LECTURE NOTES PART 8
ET 438B  Sequential Control and Data Acquisition
A simplified starting timer is to be constructed for a drag strip. To enable the start timing for a race both cars must actuate sensor switches at the start line that indicate they are in position. When the cars are in position, the race judge receives a green light on his control panel and a green light comes on of the “Christmas Tree”. He then presses a race initial button on his control panel. The Christmas tree times through the sequence shown at left.
DESIGN EXAMPLE - DRAG STRIP "CHRISTMAS TREE"

When the lower pair of green lamps come on the racers begin. A pair of photo eyes located at the finish line indicate the winner by lighting a blue light for a winner and a amber light for a loser. After the race results are indicated, the judge can press a reset button to prepare the system for the next race.
Design Example - Drag Strip
"Christmas Tree"

Design Problem
1.) Identify the states, conditions and actions for this system
2.) Construct a flow chart of the logic for this system
3.) Construct a state transition diagram for this system
4.) Design a ladder logic system to implement these functions

Part 1: States, Conditions, Actions

States
S0: reset
S1: cars at start line
S2: 1st set red lamps on
S3: 2nd set red lamps on
S4: 3rd set red lamps on
S5: Green lamps on (race start)
S6: Lane 1 Wins
S7: Lane 2 Wins
Define conditions (Inputs)

Conditions
10: reset pressed
11: racer 1 positioned
12: racer 2 positioned
13: race timing initiated
14: 1st set red lamps timed out
15: 2nd set red lamps timed out
16: 3rd set red lamps timed out
17: lane 1 finish photo eye tripped
18: lane 2 finish photo eye tripped
19: start pressed
Define actions (Outputs)

Actions

O0: light green ready lamp
O1: light red lamps set 1
O2: light red lamp set 2
O3: light red lamp set 3
O4: light green lamp set
O5: light blue lamp 1 if lane 1 wins
O6: light blue lamp 2 if lane 2 wins
O7: light amber lamp 1 if lane 1 loses
O8: light amber lamp 2 if lane 2 loses
Construct State Equations

Start pressed 19

State 0

S0⁻¹ = (S0 + I0 ∙ (S6 + S7) + I9) ∙ (I1 ∙ I2 ∙ S0)
S0⁻¹ = (S0 + I0 ∙ (S6 + S7) + I9) ∙ (I1 + I2 + S0)

Expanding and Simplifying

S0⁻¹ = S0 ∙ I1 + S0 ∙ I2 + S0 ∙ S0 + (I0 ∙ (S6 + S7) + I9) ∙ I1 + (I0 ∙ (S6 + S7) + I9) ∙ I2 + (I0 ∙ (S6 + S7) + I9) ∙ S0

Factor

S0⁻¹ = S0 ∙ (I1 + I2) + I9 ∙ (I1 + I2) + I0 ∙ (S6 + S7) ∙ (I1 + I2) + I0 ∙ (S6 + S7) ∙ S0 + I9 ∙ S0

Regroup

S0⁻¹ = S0 ∙ (I1 + I2) + [I0 ∙ (S6 + S7) + I9] ∙ (I1 + I2) + [I0 ∙ (S6 + S7) + I9] ∙ S0
Construct State Equations

State 0 continued

Factor

\[ S_0^{-1} = [S_0 \cdot (I_1 + I_2) + (10 \cdot (S_6 + S_7) + 19)](I_1 + I_2)(I_1 + I_2 + S_0) \]

Simplify & Regroup

\[ S_0^{-1} = (I_1 + I_2) \cdot [S_0 + (10 \cdot (S_6 + S_7) + 19)] \cdot [1 + S_0] \]

\[ S_0^{-1} = (I_1 + I_2) \cdot [S_0 + (10 \cdot (S_6 + S_7) + 19)] \]

\[ S_0^{-1} = (S_0 + 10 \cdot (S_6 + S_7) + 19)(I_1 + I_2) \]  
Reduced Equation
Construct Ladder Logic

\[ S_0^{+1} = (S_0 + I_0 \cdot (S_6 + S_7) + I_9) \cdot (I_1 + I_2) \]
Construct State Equations

\[ S_{I}^{+1} = (S_{I} + \Pi \cdot I2 \cdot S0) \cdot (\overline{I3} \cdot S_{I}) \]
\[ S_{I}^{+1} = (S_{I} + \Pi \cdot I2 \cdot S0) \cdot (\overline{I3} + S_{I}) \]

\[ S_{I}^{+1} = S_{I} \cdot \overline{I3} + S_{I} \cdot S_{I} + \Pi \cdot I2 \cdot S0 \cdot \overline{I3} + \Pi \cdot I2 \cdot S0 \cdot S_{I} \]
\[ S_{I}^{+1} = S_{I} \cdot \overline{I3} + \Pi \cdot I2 \cdot S0 \cdot (\overline{I3} + S_{I}) \]

\[ S_{I}^{+1} = S_{I} \cdot \overline{I3} + \Pi \cdot I2 \cdot S0 \cdot (\overline{I3} \cdot \Pi \cdot I2 \cdot S0 + S_{I} \cdot \Pi \cdot I2 \cdot S0) \]
\[ S_{I}^{+1} = S_{I} \cdot \overline{I3} + (\overline{I3} \cdot \Pi \cdot I2 \cdot S0) \cdot S_{I} \]
\[ S_{I}^{+1} = S_{I} \cdot (\overline{I3} + (\overline{I3} \cdot I2 \cdot S0) \cdot S_{I}) \]
Construct Ladder Logic

