Control System Training

MODULE 5 – Sequential Boolean Logic

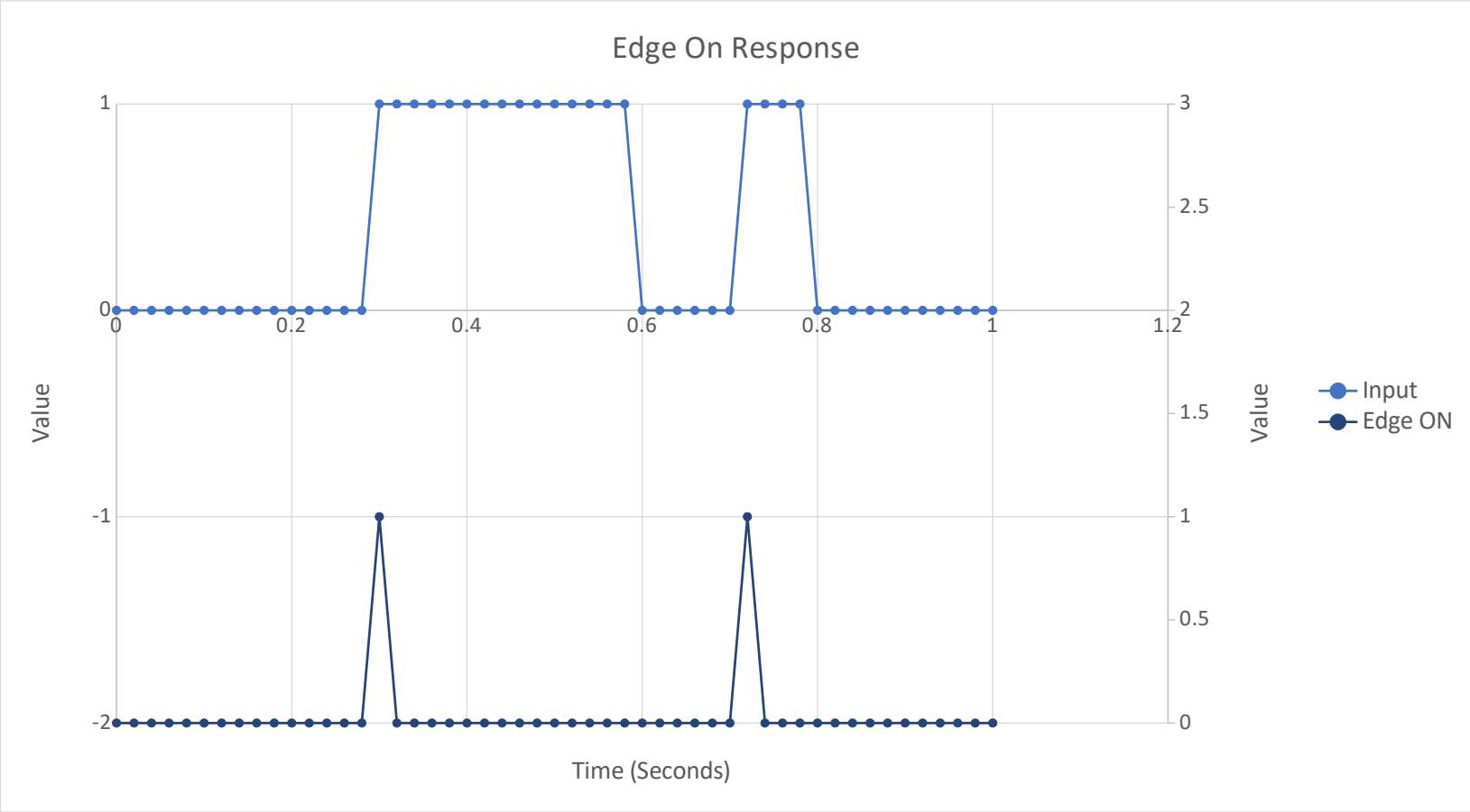
Copyright Notice

Sequential Boolean Logic

Definitions:

