Lesson 15: Boolean Representation of Ladder Diagrams

ET 438B Sequential Control and Data Acquisition
Department of Technology

Learning Objectives

After this presentation you will be able to:

- Realize logic functions as ladder diagram rungs
- Follow the logic of a multi-rung ladder diagram
- Represent ladder rungs as Boolean gates
- Design combinational sequential controllers using Boolean equations

Ladder Diagram Example

A manual mixing operation is to be automated using sequential process control methods. The process composed of three steps:

a.) filling a tank to a predetermined level
b.) agitating the liquid for 30 minutes
c.) draining the tank for use in another part of process

Does the ladder logic schematic that follows perform this function correctly?

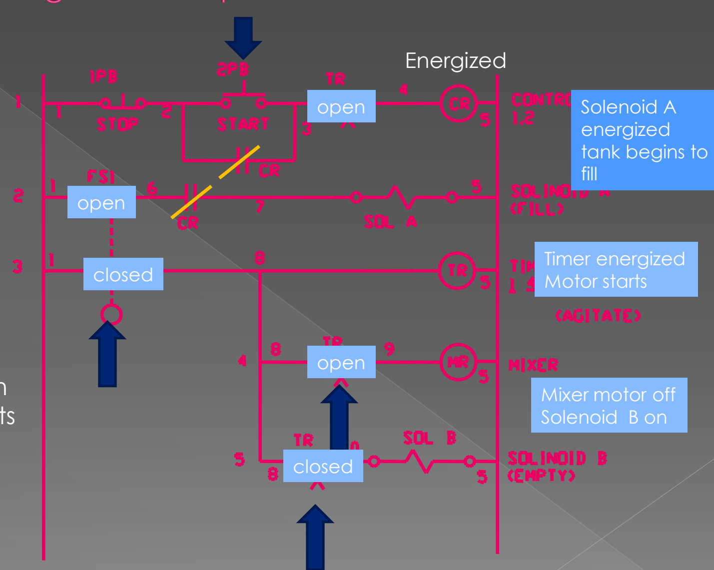
Ladder Diagram Example

Press start

Tank fills to limit

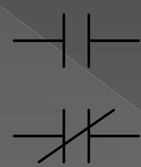
Timer goes To 30 minutes

When tank drains flow switch resets. Timer resets



Combinational and Sequential Logic with Relays and Contacts

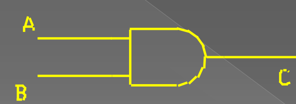
Let contact state represent a logical value



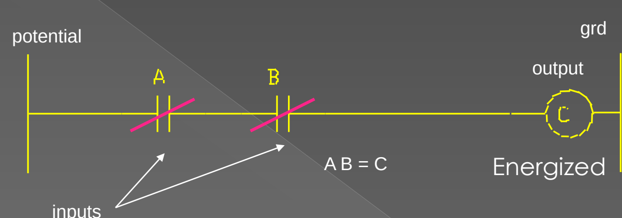
= Logic 0 Called Form A Contact

= Logic 1 Called Form B Contact

Implement AND gate



Combinational and Sequential Logic with Relays and Contacts



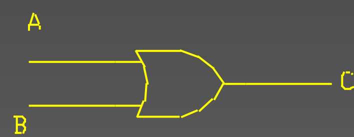
Conditions A **AND** B must be present to energize output C

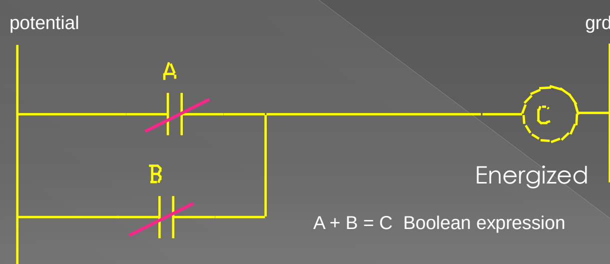
Note: all contacts are considered instantaneous and not held unless modified

With electromechanical relays fan-in and fan-out limited by number of contacts in relays

More Logic Functions

OR Function





A + B = C Boolean expression

Either A **OR** B will cause coil C to be energized
Contacts A, B represent conditions or states in the sequential process

