Chapter 3 Logic Concept

Contents

၃.၁ The Binary concept 2
၃.၂ Logic Function 2
၃.၂.၁ "AND" Function 3
၃.၂.၂ The "OR" Function 4
၃.၃ Principle of Boolean and Logic........ 9
၃.၃.၁ AND operation........10
၃.၃.၂ OR operation........10
၃.၃.၃. Not Function 10
၃.၄ Order of Operation နှင့် Group Sign........12
၃.၄ PLC circuit နှင့် logic contact သင်္ကေတများ 14
၃.၅ PLC ၌အသုံးပြုသည့် Address များ........17
၃.၆ Contact symbols used in PLC........19

DRAFT

Logic Concept

PLC များ၏အလုပ်လုပ်ပုံ နှင့် အသုံးပြုပုံများကို နားလည်ရန်အတွက် Logic concept ကို ပထမဦးစွာ နားလည်ရန် လိုအပ်သည်။ အခြေခံ Logic function ၃ မျိုးဖြစ်သည့် AND ၊ OR နှင့် NOT တို့ကို ကွဲပြားအောင် Electric Circuit များ နှင့် Ladder Diagram များနှင့်တကွ သေးစိတ်ရှင်းလင်းဖော်ပြထားသည်။ ထို Logic function ၃ မျိုးကိုအခြေခံ၍ ရိုးရှင်းလွယ်ကူသည့် Control decision များမှစ၍ ခက်ခဲ ရှုပ်ထွေးသည့် Control Decision များအထိ တည်ဆောက်ထားသည်။ နောက်ပိုင်းတွင် Boolean algebra ၏ အခြေခံ သဘောတရားများ (fundamental) နှင့် ယင်းတို့နှင့်သက်ဆိုင်သည့် Operator များကိုဖော်ပြထားသည်။ ထို့နောက် Boolean Logic နှင့် Logic contact Symbology တို့၏ ဆက်သွယ်ပုံများကို ရှင်းလင်းထားသည်။ ဤအခန်းကိုကျေညက်စွာလေ့ လာပြီးလျှင် PLC processor များနှင့် ယင်းတို့၏ programming Device များကို ဆက်လက်လေ့လာနိုင်လိမ့် မည်။

၃.၁ The Binary concept

Binary concept သည် အိုင်ဒီယာအသစ်တစ်ခု မဟုတ်ပါ။ Binary concept ကို ရှေးယခင်ကတည်းက စတင်သုံးစွဲခဲ့ကြသည်။ ဥပမာ-မီးလုံး (သို့) မီးချောင်း(Light)သည် ပွင့် (ON) နေနိုင်သည်။ ပိတ် (OFF) နေနိုင်သည်။ ခလုပ်(switch) သည် Open ဖြစ် နေနိုင်သည်။ Close ဖြစ်နေနိုင်သည်။ မော်တာ (motor) တစ်လုံးသည်မောင်း(running) နေနိုင်သည်။ ရပ်(stop) နေနိုင်သည်။ Digital System များတွင် အခြေအနေ ၂ မျိုး(two state condition) သာရှိနိုင်သည်။ signal ရှိခြင်း (Signal activated) (သို့) မရှိခြင်း (not activated)၊ high (သို့) Low ၊ ON (သို့) OFF ၊ Open (သို့) Close စသည့်တို့ကဲ့သို့ အခြေအနေ ၂ မျိုးအနက်မှ တစ်ချိန်တွင် တစ်မျိုးသာဖြစ်နေနိုင်သည်။ ထို Two State concept (Binary concept) သည် Decision များချရန် အတွက် အခြေခံဖြစ်သည်။ အခန်း ၂ တွင်ဖော်ပြထား သည် Binary Number System ၏သဘောများကို အခြေခံသည်။ Binary Number System (သို့) Binary concept သည် PLC များ၏ function Block များပြုလုပ်ရန် နှင့် Digital Computer များ၏ အခြေခံ(fundamental) သဘောတရားများ ဖြစ်သည်။

ဤစာအုပ်တွင် Binary 1 သည် Signal ရှိခြင်းကို ဆိုလိုသည်။ Binary 0 သည် Signal မရှိခြင်း (absence of signal) ကိုရည်ညွှန်းသည်။ Digital System များတွင် ဤအခြေအနေ ၂ မျိုး(two state) ကို မတူညီသည့် Voltage Level ၂ ခုဖြင့် ဖော်ပြသည်။ +V volt နှင့် 0V volt တို့ကို Table 3-1 တွင် ဖော်ပြထားသည်။ Volt တစ်ခုသည် တခြား Volt တစ်ခုထက် အပေါင်းဓါတ်အား (သို့) အဖိုဓါတ်အား (more positive) ပိုများသည်။ တစ်ခါတရံ Binary 1 (သို့) Logic 1 ကို TRUE ၊ ON ၊ HIGH စသည် အဓိပ္ပါယ်များ အဖြစ် ရည်ညွှန်းပြောဆိုပြီး Binary 0 (သို့) Logic 0 ကို FALSE ၊ OFF ၊ LOW စသည့် အဓိပ္ပါယ်များ အဖြစ်ရည်ညွှန်း ပြောဆိုသည်။

Logic 1 (+V)	Logic 0 (0V)	Example
Operating	Not operating	Limit switch
Ringing	Not ringing	Bell
On	Off	Light bulb
Blowing	Silent	Horn
Running	Stopped	Motor
Engaged	Disengaged	Clutch
Closed	Open	Valve

Table 3-1. Binary concept using positive logic.

၃.၂ Logic Function

Table 3-1 တွင်ဖော်ပြထားသည့် more Positive Voltage ကို Logic 1 အဖြစ်သတ်မှတ်သည်။ ပိုနည်းသည့် (Less positive voltage)ကို Logic 0 အဖြစ်သတ်မှတ်ထားသည်။ မိမိနှစ်သက်သလို အဆင်ပြေသလို ရွေးချယ်သတ်မှတ်ထားခြင်း ဖြစ်သည်။

Table 3-2 တွင် Logic 0 ကို positive voltage level အဖြစ် ရည်ညွှန်းသတ်မှတ်သည်။ Occurrence of event တစ်ခုရှိခြင်းကို Logic 0 ဖြင့်သတ်မှတ်သည်။ Non Occurrence of event ကို Logic 1 ဖြင့်သတ်မှတ်သည်။ Less positive voltage level ဖြင့်သတ်မှတ်သည်။ Table 3-1 မှ Positive Logic သည်

DRAFT

အသုံးပြုနေကျပုံစံ (conventional) မျိုးဖြစ်သည်။ သို့သော် Negative logic သည် application များအတွက် အဆင်ပြေ စေသည့် (convenient) နည်းမျိုးဖြစ်သည်။

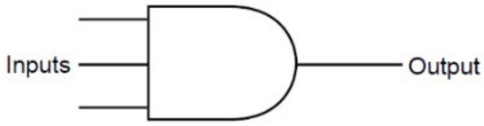
Logic 1 (+V)	Logic 0 (0V)	Example
Not operating	Operating	Limit switch
Not ringing	Ringing	Bell
Off	On	Light bulb
Silent	Blowing	Horn
Stopped	Running	Motor
Disengaged	Engaged	Clutch
Open	Closed	Valve

Table 3-2. Binary concept using negative logic.