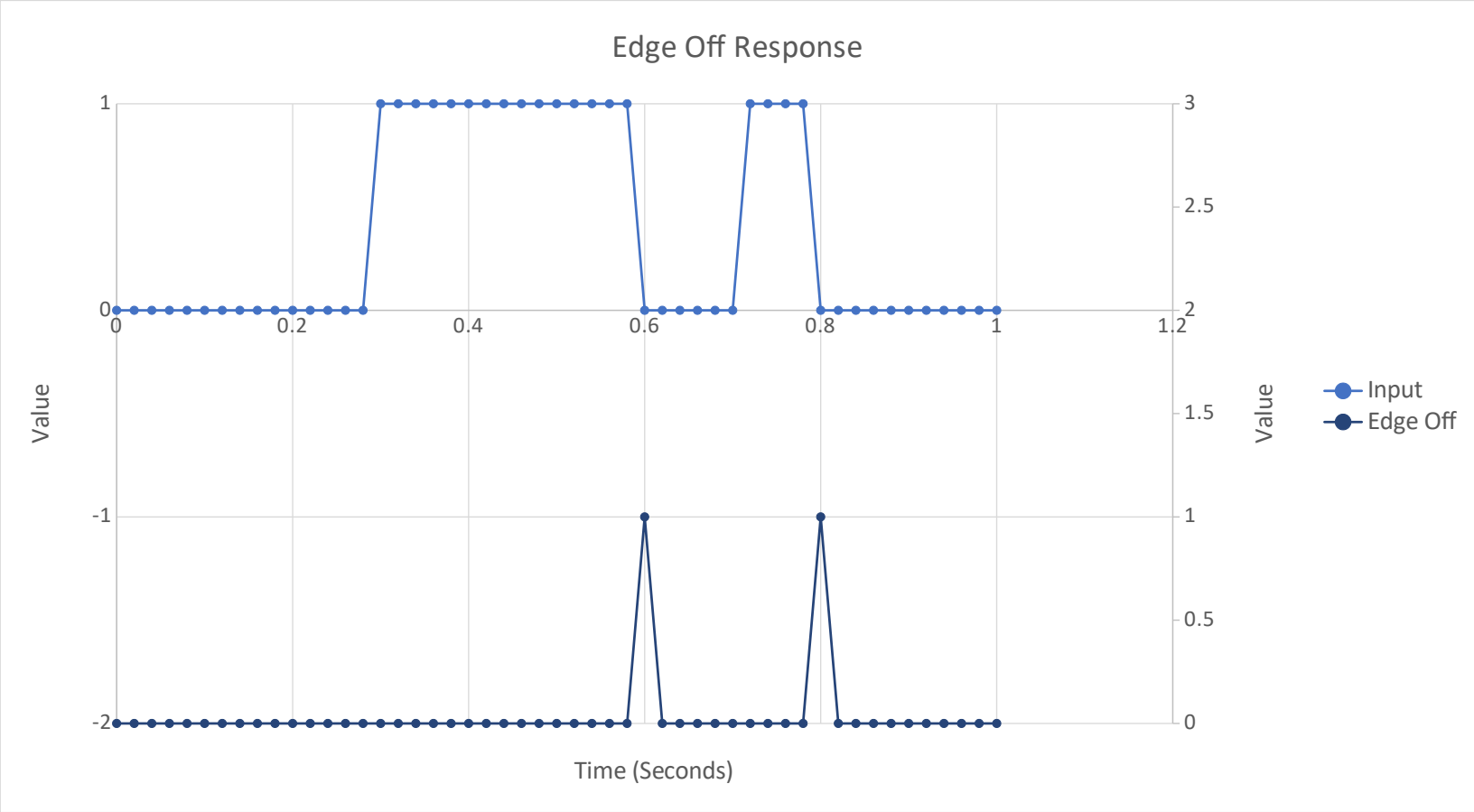
- **Boolean** – Only values are: ZERO / ONE or TRUE/FALSE
- **Combinatorial Logic** – Outcome depends only on the current value of the inputs. Nothing is depends on time (or previous values of the inputs or outputs).
- **Sequential Logic --** Outcome depends on both current values of the inputs and previous values of the inputs and output.

