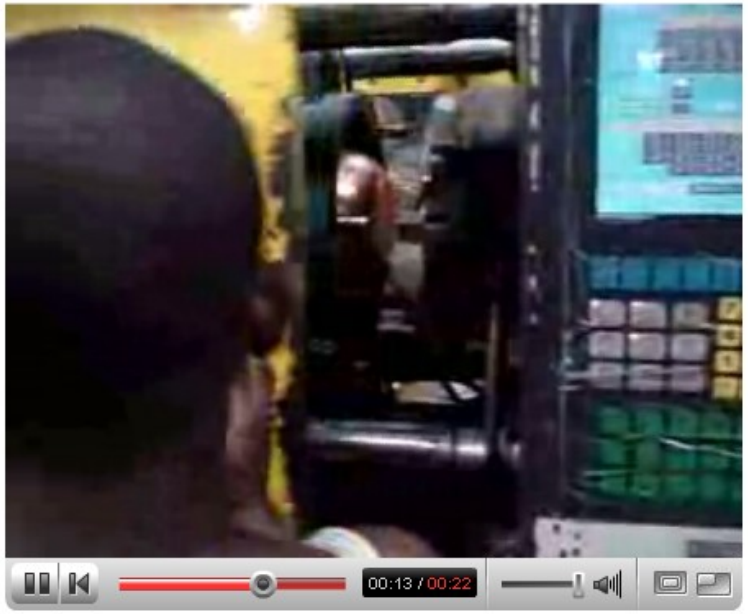
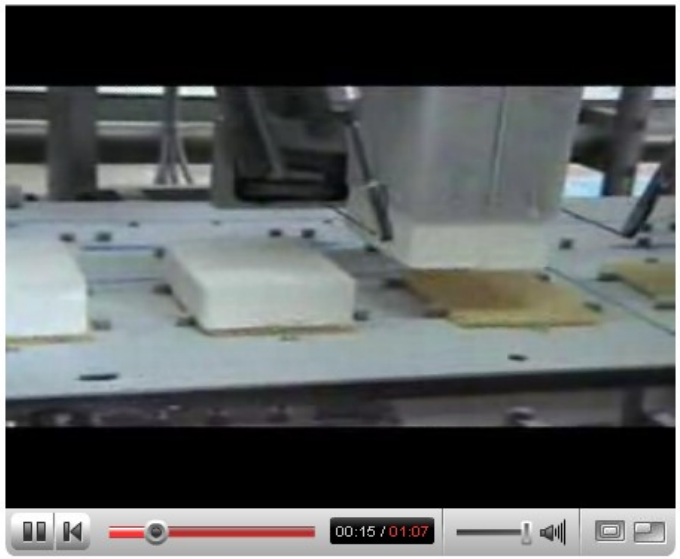
Industrial Automation

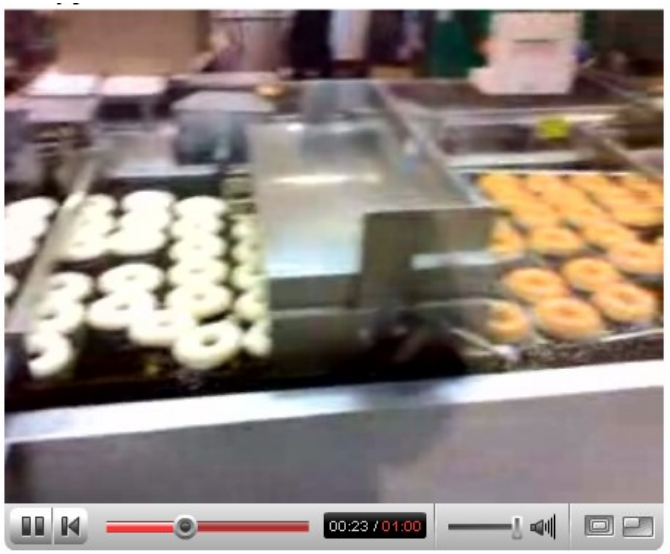
- Common knowledge: computers are used in factories...
- Robotic arm, CNC, injection molding



Industrial Automation

- Donut machine, Ice cream sandwich machines





Industrial Automation

- Not common knowledge: today this is usually accomplished with Programmable Logic Controllers (PLCs)