Binary concept သည် physical quantities (Binary variables) များ အခြေအနေ ၂ မျိုး (two state) ကို 0 နှင့် 1 ဖြင့်ဖော်သည်။ အခြေအနေ ၂ မျိုးအနက် တစ်ချိန်၌ အခြေအနေတစ်မျိုးသာဖြစ်နိုင်သည်။ ၂ခုထက်ပိုသည့် Binary variable များပေါင်းစပ်ပြီး ရရှိသည့်ရလာဒ်ကို True 1 (သို့) False 0 အခြေအနေ တစ်မျိုး မျိုးဖြင့်ဖော်ပြပုံ ဆက်လက်လေ့လာမည်။ PLC များသည် logical statement များကို အခြေခံ၍ ဆုံးဖြတ်ချက်များချမှတ် (make decision) ကြသည်။ PLC များသည် Digital equipment များ ဖြစ်ကြသောကြောင့် AND ၊ OR နှင့် NOT စသည့် အခြေခံ (fundamental) Logic function ၃ မျိုးကို အခြေခံ၍ Operation များပြုလုပ်ကြသည်။ ထို function များပေါင်းစပ်ပြီး Binary Variable များကို Statement အဖြစ်တည်ဆောက်ကြသည်။

Function တိုင်းတွင် စည်းကမ်းချက် (rule) များရှိကြသည်။ ထိုစည်းကမ်းချက် (rule) များအရ ရလာဒ် (result) ကိုထုတ်ပေးသည်။ ဆုံးဖြတ်ပေးသည်။ ကိုယ်ပိုင်သင်္ကေတ (Symbol) များရှိကြသည်။ နားလည်မှုရှင်းလင်းစေရန်အတွက် ရလာဒ် (result of statement) ကို output ဟုသတ်မှတ် ခေါ်ဆိုသည်။ output ကို အင်္ဂလိပ်အက္ခရာ (Y) ဖြင့်သတ်မှတ်ဖော်ပြသည်။ Statement များ၏ အခြေအနေ(Condition) များကို input (A နှင့် B) များဟု သတ်မှတ်သည်။ input နှင့် output ၂ ခုစလုံးကို အခြေအနေ ၂ မျိုး (two state variable) ဖြင့်ဖော်ပြနိုင်သည်။

၃.၂.၁ "AND" Function



ပုံ 3-1. Symbol for the AND function.)

ပုံ 3-1 သည် AND gate တစ်ခု၏ သင်္ကေတ (Symbol) ဖြစ်သည်။ AND function ကိုပုံဖြင့် ဖော်ပြထားခြင်းဖြစ်သည်။ Input များအားလုံး True (1) ဖြစ်မှသာ AND output သည် True (1) ဖြစ်သည်။

$\textit{The AND output is True } (1) \textit{ only if all inputs are True } (1).$

DRAFT

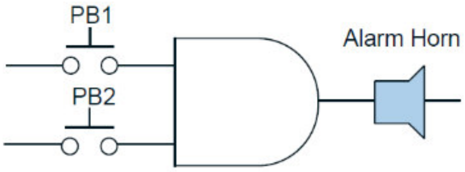
AND Truth Table		
Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

ပုံ 3-2 Two-input AND gate and its truth table.

ဥပမာ - ၃-၁

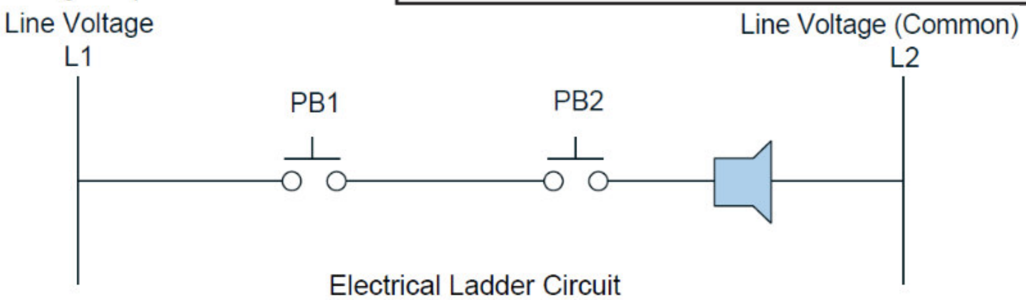
Input ၂ ခုဖြစ်သည့် Push Button 1 (BP1) နှင့် Push Button 2 (BP2) နှစ်ခုတို့ တစ်ပြိုင် နက်တည်း ဖိ (Press) မှသာလျှင် (ON ဖြစ်မှသာ) Alarm Horn မှအသံမြည်စေရန်အတွက် Logic gate ၊ truth Table နှင့် Electric Circuit ၊ Ladder Diagram (Circuit) များဖြင့် ဖော်ပြပေးပါ။

Solution



Logic Representation

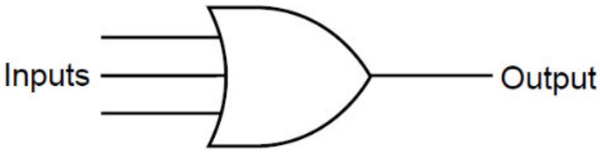
PB1	PB2	Alarm Horn
Not pushed (0)	Not pushed (0)	Silent (0)
Not pushed (0)	Pushed (1)	Silent (0)
Pushed (1)	Not pushed (0)	Silent (0)
Pushed (1)	Pushed (1)	Sounding (1)



Electrical Ladder Circuit

Electrical Circuit

၃.၂.၂ The "OR" Function



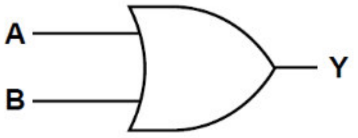
ပုံ 3-3. Symbol for the OR function.