Edge Triggered – ON



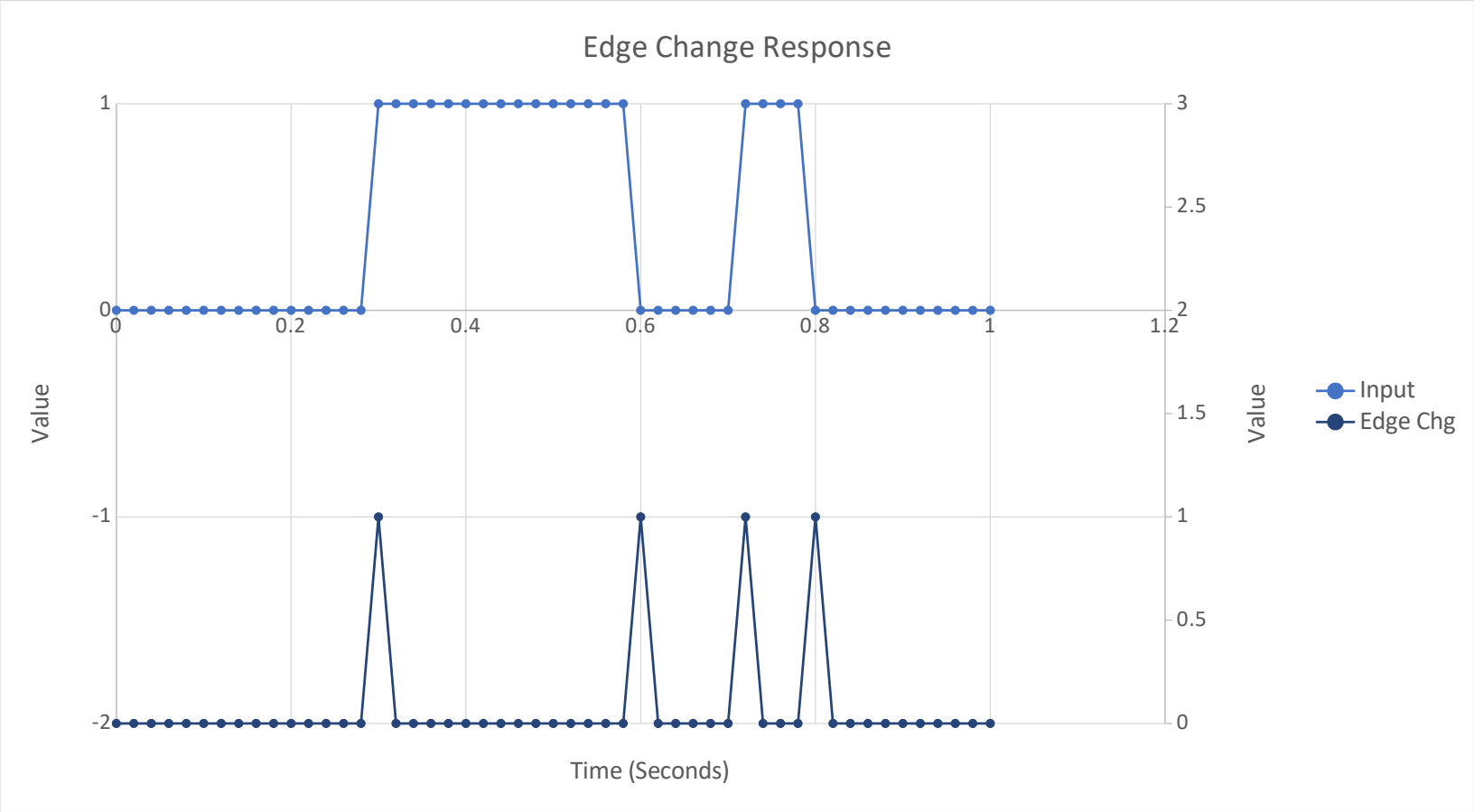
- **Only true for a single scan.**

Edge Triggered - OFF



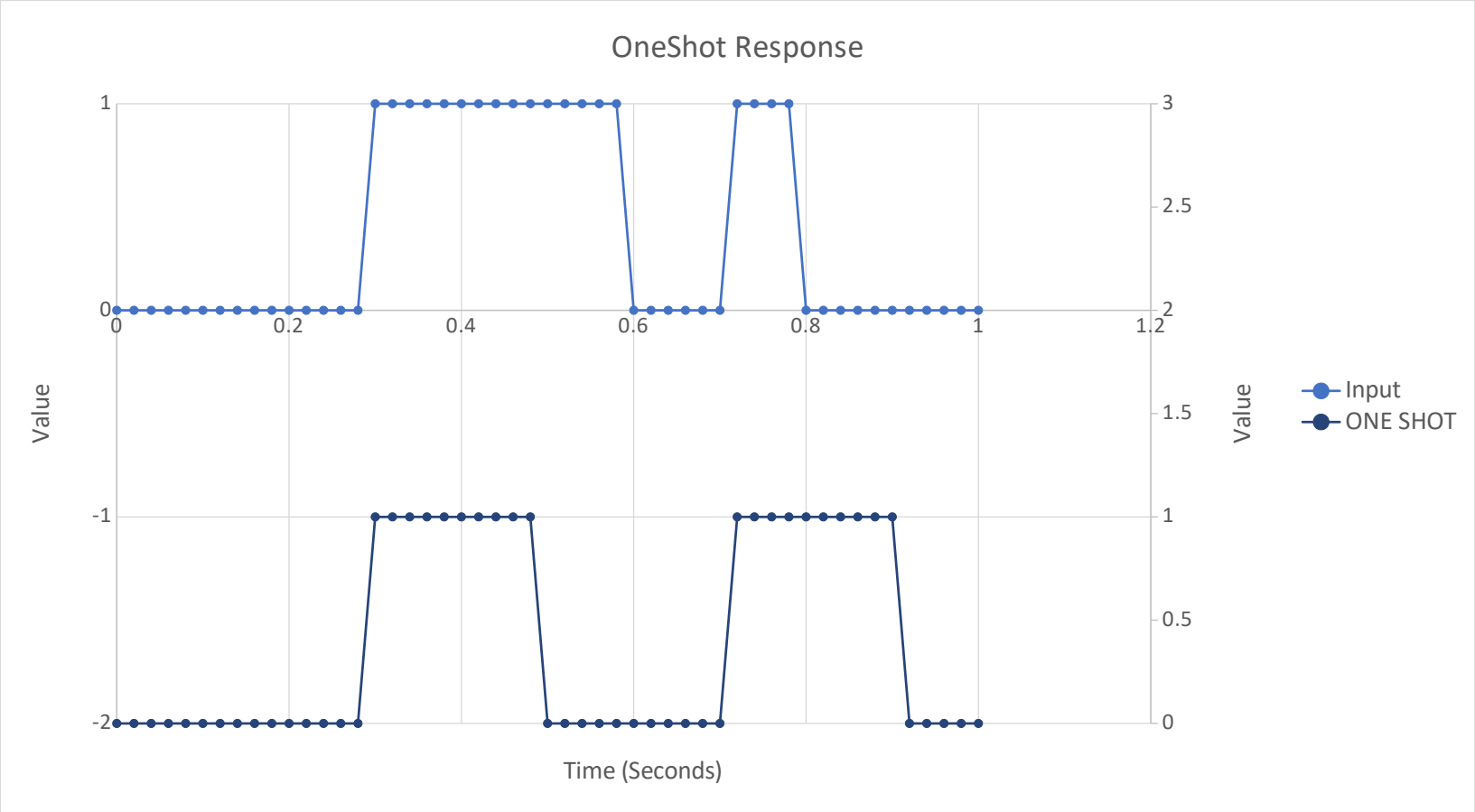
- **Only true for a single scan**

Edge Triggered - CHANGE



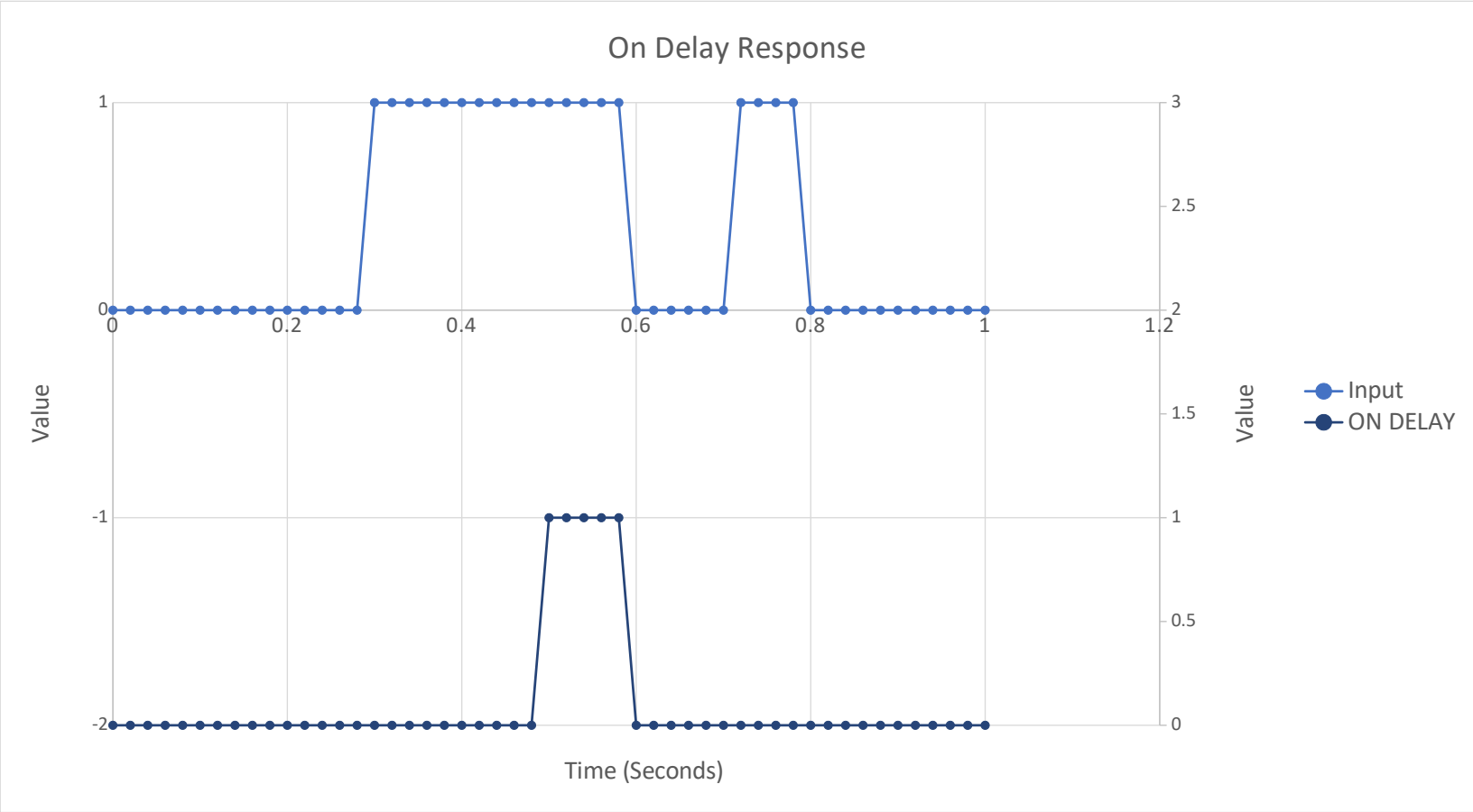