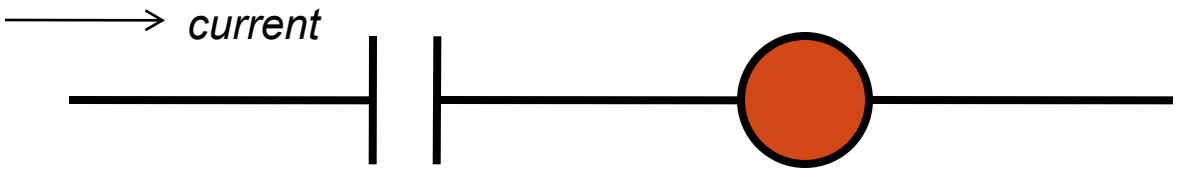
- PLCs are the answer to a variety of needs: durability, reliability, flexibility, scalability, reprogrammability, etc...

Industrial Automation

- Why should you care? Because you will run into PLCs...
- Median starting salary for entry-level “Electrical Controls Engineer” is $57,452. (EE is $55K, HW Eng is $48K, SW Eng is $53K) [monster.com]
- As long as there is industry, it will be computer controlled and engineers will earn paychecks.

Relay Logic

- Conditional logic can be represented in terms of *contacts* and *coils*.



- Contact: A simple input switch.
- Coil: An output load, e.g., a relay or motor.
- Symbolic representation called *ladder logic*.

Relay Logic

- To clarify: “Ladder Logic” is a notation **originally** used to describe & document relay logic configurations.
- Later became the basis for PLC programming languages

Ladder Logic

- Power supply rails drawn as parallel vertical lines on left and right
- Connection of rails implies current will flow
- An output is “on” when a connection is completed and current flows through the load’s coil

Ladder Logic

- Simple “always on” load:

[Always_On = 1] note: sometimes illegal

- Boring... Load controlled by a single contact:

[Switch_Con = Switch]

Ladder Logic

- Boolean logic - C = A and B

- C = A or B

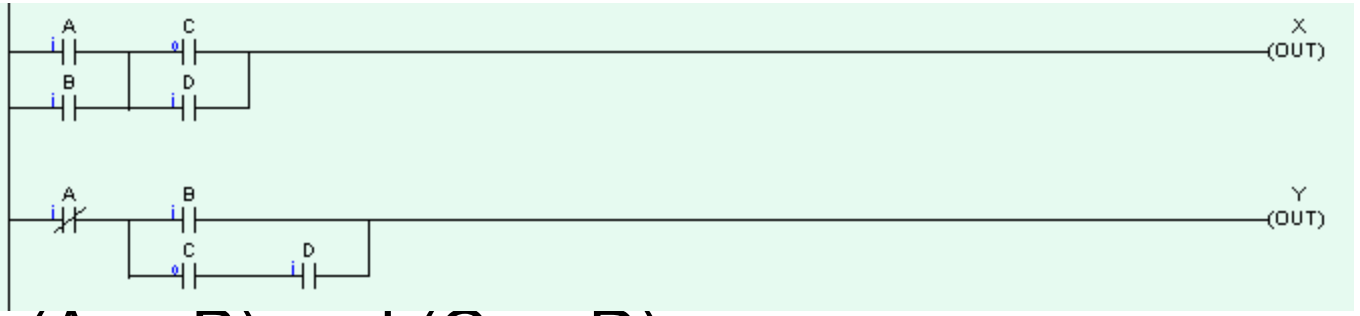
Ladder Logic

- C = not A

- A contact with a slash through it is “normally closed.” This indicates a connection when A is NOT triggered.
- So when sensor/input A is activated, there is an *open* circuit

Ladder Logic

- Each *rung* of the ladder is a statement that is **asynchronous** when implemented in relay logic, but evaluated **sequentially** by the PLC.