ပုံ 3-3 သည် OR gate တစ်ခုကို သင်္ကေတ (Symbol) ဖြင့်ဖော်ပြထားသည်။ OR function ကို ပုံဖြင့် ဖော်ပြထားခြင်းဖြစ်သည်။

The OR output is True (1) ***if on or more inputs are True*** (1).

OR gate ၏ input များအားလုံးအနက်မှ အနည်းဆုံး input ၁ ခု True ဖြစ်လျှင် output သည် True (1) ဖြစ်သည်။

AND function (AND Gate) နှင့် OR function (OR Gate) တို့တွင် input ပေါင်းများစွာရှိနိုင်သော်လည်း output ၁ ခုသာရှိမည်။ ပုံ 3-4 တွင် OR function ၏ True Table ကို ဖော်ပြထားသည်။ input များအားလုံး၏ ဖြစ်နိုင်သည့် Combination ကိုအခြေခံ၍ output Y ကိုထုတ်ပေးထားသည်။



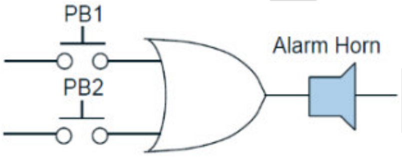
OR Truth Table		
Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

ပုံ 3-4. Two-input OR gate and its truth table.

ဥပမာ ၃-၂

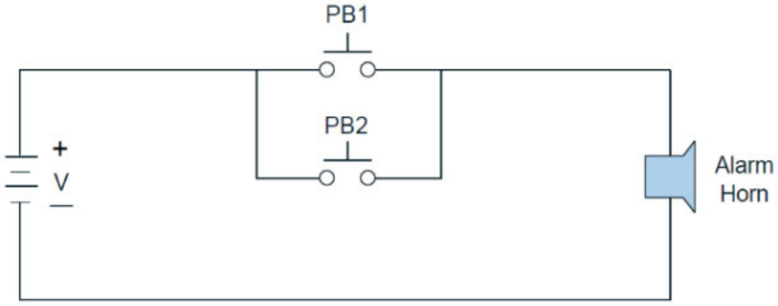
Input ၂ ခုဖြစ်သည် Push Button (BP1) နှင့် Push Button (BP2) တို့အနက် မည်သည့် Push Button တစ်ခုခု နှိပ်လျှင် (ON ဖြစ်လျှင်) Alarm Horn အသံထွက်စေမည့် Logic Gate ၊ true tableနှင့် Circuit တို့ရေးဆွဲဖော်ပြပါ။

Solution

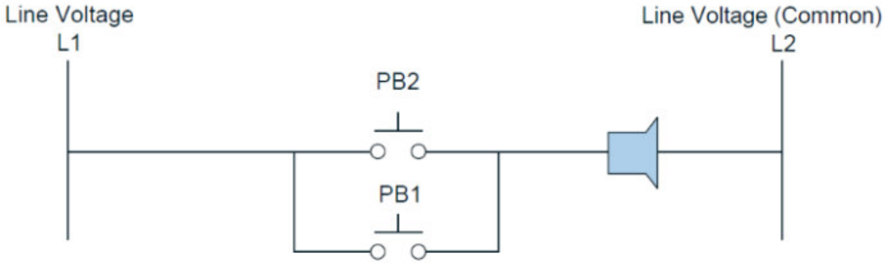


Logic Representation

PB1	PB2	Alarm Horn
Not pushed (0)	Not pushed (0)	Silent (0)
Not pushed (0)	Pushed (1)	Sounding (1)
Pushed (1)	Not pushed (0)	Sounding (1)
Pushed (1)	Pushed (1)	Sounding (1)



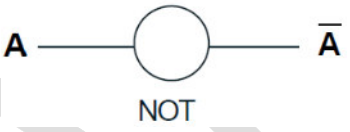
Electrical Circuit



Electrical Ladder Circuit

၃.၂.၃ The "NOT" function

ပုံ 3-5 NOT function ကို သင်္ကေတ (Symbol) ဖြင့်ဖော်ပြထားသည်။ NOT function ၏ input မှာ FALSE (0) ဖြစ်လျှင် output သည်TRUE (1) ဖြစ်သည်။ အပြန်အလှန်အားဖြင့် input မှာ True (1) ဖြစ်လျှင် output သည် FALSE (0) ဖြစ်သည်။ NOT Gate ၏ Operation မှာ input ၏ state ကိုပြောင်းပြန်(inverse) လုပ်ပေးရန် ဖြစ်သည်။ ထို့ကြောင့် NOT Gate ကို inverter ဟုလည်းခေါ်သည်။



NOT Truth Table	
Input	Output
A	Ā
0	1
1	0

ပုံ 3-5. Symbol for the NOT function.

ပုံ 3-6. NOT gate and its truth table.

တခြားသော AND function နှင့် OR function များနှင့်မတူသည့်အချက်မှာ NOT function တွင် input တစ်ခုတည်း သာပါရှိခြင်းဖြစ်သည်။ လက်ခံနိုင်ခြင်းဖြစ်သည်။ NOT function ကို တစ်ခုတည်း အသုံးပြုလေ့မရှိပေ။ AND gate (သို့) OR gate တို့ဖြင့်တွဲ၍ အသုံးပြုလေ့ရှိသည်။

ပုံ ၃-၆ တွင် NOT Operation နှင့် ယင်း၏ true table ကိုဖော်ပြထားသည်။ A အပေါ်တွင်ဘား (Bar) ကလေးတင်၍ Ā အဖြစ်ဖော်ပြခြင်းသည် NOT A ကို ရည်ညွှန်းသည်။

NOT function အလွယ်တကူသိမြင်နိုင်ရန်(Visualize လုပ်ရန်) ခဲယဉ်းသည်။ AND နှင် OR တို့၏ function ကို သိမြင်နိုင်ရန်(Visualize လုပ်ရန်) လွယ်သည်။ သိလွယ်မြင်လွယ်သည်။ သို့သော် NOT function သည် အလွန်ရိုးရှင်းပြီး၊ အလွန်အသုံးဝင်ကြသည်။