State 1 Output Equation

\( S_1^{+1} = (S_1 + I_1 \cdot I_2 \cdot S_0)(I_3) \)

\( 00 = S_1 \)
Construct State Equations

State 2
\[ S_{2}^{+1} = (S_2 + I_3) \cdot (I_4) \]

TON\#(c,t) = On-delay timer #
c = input condition
t = time delay

SO
\[ I_4 = TON_l(S_2,3) = \text{timer done} \]

\[ S_{2}^{+1} = (S_2 + I_3) \cdot (TON_l(S_2,3)) \]

\[ S_{2}^{+1} = (S_2 + I_3) \cdot (TON_l(S_2,3)) \]

State 2 Output Equation
\[ O_1 = S_2 \]

State 3
\[ S_{3}^{+1} = (S_3 + I_4 \cdot S_2) \cdot (I_5) \]

\[ I_5 = TON_2(S_3,3) = \text{timer done} \]

\[ S_{3}^{+1} = (S_3 + TON_l(S_2,3) \cdot S_2) \cdot (TON_2(S_3,3)) \]

State 3 Output Equation
\[ O_2 = S_3 \]
Construct State Equations

State 4

\[ S_{4+1} = (S_4 + 16 \cdot S_3) \cdot \overline{(16)} \]

\[ I_6 = TON_3(S_4, 3) = \text{timer done} \]

\[ S_{4+1} = (S_4 + TON_2(S_3, 3) \cdot S_3) \cdot \overline{(TON_3(S_4, 3))} \]

State 4 Output Equation \[ O_3 = S_4 \]

State 5

\[ S_{5+1} = (S_5 + 16 \cdot S_4) \cdot \overline{(18 + 19)} \]

\[ I_6 = TON_3(S_4, 3) = \text{timer done} \]

\[ S_{5+1} = (S_5 + TON_3(S_4, 3) \cdot S_4) \cdot \overline{(17 + 18)} \]
Construct State Equations

State 5 simplification

\[ S_5^{+1} = (S_5 + TON3(S4,3) \cdot S4) \cdot (I7 + I8) \]
\[ S_5^{+1} = (S_5 + TON3(S4,3) \cdot S4) \cdot (I7 \cdot \overline{I8}) \]

State 5 Output Equation \[ O4 = S5 \]

State 6

\[ S_6^{+1} = (S_6 + I7 \cdot \overline{I8} \cdot S5) \cdot (\overline{I0}) \]

State 6 Output Equations
\[ O5 = S6 \]
\[ O8 = S6 \]
Construct State Equations

State 7

S0
Reset pressed

I0

S7

I8

O6, O7 Light correct lamps
Lane 2 wins

S7^1 = (S7 + I7 \cdot I8 \cdot S5) \cdot (I0)

Block Lane 1

State 7 Output Equations

O6 = S7
O7 = S7
Construct Ladder Logic

State 6

\[ S_{6+1} = (S6 + 17 \cdot \bar{S}5) \cdot (\bar{I0}) \]

O5 = S6
O8 = S6
Construct Ladder Logic

State 7

\[ S_7^{*1} = (S_7 + \overline{17} \cdot 18 \cdot S_5) \cdot (\overline{10}) \]

O6 = S7
O7 = S7

Lane 2 Wins
Design Considerations

Personnel and Equipment Safety

**Fail-safe operation** - component fail results in little or no damage or inconvenience

**Common Practice**
1.) start sequence by closing NO contacts
2.) stop a sequence by opening NC contact

Practice Results: if device fails to start process stops, If started would stop **Example: Motor starter**
Troubleshooting Tips

Common causes of Failure
1.) Dirty or oxidized contacts
2.) Broken wire or loose connection
3.) Up stream power source interrupted causing input device to de-activate

Other Issues

Contact operation sequence
break-before-make standard
See previous example limit switches
Programmable Logic Controllers (PLC)

Microprocessor-based controller that implements ladder logic through software and hardware interface.

Definition of PLC
Digital apparatus using programmable memory and stored programs for implementing Logic Timing Sequencing Counting

Basic Block Diagram of A PLC

Field devices (sensors) → INPUT MODULES → CPU → OUTPUT MODULE → Field devices

USER: Programmer ↔ Memory

(control values, solenoids)
PLC Operation in Run Mode

1. Scan all Inputs
   Detect change in status of field devices (Limit switches Pressure switches, etc.)

2. Execute control program based on user logic design

3. Test output status against program values.

4. Update output to fit changes dictated by input change

Time from 1 to 4 called scan time. Can be important in programs.
Typical PLC Input/Output Modules

PLC I/O designed to connect to industrial devices

Dry contacts (standard switches)
- motor contactors
- Pressure, temperature, limit, flow switches

Solid state sensors (electronic)
- proximity switches, photo eyes etc.