- **Only true for a single scan**

One Shot



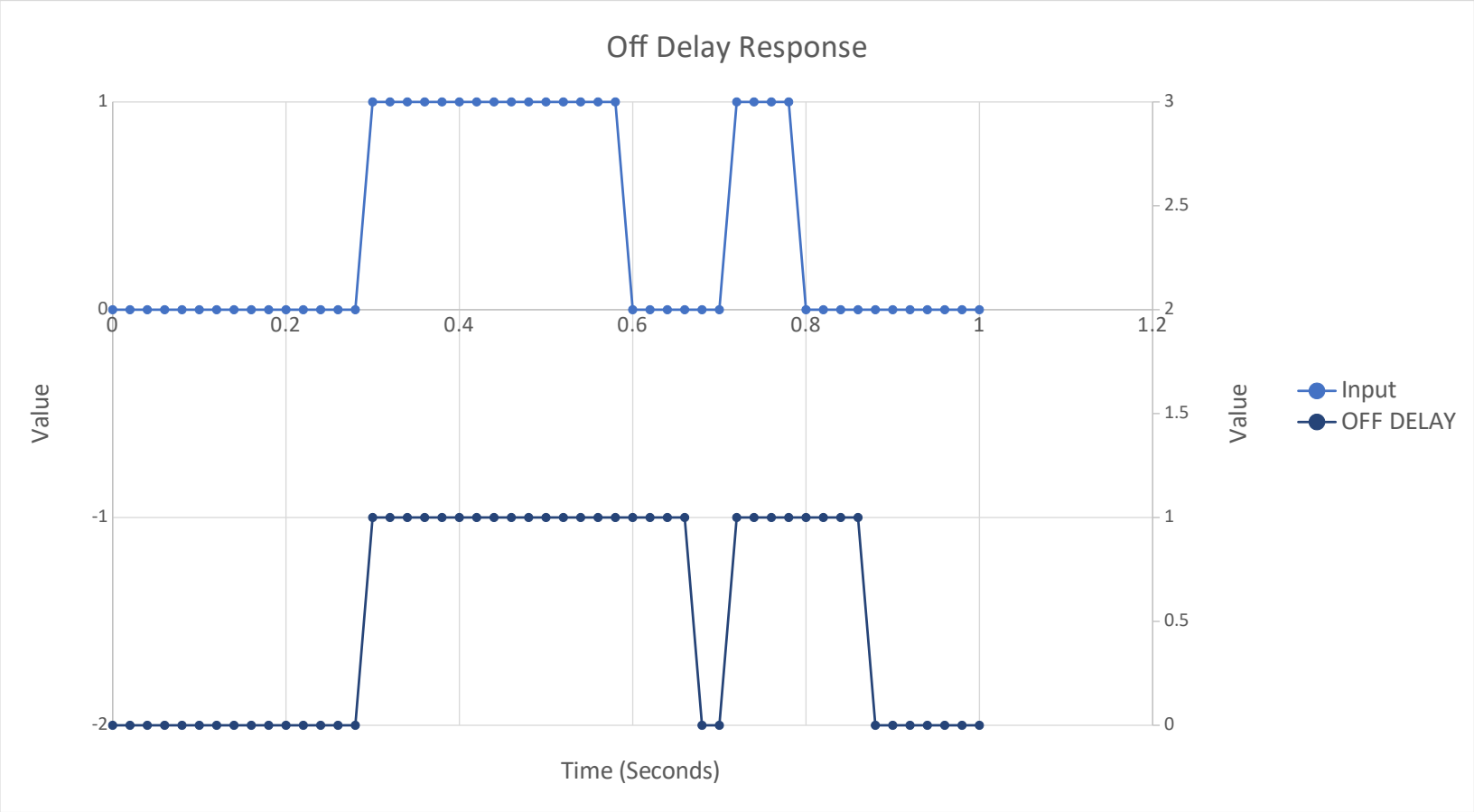
- **Oneshot time is 0.200 seconds**

On Delay



- **Delay is 0.200 Seconds**
- **If input goes false before delay expires, output is never true.**

Off Delay

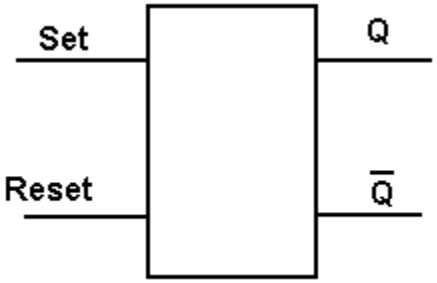


- **Off delay time is 0.080 seconds.**

Set / Reset Flip Flop

Set / Reset Flip Flop

Input			
Set	Reset	Override	Output
1	0	N/A	1
0	1	N/A	0
0	0	N/A	Prev Output
1	1	Set	1
1	1	Reset	0