More Logic Functions

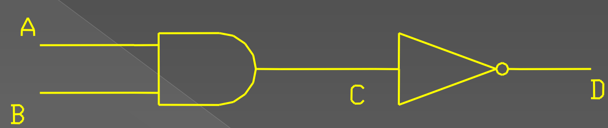
NOT Function

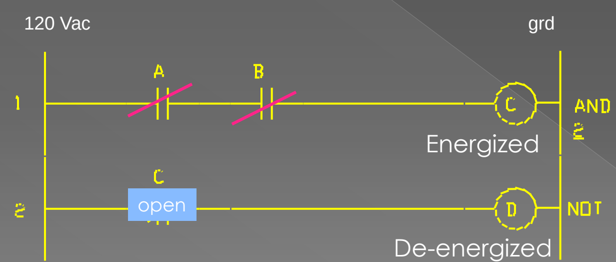
Boolean Expression
$B = \overline{A}$

Contact of opposite state creates inversion

Constructing Other Logic Functions

Combine the **AND** function with the **NOT** function to get a **NAND** operation.

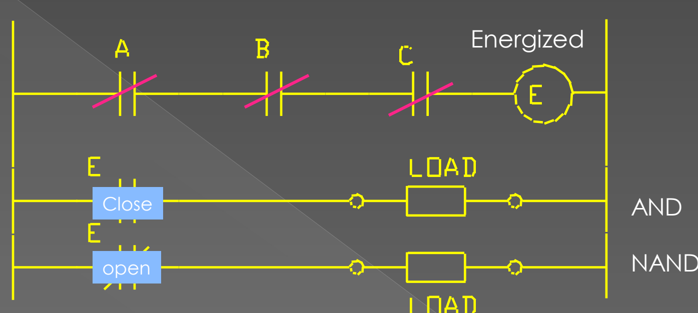




Any contact associated with coil D will change state like a NAND TTL gate.

Rung 1 implements the AND function
Rung 2 implements the NOT function

Multiple Input AND/NAND



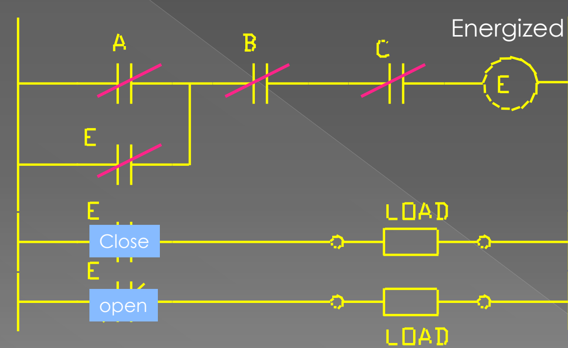
$ABC = E$ and $\overline{ABC} = E$

Can add a memory action to the above by including a feedback from the output coil to the inputs

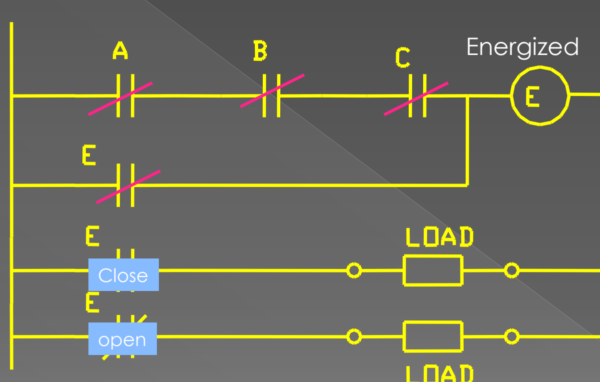
Memory Action AND/NAND

Can add a memory action to the above by including a feedback from the output coil to the inputs