အစပိုင်းတွင် ရှင်းပြထားသည့်အချက် (၃) ချက်

(၁) 1 သို့ 0 မိမိနှစ်သက်သည့်အတိုင်းရွေးချယ်ပါ။

(၂) logic 1 သည် ပုံမှန်အားဖြင့်(များသောအားဖြင့်) TRUE ၊ HIGH ၊ ON ဖြစ်သည်။ logic 1 သည် Device တစ်မျိုးမျိုးကို Activate လုပ်ရန်ရည်ရွယ်သည်။ ဥပမာ- Output Y=1 ဖြစ်လျှင် မော်တာကို မောင်းရန်ဖြစ်သည်။

(၃) logic 0 သည် ပုံမှန်အားဖြင့် (များသောအားဖြင့်) FALSE ၊ LOW ၊ OFF ဖြစ်သည်။ logic 0 သည် device တစ်မျိုးမျိုးကို deactivate လုပ်ရန်ဖြစ်သည်။

ဥပမာ- Output Y =0 ဖြစ်လျှင် မော်တာကိုရပ်ရန်ဖြစ်သည်။ အကယ်၍ ထိုအခြေအနေကို ပြောင်းလိုလျှင် NOT function ကိုသုံးသည်။ ဥပမာ- logic 0 သည် Device တစ်မျိုးမျိုး Active လုပ်လိုလျှင် NOT function ကို အသုံးပြုသည်။ ဥပမာ- output Y=0 ဖြစ်လျှင် မော်တာကိုမောင်းရန်ဖြစ်သည်။ NOT function သည်အသုံးဝင်သည်။

(က) NOT ကိုအသုံးပြုသည်။ logic 0 (low condition) ဖြစ်လျှင် Device တစ်မျိုးမျိုးကို Active လုပ်ရန်

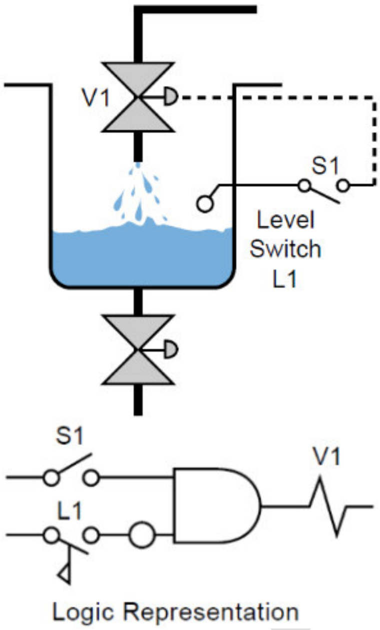
(ခ) Logic 1 (High Condition) ဖြစ်လျှင် Device တစ်မျိုးမျိုးကို deactivate လုပ်ရန်အတွက် NOT ကိုအသုံးပြုသည်။

အောက်တွင် NOT function ကိုသုံးထားသည့် ဥပမာ ၂ ခုကိုဖော်ပြထားသည်။

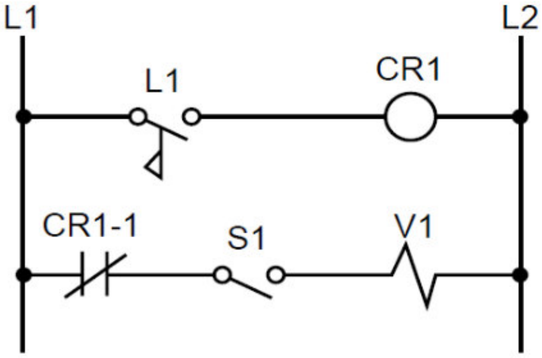
NOT function ကို AND နှင့် OR function များဖြင့်အသုံးပြုလေ့ရှိကြသော်လည်း ပထမဥပမာတွင် NOT function တစ်ခုတည်းကို သာသုံးထားသည်။

ဥပမာ ၃-၃

Selector switch (S1) သည် ON နေသည့်အခိုက် Level switch (L1) သည် သတ်မှတ်ထားသည့်ရေအမြင့် (level) သို့မရောက်ခင် (NOT ON) ဖြစ်လျှင် (တပြိုင်နက်ဖြစ်လျှင်) Solenoid value (V1) ကိုပွင့်စေရမည်၊ (ON) logic gate ၊ True Table နှင့် Circuit များကိုဖော်ပြပါ။



Logic Representation

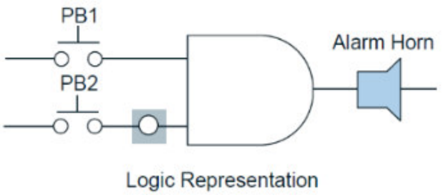


Electrical Ladder Circuit

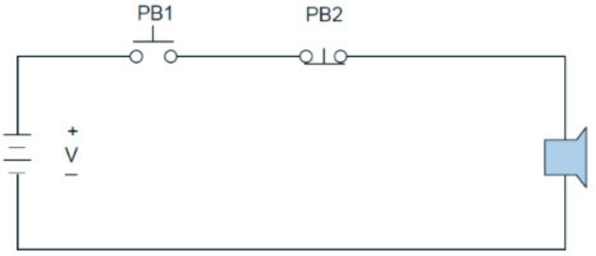
S1	L1	($\overline{L1}$)	V1
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

Truth Table

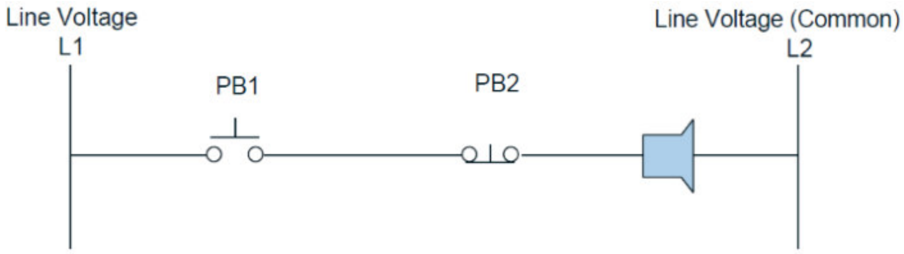
မှတ်ရန်။ ။ ဤဥပမာတွင် Field device element ကို Not function နှင့်တွဲသုံးရန်အတွက် Normally Closed (NC) အဖြစ်ထားလေ့ရှိသည်။ Switch (PB2)သည် normally open ဖြစ်မနေပေ။ (Not Normally Open) ဤဥပမာ၏ logical representation မှာ push botton switch တွင်သုံးမည့် NOT function ကို သင်္ကေတဖြင့်ဖော်ပြသည်။ ပြီးခဲ့သည့် ဥပမာ ၂ ခုတွင် NOT သင်္ကေတ (Symbol) ကို Gate ၏ input နေရာတွင်ထားရှိသည်။ AND gate ၏ output (အထွက်၌) NOT သင်္ကေတ (Symbol)ကိုထားလျှင် Negate (သို့) invert ဖြစ်သည်။ AND gate ၏ output ကို NOT function ထည့်ထားလျှင် NAND gate ဖြစ်သည်။ (တနည်း negated AND gate ဖြစ်သည်။)