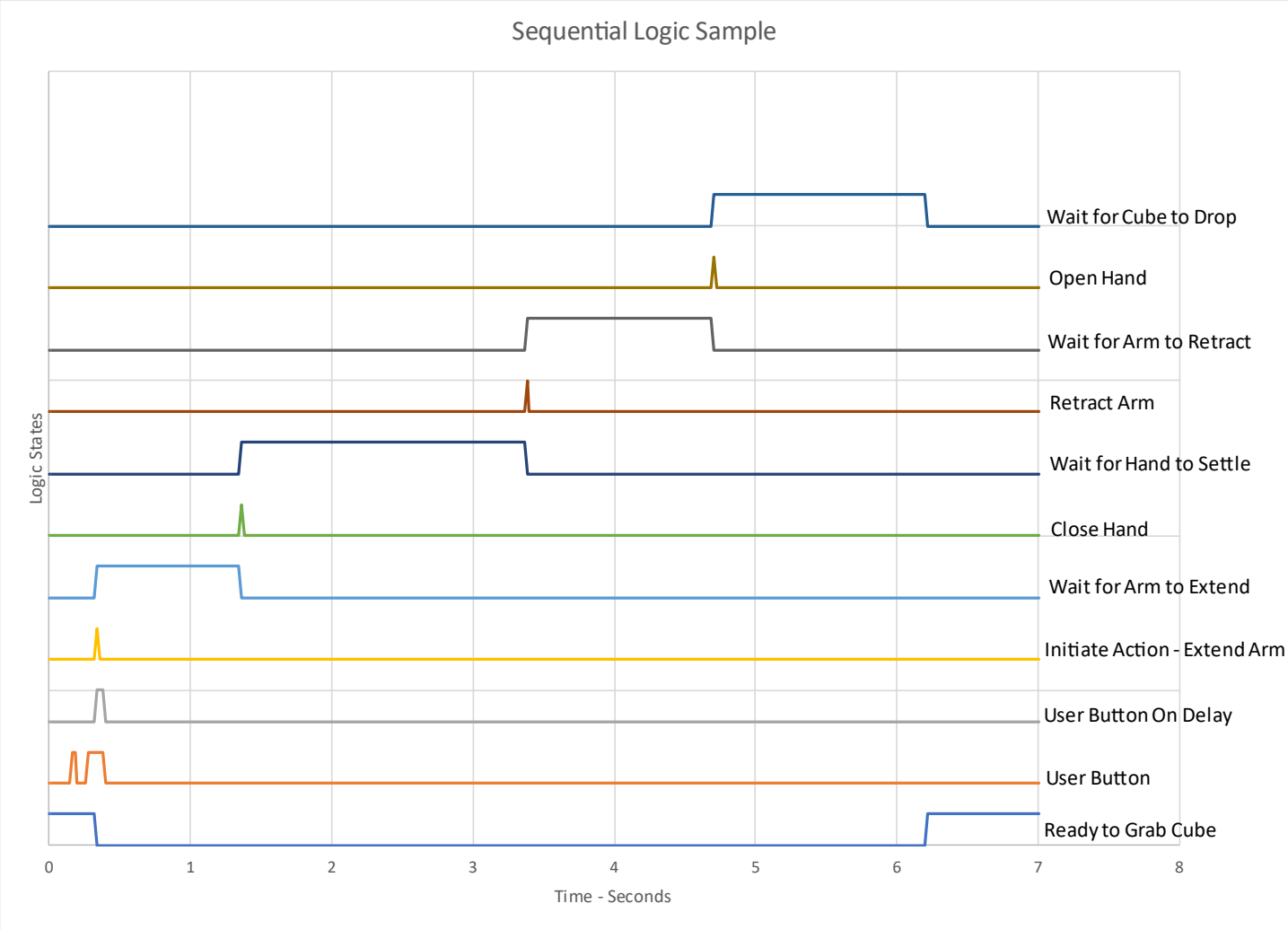
- **Can think of this as "Boolean memory". This is the building block of all computers.**
- **If both are true at the same time, one overrides the other. This is usually selectable.**

Designing Sequential Logic 1/4

- **Sample Problem – Cube Capture**
  - Control system uses a 20 msec loop time
  - System is ready when “hand” is opened and “arm” is retracted
  - Users pushes button to initiate “cube capture”. Auto repeat of cube capture is not allowed.
  - Ensure user pushed button for 60 msec
  - Close “arm” extension solenoid. Wait 1 second for arm to extend.
  - Close “hand” solenoid. Wait 2.0 seconds for “hand” to settle.
  - Open “arm” extension solenoid. Wait 1.3 seconds for arm to retract.
  - Open “hand” solenoid to release potential cube into bin. Wait 1.5 seconds for cube to drop before allowing next “capture” action.
- **Enhancement – Add a cancel button**