Voltage levels
- 24 - 240 Vac
- 24-240 Vdc
- TTL levels, Sourcing and sinking I/O

1 set terminals (com, n) = 1 I/O point

I/O grouped on cards with 8, 16, 24, 32 inputs
PLC I/O Interfacing

PLC’s use memory mapped I/O

PLC uses microprocessors with 8-16 bit words. Each I/O point identified by location in memory. Terminals have unique addresses represented by decimal, octal or binary number. (commonly decimal)

![Diagram showing memory mapping and bit representation for I/O points]

- Each memory word maps to group of I/O points
- Bits represent status of I/O field devices
- 1=on, 0=off
PLC I/O Interfacing

For expandable PLCs

Level 1: Rack or chassis identifier

Level 2: Slot identifier (type of I/O card)

Level 3: I/O point. Type of point and terminal number

Non-expandable PLCs use fixed addressing: All slot 0

I/O Status Table
PLC I/O Interfacing

Addressing of I/O cards (Allen-Bradley (A-B))
A-B uses decimal expandable addressing for most PLCs

Function I/O

Addressing Specific I/O points (Allen-Bradley)

Function I/O

For fixed PLC designs, all I/O addressed to slot 0
### PLC I/O Interfacing

#### I/O Addressing Examples

<table>
<thead>
<tr>
<th>Function</th>
<th>Chassis Slot</th>
<th>Terminal #</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>1</td>
<td>4</td>
<td>I:1/4</td>
</tr>
<tr>
<td>output</td>
<td>2</td>
<td>13</td>
<td>O:2/13</td>
</tr>
<tr>
<td>input</td>
<td>3</td>
<td>2</td>
<td>I:3/2</td>
</tr>
<tr>
<td>input</td>
<td>4</td>
<td>4</td>
<td>I:4/4</td>
</tr>
</tbody>
</table>

Note: Decimal addressing used above

Other PLC data types:
- Bit data, unsigned integer, signed integer, BCD
- (binary coded decimal 0-9 binary)
Input Modules

Sequential control uses discrete (binary) inputs (on/off) from field devices (switches sensors, etc.)

Typical Type of Input Modules

**Ac**
- 24, 48, 120, 240 V
- 120, 240 V isolated
- 24 Vac/dc

**Dc**
- 24, 48, 1-60 120 Vdc
- sink/source 5-50 Vdc
- 5 V TTL
- 5/12 V TTL

Input module considered load of field device (switch)
Input Module Specifications

<table>
<thead>
<tr>
<th>Voltage</th>
<th>8 to 122 VAC</th>
<th>105 to 230 VAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Points per Common</td>
<td>0.025 Amps</td>
<td>0.025 Amps</td>
</tr>
<tr>
<td>Backplane Current Draw at 5 VDC</td>
<td>3mA</td>
<td>3mA</td>
</tr>
<tr>
<td>Maximum Signal Delay</td>
<td>2ms</td>
<td>2ms</td>
</tr>
<tr>
<td>Maximum Off-State Current</td>
<td>3mA</td>
<td>3mA</td>
</tr>
<tr>
<td>Input Current Nominal</td>
<td>12mA at 120 VAC</td>
<td>12mA at 240 VAC</td>
</tr>
<tr>
<td>Maximum初级 Current</td>
<td>0.15A</td>
<td>0.15A</td>
</tr>
</tbody>
</table>

Figure 6-7 AC input module specifications for Allen-Bradley SLC 500 120 VAC and 240 VAC input modules. (Compiled from Allen-Bradley Discrete I/O modules data.)

Explanation of Specifications

**Backplane draw current** - module current drawn by electronics

**Maximum signal delay** - time required for PLC to sense change in field device and store it in memory

**Maximum off state current** - max current that can flow so that input remains in off state. (leakage from solid state sensors)

**Nominal input current** - current drawn by the input point with nominal voltage applied
Input Module Specifications

Additional Specifications

Voltage Drop
Leakage Current
Load Current
Power-up Delay

Useful when applying active (solid-state) switches and proximity sensors

Leakage Current

1.7-3.5 mA must flow to keep sensor active (2 wire sensor)

Voltage Drop

V_s must be low
Typically 6-10 for 2-wire solid state sensor
Inputs with Solid State Sensors

When solid-state 2-wire sensor is used with switch, sensor will be inactive until circuit is completed

Power-up Delay

Dealing with power up delay - add parallel resistor.

Must allow enough current to activate sensor but not turn on input module.
Inputs with Solid State Sensors

Dealing with power up delay - add parallel resistor.

Example: size $R_v = 115 \text{ Vac}$
$I_v = 1.7 \text{ mA}$

$R = \frac{115 \text{ V}}{1.7 \text{ mA}} = 67,647 \text{ k}\Omega$
**Inputs with Solid State Sensors**

**Minimum load current** - lowest I value that keeps the sensor active

May need to parallel a resistor with the input card if it has a high impedance input or sensor needs more current than card can handle without turning on the input.

R called bleeder resistor. Usually sized according to manufacturer charts. Based on concept of current division.