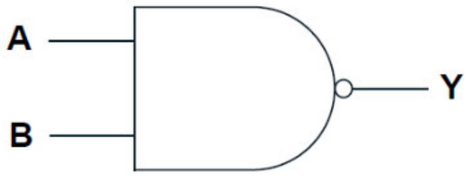
PB1	PB2	Alarm Horn
Not pushed (0)	Not pushed (0)	Silent (0)
Not pushed (0)	Pushed (1)	Silent (0)
Pushed (1)	Not pushed (0)	Sounding (1)
Pushed (1)	Pushed (1)	Silent (0)



Electrical Circuit



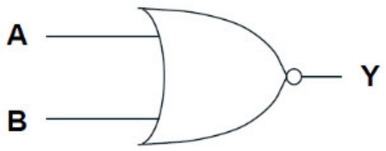
၃.၂.၄ NAND Gate



ပုံ 3-7. Two-input NAND gate and its truth table.

NAND Truth Table		
Inputs		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

ပုံ 3-7 တွင် NAND gate ၏ logic symbol နှင့် truth table ကိုဖော်ပြထားသည်။ OR gate ၏ output ၌ NOT သင်္ကေတ (symbol)ထည့်လျှင်လည်း အလုပ်လုပ်ပုံမှာတူညီသည်။ OR gate ၏ normal output ကို function ထည့်ထားသောကြောင့် NOR gate ဖြစ်သည်။ ပုံ 3-8 တွင် NOR gate ၏ logic symbol နှင့် truth table ကိုဖော်ပြထားသည်။



ပုံ 3-8. Two-input NOR gate and its truth table.

NOR Truth Table		
Inputs		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

၃.၃ Principle of Boolean and Logic

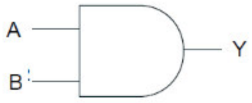
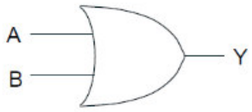
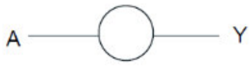
ဤစာအုပ်တွင် Boolean Algebra အကြောင်းအသေးစိတ်က မဖော်ပြထားပါ။ သို့သော် Control Program များနှင့် conventional Ladder Diagram များတွင် Boolean statement များကို အသုံးပြု တတ်စေရန်အတွက် (complex logical statement များကိုတည်ဆောက်တတ်စေရန်အတွက်) Boolean technique များကိုနားလည်ရန်လို သည်။

၁၈၄၉ ခုနှစ်တွင် အင်္ဂလိပ်လူမျိုး George Boole က Boolean Algebra ကို စတင်တီထွင်ခဲ့ သည်။ logical reasoning နှင့် Ancient philosophy များတွင် အသုံးပြုရန်အတွက်ဖြစ်သည်။ ရှုပ်ထွေးခက်ခဲသည့် "logical statement"များကို လွယ်ကူရှင်းလင်းသည့်ပုံစံမျိုးဖြင့်ပြန်လည်ရေးသားနိုင်သည်။

၁၉၆၀ ပြည့်နှစ်များနောက်ပိုင်းတွင် Digital logic များပေါ်ထွန်းလာပြီးနောက် Boolean Algebra သည် Digital logic statement များကို ခွဲခြားစိတ်ဖြာခြင်း (analyze)၊ ဖော်ပြခြင်း (express) ပြု လုပ်ပြနိုင်ခဲ့သည်။ Digital logic statement များသည် TRUE နှင့်FALSE ၊ 0 နှင့် 1 စသည့် Binary state များ ဖြင့်သာ ဖော်ပြထားခြင်းဖြစ်သည်။

Digital logic နှင့် Boolean logic တို့သည် Binary state ကို အခြေခံထားသည့် logic များ ဖြစ်သည်။ ထို့ကြောင့် တစ်ခါတရံ logic gate များကို Boolean gate များဟုခေါ်ဆိုလေ့ရှိသည်။ logic gate (သို့) Boolean gate များပါဝင်သည် network ကို Boolean network ဟုခေါ်ဆိုသည်။ PLC language ကို ပင် Boolean language ဟုခေါ်ဆိုသည်။

ပုံ 3-9 အခြေခံ Digital logic function များဖြစ်သည့် AND ၊ OR ၊ NOTတို့ကို အခြေခံ Boolean operator များအဖြစ်ပြန်လည်ဖော်ပြထားသည်။ operator များ၌ စာလုံးကြီး (capital letter) ကို အ သုံးပြုထားသည်။ AND operation ကို အမြှောက်လက္ခဏာ (multiplication sign) (•) ဖြင့်ဖော်ပြထားသည်။ OR function ကို အပေါင်းလက္ခဏာ (addition sign) (+) ဖြင့်ဖော်ပြထားသည်။ NOT operation ကိုစာလုံး (Letter) ၏ အပေါ်တွင် ဘား (Bar) ထည့်၍ ဖော်ပြထားသည်။

Logical Symbol	Logical Statement	Boolean Equation
	Y is 1 if A AND B are 1	$Y = A \cdot B$ or $Y = AB$
	Y is 1 if A OR B is 1	$Y = A + B$
	Y is 1 if A is 0 Y is 0 if A is 1	$Y = \overline{A}$

ပုံ 3-9. Boolean algebra as related to the AND, OR, and NOT functions.