B and C are not sealed

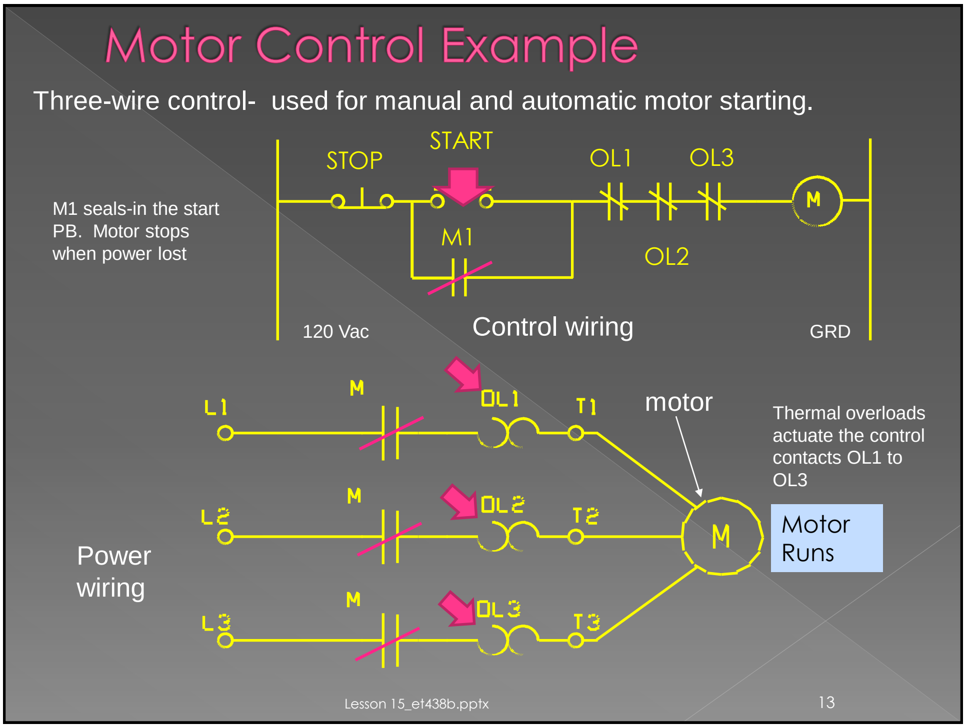


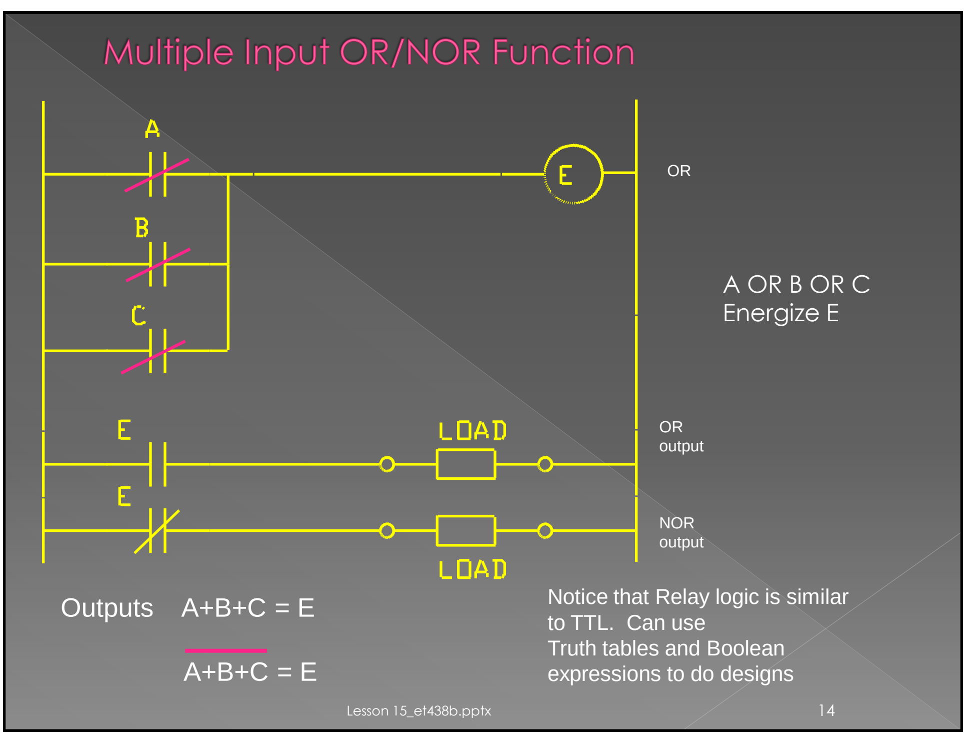
All Inputs Latched AND/NAND



The output can not change unless the circuit is de-energized.