Designing Sequential Logic 2/4

- **Draw the Logic Diagram Graph**

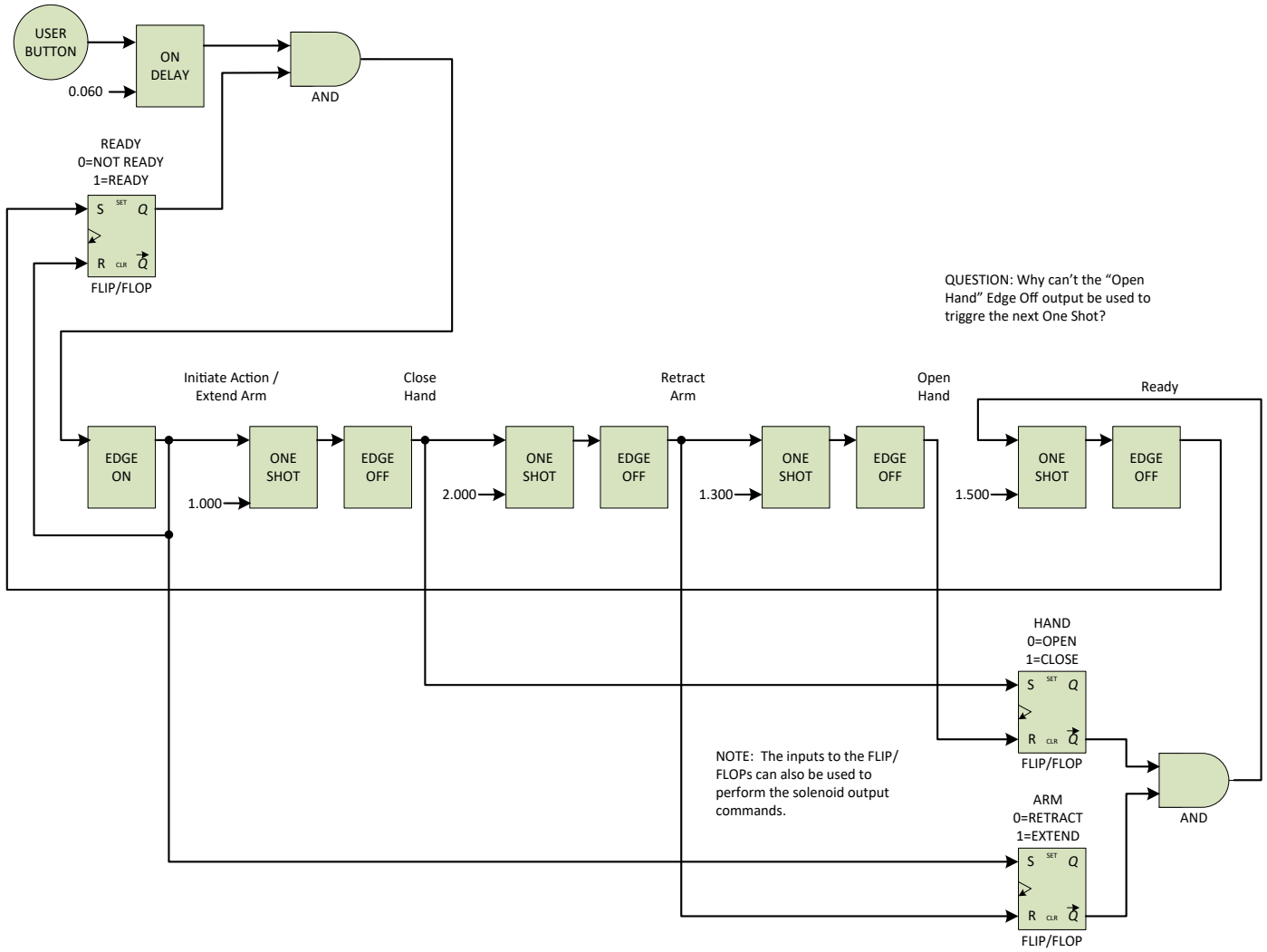


Designing Sequential Logic 3/4

- **Start with inputs**
- **Determine relationships**
  - What inputs does an output relate to
- **Draw Logic Diagram**
- **Add intermediate Logic to Graph**
- **Repeat last two steps if needed to refine logic**