DRAFT

၃.၃.၁ AND operation

ပုံ 3-9 AND gate တစ်ခုတွင် input signal ဖြစ်သော A နှင့် B ပါရှိပြီး output signalသည် Y ဖြစ်သည်။ output ကို logical statement ဖြင့်အောက်ပါအတိုင်းဖော်ပြနိုင်သည်။

A နှင့် B သည် 1 ဖြစ်လျှင် Y သည် 1 ဖြစ်သည်။
Y is 1 if A AND B are 1 .
Y= A•B

အောက်ပါအတိုင်းအသံထွက်ဖတ်နိုင်သည်။ Y equals A ANDed with B, ရေးသားဖော်ပြသည့်အခါတွင် Boolean သင်္ကေတဖြစ်သည့် (•) [AND operation အတွက်] ကိုချန်ထားပြီး Y=AB ဟုရေးသားနိုင်သည်။

၃.၃.၂ OR operation

OR gate တစ်ခုတွင် input signal များ A နှင့် B ဖြစ်ပြီး output signal မှာ Y ဖြစ်သည်။
“OR” output ကို logical statement ဖြင့်အောက်ပါအတိုင်းရေးသားနိုင်သည်။

Y= A+B
“Y equals A ORed with B.” ဟုဖတ်လေ့ရှိသည်။

၃.၃.၃. Not Function

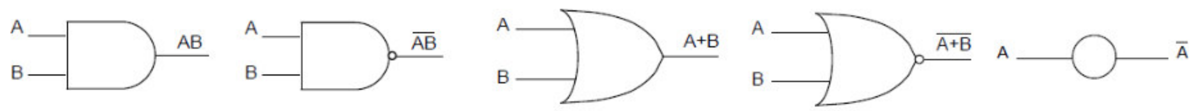
Y သည် A ၏ ပြောင်းပြန်ဖြစ်သည်။ Y is reverse of A.
Boolean Expression ဖြင့် အောက်ပါအတိုင်းရေးနိုင်သည်။

$Y = \bar{A}$
Y equals to NOT A
B.E Y2 A+B တွင် ဟုဖတ်လေ့ရှိသည်။

ဇယား 3-3 တွင် Boolean Operation ANDing(AND လုပ်ခြင်း)၊ ORing (OR လုပ်ခြင်း) နှင့် inversion (NOT လုပ်ခြင်း) တို့ကိုဖော်ပြထားသည်။

အလိုရှိသည့် logic combination ရရန်အတွက် function များ မည်ကဲ့သို့ပေါင်းစပ်ယူရမည်ကို ဖော်ပြထားသည်။

1. Basic Gates. Basic logic gates implement simple logic functions. Each logic function is expressed in terms of a truth table and its Boolean expression.



DRAFT

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

AND

A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

NAND

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

OR

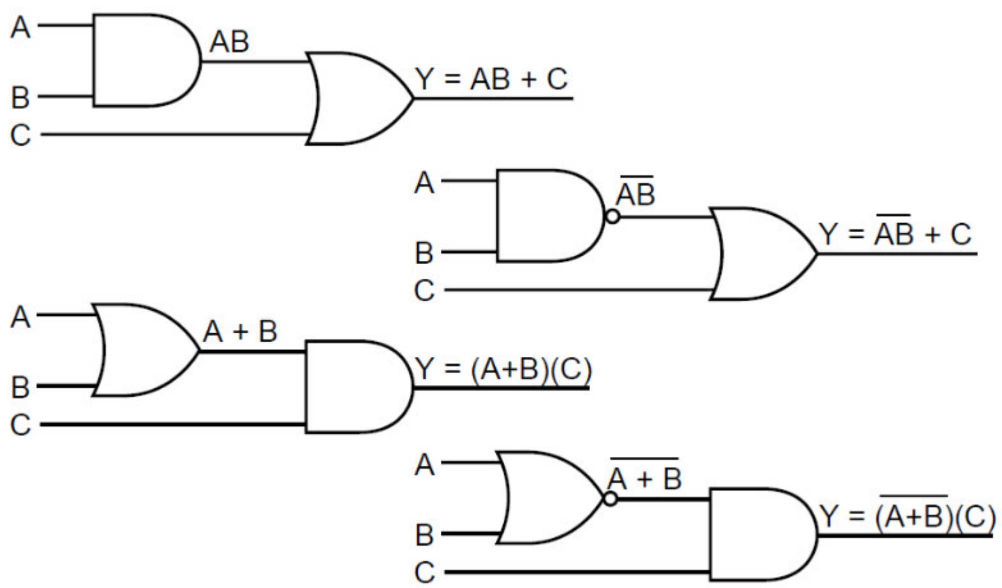
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

NOR

A	$\overline{A}$
0	1
1	0

NOT

2. Combined Gates. Any combination of control functions can be expressed in Boolean terms using three simple operators: (•), (+), and (‾).



3. Boolean Algebra Rules. Control logic functions can vary from simple to very complex combinations of input variables. However simple or complex the functions may be, they satisfy the following rules. These rules are a result of a simple combination of basic truth tables and may be used to simplify logic circuits.

Commutative Laws

$A+B=B+A$

$AB=BA$

Associative Laws

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

$A(BC)=(AB)C$

De Morgan's Laws

$\overline{(A+B)}=\overline{A}\,\overline{B}$

$\overline{(AB)}=\overline{A}+\overline{B}$

$\overline{\overline{A}}=A,\ \overline{1}=0,\ \overline{0}=1$

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

$AB+AC+B\overline{C}=AC+B\overline{C}$

Distributive Laws

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

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

Law of Absorption

$A(A+B)=A+AB=A$

Table 3-3 Logic operations using Boolean algebra.

၃.၄ Order of Operation နှင့် Group Sign

AND ၊ OR ၊ NOT စသည့် Boolean operation ပြုလုပ်သည့်အစီအစဉ်(Order) သည် အလွန်အရေးကြီးသည်။ operation လုပ်သည့် အစီအစဉ်(Order)မတူလျှင် expression မှ ရရှိသည့် ရလာဒ် Logic value သည်လည်းတူညီလိမ့်မည်မဟုတ်ပေ။

ဥပမာအနေဖြင့် input signal A ၊ B နှင့် C တို့ပါဝင်သည့် expression Y=A+B • C ကို စဉ်းစားကြစို့။
ပထမဦးစွာ OR operation ပြုလုပ်ပြီးမှ AND Operation လုပ်ရန် (A+B) • C ဟုရေးထားသားသည်။
AND Operation ပြုလုပ်ပြီးမှ OR operation ပြုလုပ်ရန် A+ (B • C) ဟုရေးသားထားသည်။
(A+B) • C နှင့် A+ (B • C) တို့၏ ရလာဒ် (result)တို့သည် တူညီလိမ့်မည်မဟုတ်ပေ။ Boolean expression များ၌ grouping လက္ခဏာ (sign) များဖြစ်သည့် လက်သည်းကွင်း-parentheses () များ ၊ Bracket [] များ ၊ Brace များ သို့မဟုတ် Vinculum ဖြင့် ရေးသားဖော်ပြထားခြင်းမရှိလျှင် ပထမဦးစွာ NOT operation (inversion) လုပ်ပါ။ ဒုတိယ AND operation လုပ်ပါ။ OR operation ကိုနောက်ဆုံးမှ ပြုလုပ်ပါ။