Contact E in rung 2 is a feedback from the output that makes circuit ignore state changes of A, B and C after the condition A B C is detected.

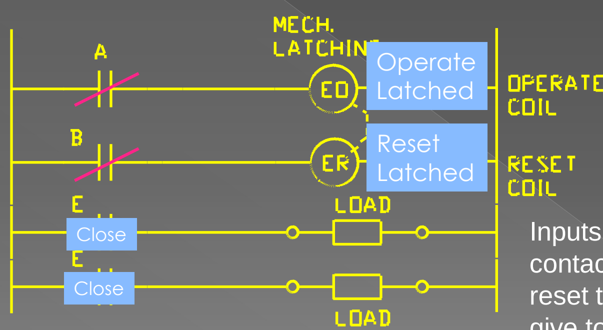

Motor Control Example
Three-wire control- used for manual and automatic motor starting.
M1 seals-in the start PB. Motor stops when power lost
STOP
START
OL1
OL3
M
M1
OL2
120 Vac
Control wiring
GRD
L1
L2
L3
M
M
M
OL1
OL2
OL3
T1
T2
T3
motor
Thermal overloads actuate the control contacts OL1 to OL3
M
Motor Runs
Power wiring
Lesson 15_et438b.pptx
13


Multiple Input OR/NOR Function
A
B
C
E
OR
A OR B OR C Energize E
E
LOAD
OR output
E
LOAD
NOR output
Outputs A+B+C = E
$\overline{A+B+C} = E$
Notice that Relay logic is similar to TTL. Can use Truth tables and Boolean expressions to do designs
Lesson 15_et438b.pptx
14

Ladder Logic Memory Elements

Mechanically latched relay - maintains state even when power removed. Has two coils (operate, reset)

Typical wiring



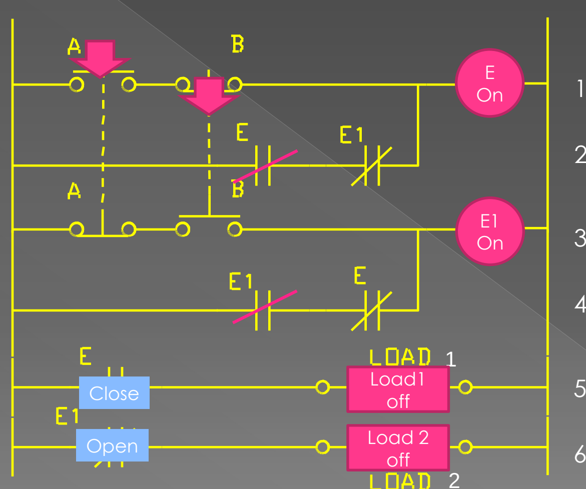
Inputs A and B set the output contacts E and reset then respectively. This give toggle action that “remembers” the last input state even when power is removed

Typical Applications
Reversing Motor starters. Reclose Relay Cut-out

Lesson 15_et438b.pptx 15

Off-Return Memory

Energize and re-energize circuit - Load 2 on
No continuity in rungs 1-4
Continuity in rung 6



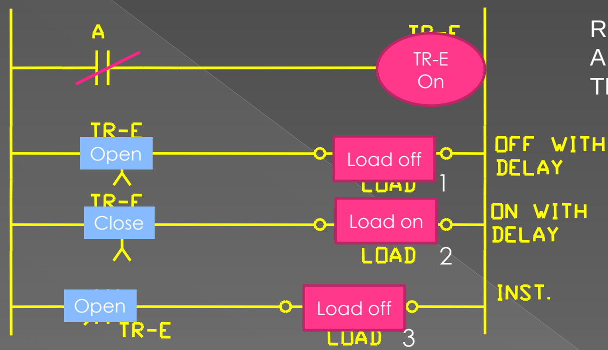
Press A: continuity rung 2 Both loads on