- X = (A or B) and (C or D),

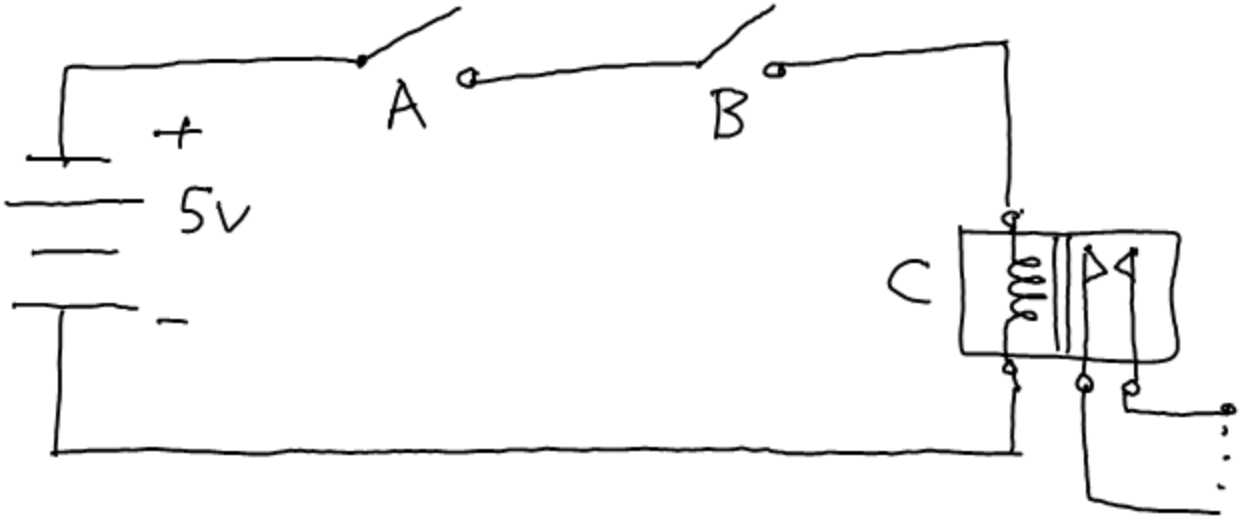
  Y = ~A and [B or (C and D)]

Ladder Logic

- Converting between ladder logic and physical electronics is straight forward.
- So this…

Ladder Logic

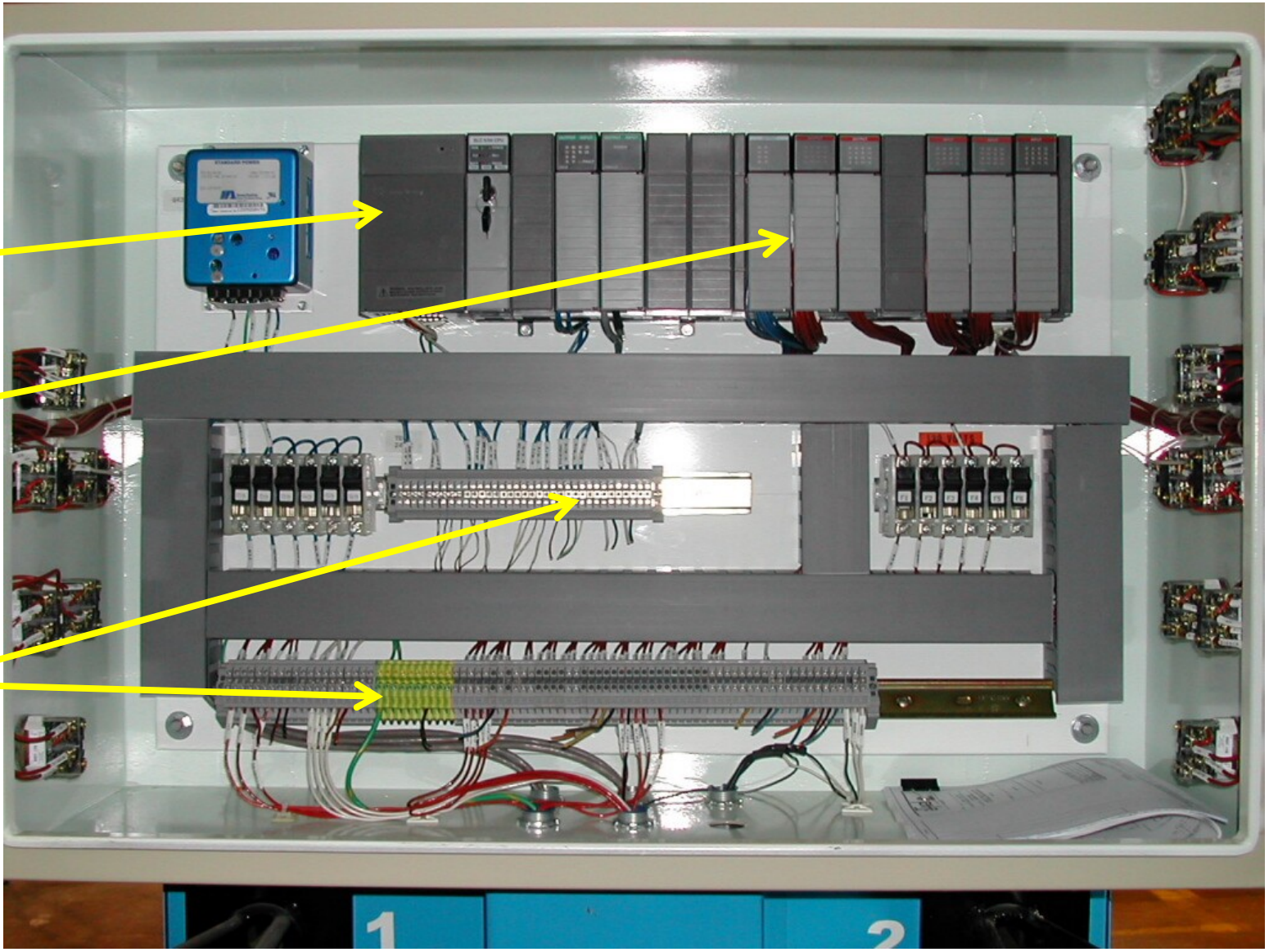
- …becomes this:



PLC

- The first PLC was invented by Dick Morely in 1978.
- Morely designed a computer with three components: a processor, memory, and a logic solver.
- “[The logic solver] allowed us to get the speed we needed in this application-specific computer to solve the perceptually simple problem of several cabinets full of relay wiring.” -Morely

Industrial Installation



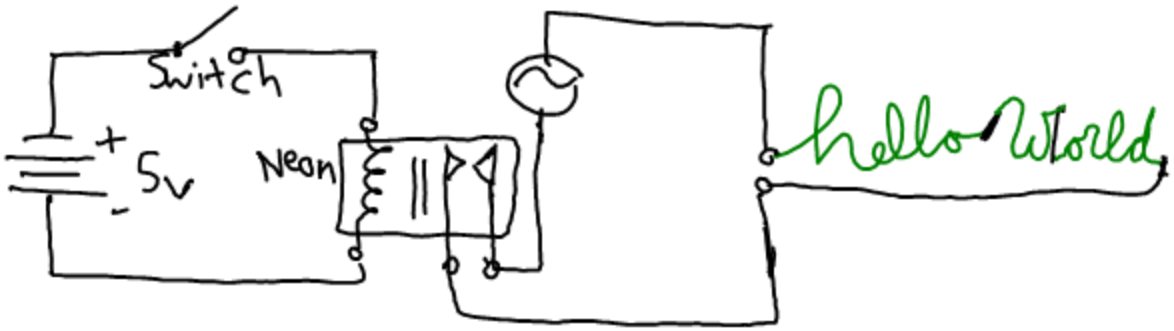
PLC

- The first PLC (the 084) was extremely durable and reliable...
- “We used to test the programmable controllers with a Tesla coil that struck a quarter inch to half-inch arch anywhere on the system, and the programmable controller still had to continue to run.” –Morely

- FYI, this is a Tesla coil: http://www.youtube.com/watch?v=FY-AS13fI30

PLC

- Hello World on the PLC. Real hardware:



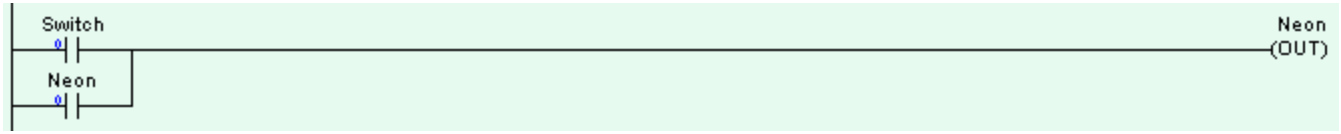
- In ladder logic:

PLC

- Not very interesting… how about a “stay-on” variation? (When the switch is released, the light stays on)
- PLC benefit: The state of an “output” in one rung may be used as a “contact” in another.
- In fact, there are “internal utility relays” – *virtual* outputs that act as intermediate steps toward real outputs.

PLC

- Latched (“stay-on”) Hello World:



- When the switch is pressed, “Neon” will be active in the first evaluation.
- In subsequent evaluations, “Neon” will force itself to stay on.
- NOTE: OUTPUTS CAN ALSO BE INPUTS