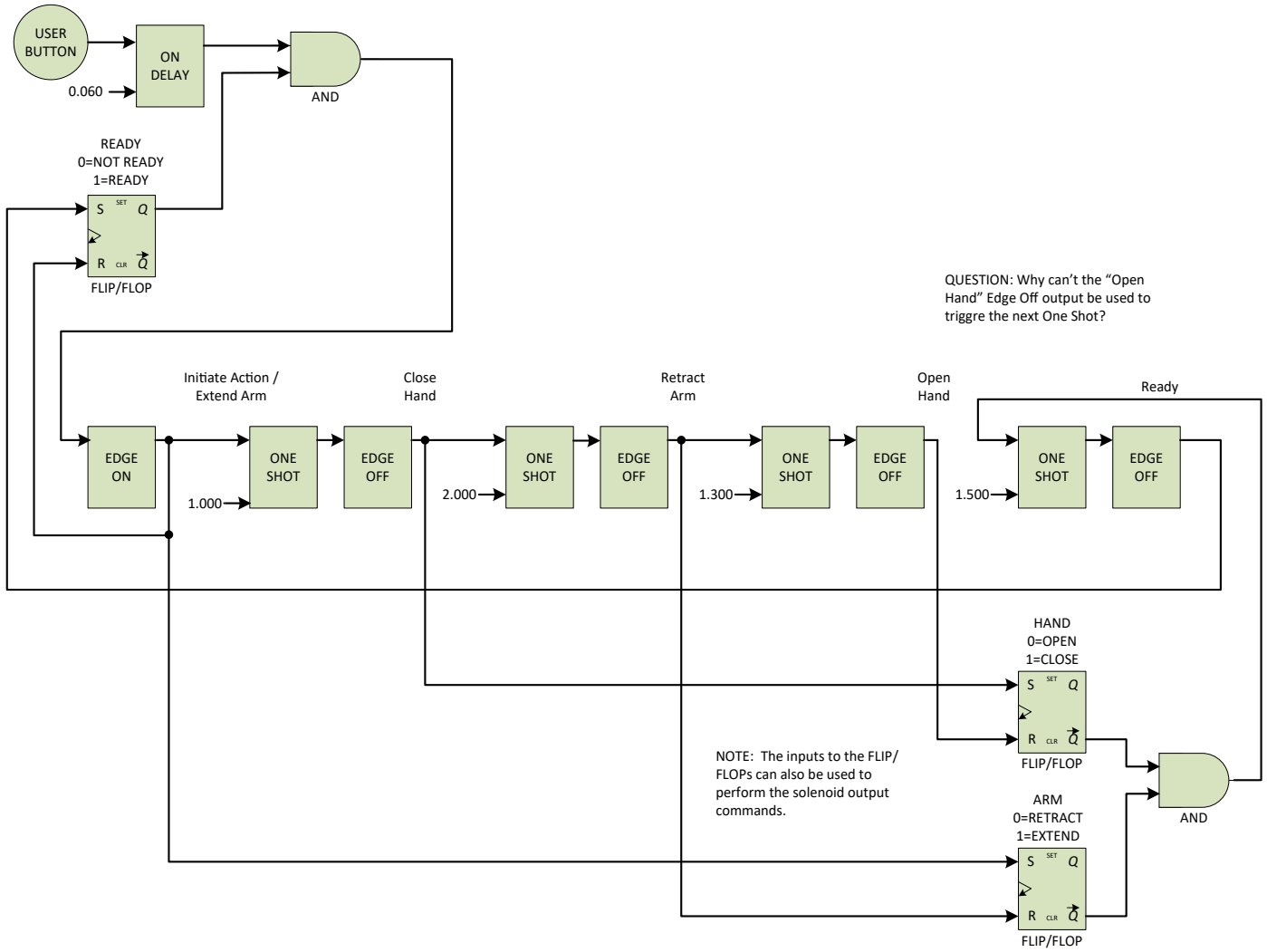
Designing Sequential Logic 4/4

- **Final Logic Diagram**



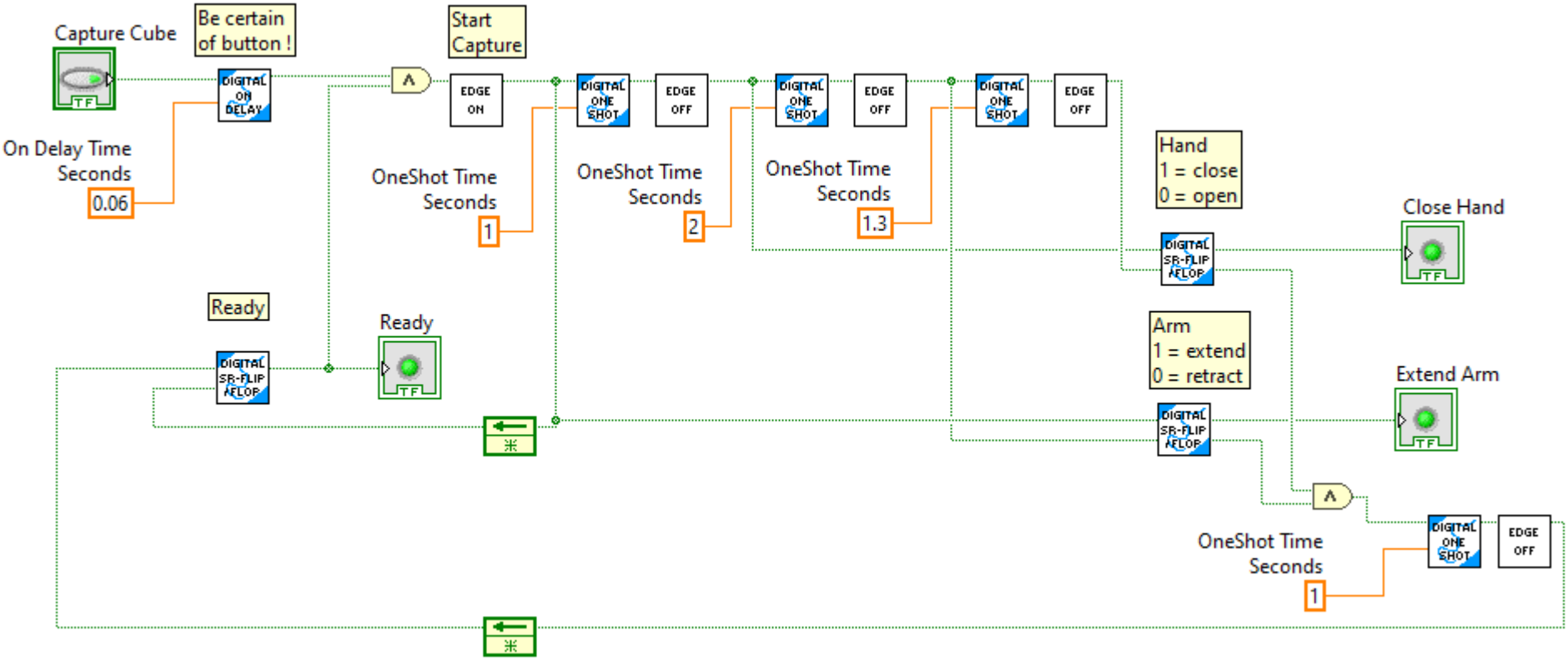
Designing Sequential Logic 4/4

- **Alternate Final Logic Diagram**



Designing Sequential Logic 4/4 - LabVIEW

- **Sample LabVIEW logic**



Exercise 5.1 – Shoot Flying Disc

- **User pushes a button to shoot frisbee.**
- **Ensure user meant to push button. Button must be pressed for three cycles before initiating action. (Cycle time is 0.020 seconds).**
- **Can only shoot a frisbee if we have one. A limit switch indicates this. Also battery voltage must be > 11.5 volts. Can only shoot one frisbee at a time.**
- **Motors take 3 seconds to spin up to speed.**
- **Engage solenoid for 2 second to push frisbee into shooting wheel.**
- **Allow 2 more seconds for shooting to occur.**
- **After shooting is done, stop motor. (For now, don't allow continuous shooting.)**
- **Allow user to press a Cancel button. The cancel button must be pressed for at least 3 cycles before becoming active. After the Cancel, force a 5 second reset before allowing a new shot.**
- **It takes 5 seconds after shooting for a new frisbee to be in place ready to shoot.**
- **Design shooting logic. Also provide "ready to shoot" digital for dashboard display. Use ONLY the algorithms discussed in this module, and perhaps module 4.**

Robot Programming 02

- **Complete Robot Programming Training 02**

Exercise 5.2 – Shoot Flying Disc Robot Code

- **Implement Exercise 5.1 in robot code.**