Press B: continuity rung 4 Both loads off

Remember all contacts are drawn with the coils de-energized

Lesson 15_et438b.pptx 16

Timer Sub-Circuits



Rung 1: when input A is energized timer TR-E starts

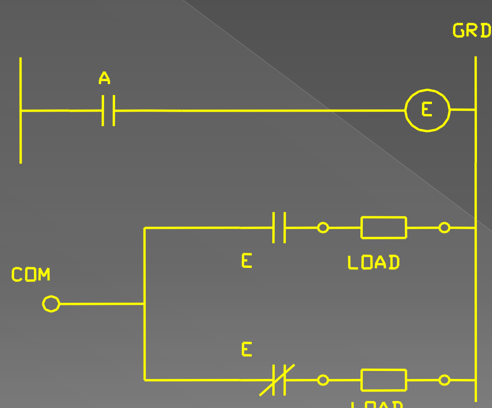
Schematic indicates that this is a on-delay timer. After defined interval TR-E in rung 2 opens and TR-E in rung 3 closes

Load 1 is deactivated after time delay
Load 2 is activated after time delay

Load 3 is instantaneously deactivated by TR-E

Form "C" Contact

Loads are toggled between a common point



Typical "Form C" contacts include both a NO and NC contact arrangement.

Used in some sensors for more flexibility

Contact A creates a remote control toggle switch

Designing Sequential Control Systems

Combinational Systems

- Use true tables, Boolean Algebra
- Multiple inputs and/or outputs
- Sum of Products or product of sums Boolean Implementations
- Reduce to minimum implementation

Sequential Systems

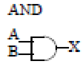
- Follow steps, transition from one step to another.
- Use state transition diagrams or tables with Boolean Algebra
- State Machine implemented in software or hardware
- Decisions made base on current condition of system and input information

Review of Logic Gates and Boolean Algebra

Boolean Variables False =0 True =1

Boolean Operators

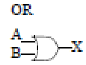
AND



$X = A \cdot B$

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

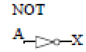
OR



$X = A + B$

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

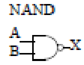
NOT



$X = \bar{A}$

A	X
0	1
1	0

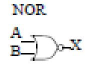
NAND



$X = \overline{A \cdot B}$

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

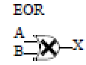
NOR



$X = \overline{A + B}$

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

EOR



$X = A \oplus B$

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

EOR=XOR

Alternate Implementation

$X = A\bar{B} + \bar{A}B$

Review of Logic Gates and Boolean Algebra

Axioms of Boolean Algebra

Idempotent

$A + A = A$

$A \cdot A = A$

Associative

$A + B) + C = A + (B + C)$

$A \cdot B) \cdot C = A(B \cdot C)$

Distributive

$A + (B \cdot C) = (A + B)(A + C)$

$A(B + C) = (A \cdot B) + (A \cdot C)$

Identity

$A + 0 = A$

$A + 1 = 1$

$A \cdot 0 = 0$

$A \cdot 1 = A$

Complement

$A + \overline{A} = 1$

$A \cdot \overline{A} = 0$

$\overline{\overline{A}}) = A$

$\overline{1} = 0$

DeMorgan's Theorem

$\overline{A + B)} = \overline{A} \cdot \overline{B}$

$\overline{A \cdot B)} = \overline{A} + \overline{B}$

Absorption

$A + \overline{A} \cdot B = A + B$

$A + A \cdot B = A$

Order of Operations
1. NOT
2. AND
3. OR

Review of Logic Gates and Boolean Algebra

Example: Simplify the following expression using the axioms of Boolean Algebra.

$X = (\overline{A + B \cdot C}) + A(B + \overline{C})$

Add Parentheses

$X = (\overline{A) + (B \cdot C}) + A(B + \overline{C})$

Apply DeMorgans's Theorem to first term

$\overline{A}(\overline{B \cdot C}) = (\overline{A) + (B \cdot C)}$

Apply DeMorgan's Here

$X = \overline{A}(\overline{B \cdot C}) + A(B + \overline{C})$

$\overline{B \cdot C}) = (\overline{B} + \overline{C})$

$X = \overline{A}(\overline{B} + \overline{C}) + A(B + \overline{C})$

Expand Expressions

$X = \overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{C} + A \cdot B + A \cdot \overline{C}$

Collect common terms and factor

$\overline{C}(A + \overline{A}) = A \cdot \overline{C} + \overline{A} \cdot \overline{C}$

Review of Logic Gates and Boolean Algebra

Example Continued

$$X = \overline{A} \cdot \overline{B} + A \cdot B + \overline{C}(A + \overline{A})$$

Use Complement Axiom

$$A + \overline{A} = 1$$

$$X = \overline{A} \cdot \overline{B} + A \cdot B + \overline{C} \cdot 1$$

Use Identity Axiom

$$\overline{C} \cdot 1 = \overline{C}$$

$$X = \overline{A} \cdot \overline{B} + A \cdot B + \overline{C}$$

Simplified Expression

Logic Design

1.) Obtain description of process
2.) Define control action
3.) Define Inputs and Outputs
4.) Develop Truth Table or Boolean Equation of Process

Process control description

A heating oven with two bays can heat one ingot in each bay. When the heater is on it provides enough heat for two ingots. If only one ingot is present, the oven may overheat so a fan is used to cool the oven when it exceeds a set temperature.

Control Action

When only one ingot is in the oven and the temperature exceeds the setpoint, turn on the fan

Logic Design

Define I/O variables

Inputs: B1 = bay1 ingot present
B2 = bay2 ingot present
T = temperature sensor

Output: F= fan start

Create Truth Table

T	B2	B1	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

If there is no over temperature don't start the fan

Over temperature in empty oven: safety fan start

Start fan in lightly load ovens with over temp.

Over temperature in full oven: safety fan start

Logic Design

Select elements from truth table in SOP (sum-of-products) form then simplify.

T	B2	B1	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$$F = T\cdot\overline{B1}\cdot\overline{B2} + T\cdot B1\cdot\overline{B2} + T\cdot\overline{B1}\cdot B2 + T\cdot B1\cdot B2$$

$$F = T\cdot(\overline{B1}\cdot\overline{B2} + B1\cdot\overline{B2} + \overline{B1}\cdot B2 + B1\cdot B2)$$

$$F = T\cdot(\overline{B2}\cdot(\overline{B1} + B1) + B2\cdot(\overline{B1} + B1))$$

$$F = T\cdot(\overline{B2} + B2)$$

$$F = T$$

Requires only Temp control

Ignore unloaded and full load cases and try again

Logic Design

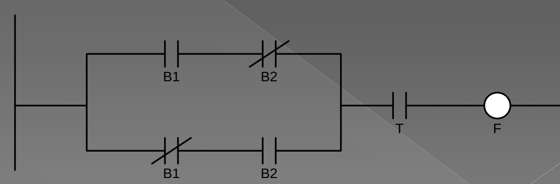
$$F = T \cdot B1 \cdot \overline{B2} + T \cdot \overline{B1} \cdot B2$$

Revised Truth Table

T	B2	B1	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	**<u>0</u>**
1	0	1	1
1	1	0	1
1	1	1	**0**

$$F = T(B1 \cdot \overline{B2} + \overline{B1} \cdot B2)$$

Ladder Logic Representation

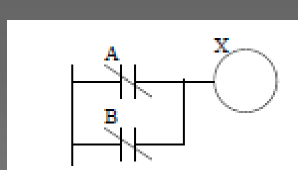


Simplified Forms of Functions

Avoid multiple complemented variables in ladder logic (No NAND, NOR)

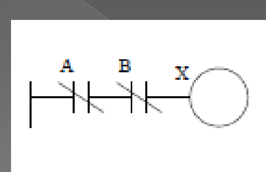
NAND

$$X = \overline{A \cdot B} = \overline{A} + \overline{B}$$



NOR

$$X = \overline{A + B} = \overline{A} \cdot \overline{B}$$



NAND/NOR can not be implemented effectively using software. (Programmable Logic Controllers)

End Lesson 15: Boolean Representation of Ladder Diagrams

ET 438B Sequential Control and Data Acquisition
Department of Technology

29

Lesson 15_et438b.pptx