ဤစည်းကမ်းအရ အထက်တွင်ဖော်ပြခဲ့သော Boolean expression Y=A+B • C တွင် grouping sing မပါဝင်သောကြောင့် B နှင့် C ကို AND operation လုပ်ပြီးမှ ရသည့် ရလာဒ် နှင့် B ကို OR operation ပြုလုပ်ရန်ဖြစ်သည်။

Y=A+B • C ၏ operation လုပ်ရမည့် အစီအစဉ်ကို သိပြီးနောက် expression ကို ဤသို့ Y=A+BC ပို၍ ရှင်းလင်းစွာရေးသားနိုင်သည်။Boolean expression များ ၌ AND operator ကိုမြုတ်၍ရေးသားလေ့ရှိသည်။
Boolean expression Y=A+BC တွင် B နှင့် C ကို အရင်ဆုံး AND operation ပြုလုပ်လိုပါက လက်သည်းကွင်း-parentheses () အတွင်း၌ ထည့်ရန်မလို ပေ။ သို့သော် A နှင့် B ကိုအရင်ဦးစွာ OR operation ပြုလုပ်ရမည်ဆိုလျှင် ( ) ထဲတွင် ထည့်ရေးရန်လိုသည်။
လက်သည်းကွင်း-parentheses (), Bracket [ ], Brace { } စသည့် grouping sign များကိုသုံး၍ operation order များမှန် အောင်ရေးရန်အရေးကြီးသည်။ Grouping sign များကိုသုံး၍ ရေးပြထားသည်။

$$\boldsymbol{Y1} = \boldsymbol{Y2} + \boldsymbol{Y5}[\,\boldsymbol{X1}(\boldsymbol{X2} + \boldsymbol{X3})] \;+ \{\boldsymbol{Y3}[\boldsymbol{Y4}(\boldsymbol{X5} + \boldsymbol{X6})]\}$$

Hardwired logic များတွင် timing ၊ sequencing နှင့် control စသည့် control function များကိုလိုက်၍ Device ချိတ်ဆက်(connect) ပုံကွဲပြားကြသည်။ hardwired နှင့်မတူသည့်အချက်မှာ PLC များတွင် logic function များကို အလွယ်တကူပြောင်းလဲနိုင်ခြင်းဖြစ်သည် ။ hardwired logic များသည် ဝါကြိုးများဖြင့် အသေဆက်သွယ်ထားသောကြောင့် ပြောင်းလိုလျှင် ဝါယာကြိုးများကို ဖြုတ်၍ ပြန်တပ်ယူရသည်။

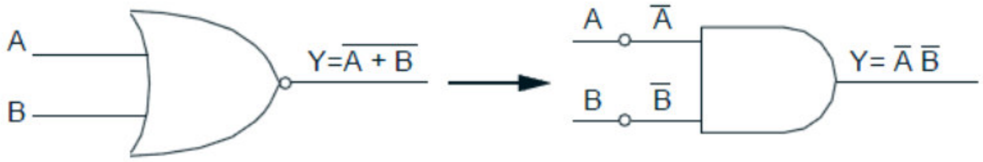
PLC များ၏ အထူးကောင်းမွန်ချက်မှ အလွယ်တကူပြုပြင်ပြောင်းလဲ နိုင်ခြင်းဖြစ်သည်။ပုံ ၃-၁၀a သည် Hardwired Relay Logic Circuit ဖြစ်သည်။ ပုံ ၃-၁၀b သည် ထို Hardwired Relay Logic Circuit ကို PLC Ladder Diagram ဖြစ်သို့ပြောင်းထားပုံဖြစ်သည် ။ ပုံ ၃-၁၀a နှင့်ပုံ ၃-၁၁ b တို့သည် HRLC နှင့် LD တို့၏ တူညီချက်များကိုဖော် ပြထားသည်။ Hardwired Relay Logic Circuit အစား software instruction အဖြစ်ပြောင်းခြင်းဖြစ်သည်။ ထို့ကြောင့် Ladder Diagram ကို Software ဟုခေါ်ဆိုသည်။

5. Application of De Morgan's Laws. De Morgan's Laws are frequently used to simplify inverted logic expressions or to simply convert an expression into a usable form.

According to De Morgan's Laws:

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

$$\text{and } \overline{A + B} = \overline{A}\,\overline{B}$$



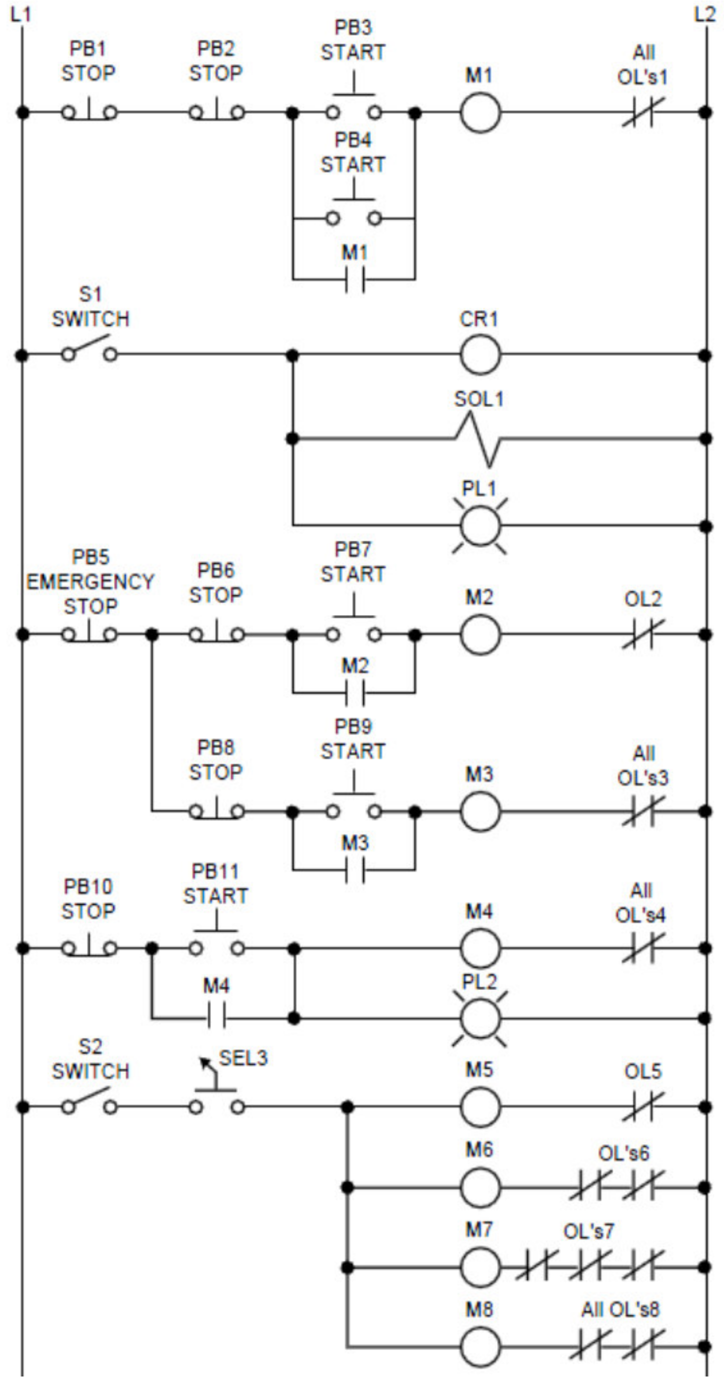
PLC တွင်အသုံးပြုမည့် Ladder Diagram logic များကို အခြေခံ (Base) logic function (AND, OR နှင့် NOT) သုံးမျိုးကို တစ်ခုချင်းစီသော်လည်းကောင်း၊ ပေါင်းစပ်၍သော်လည်းကောင်း တည်ဆောက်လေ့ရှိသည်။ Ladder Diagram logic များသည် Value များဖွင့်ခြင်း၊ ပိတ်ခြင်းပြုလုပ်ရန်၊ မော်တာများမောင်းခြင်း ရပ်ခြင်းပြုလုပ်ရန် စသည့် instruction များ ဖြစ်သည်။ PLC ကိုခိုင်းစေရန် Command များ ၊ instruction များ စနစ်တကျ ရေးသားထားခြင်းကို program သို့မဟုတ် Control program ဟုခေါ်သည်။

ထို program များကို ရေးသားသည့် ပုံစံအမျိုးမျိုး၊ style အမျိုးမျိုးကို language ဟုခေါ်ဆိုသည်။ အသုံးများသည့် language မှာ Ladder Diagram နှင့် Boolean ဖြစ်သည်။

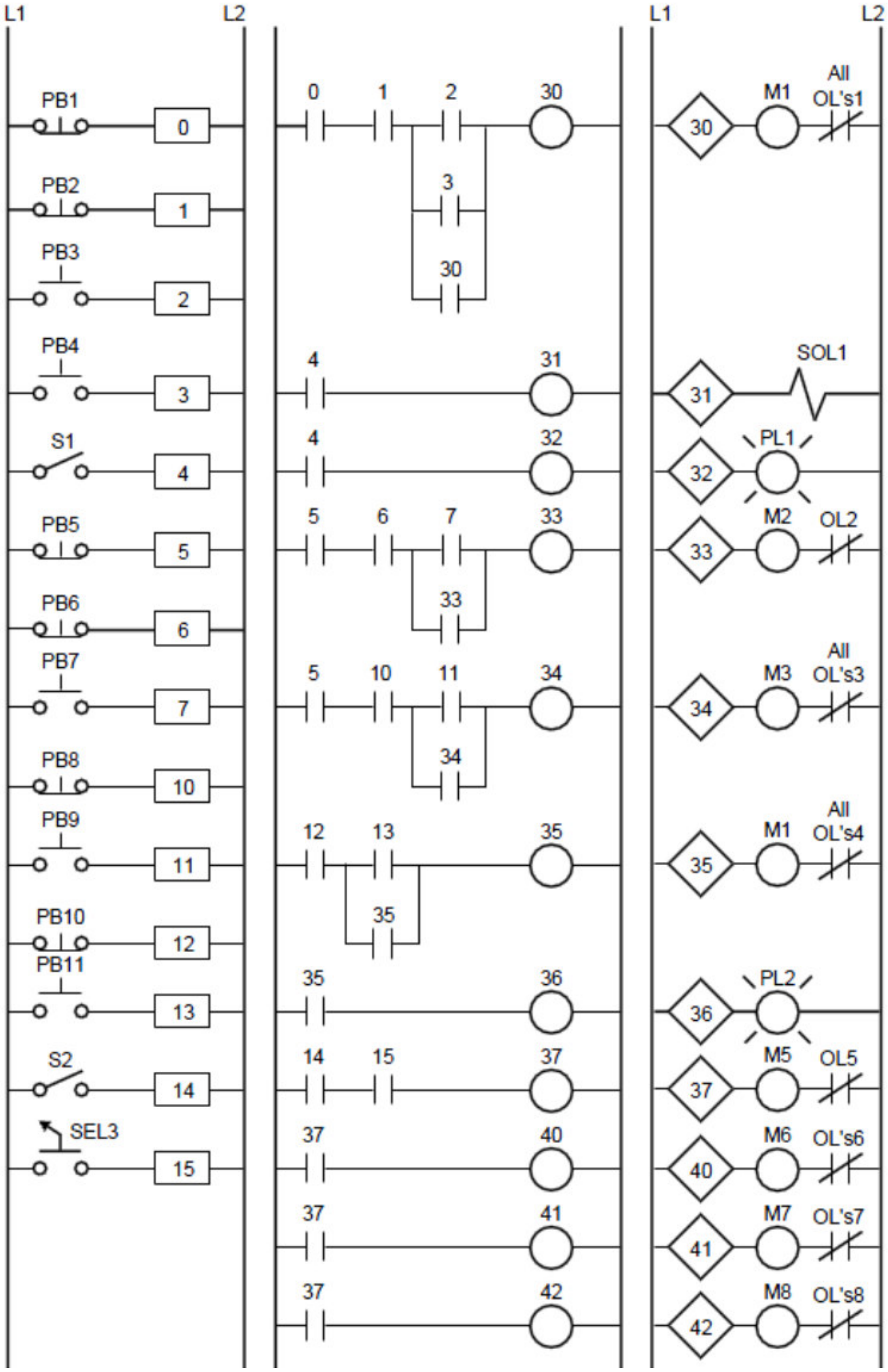
အခန်း ၉ (chapter 9) Programming များအကြောင်းကိုအသေးစိတ်ရှင်းပြထားသည်။ ရှေးယခင်ကတည်းက သုံးစွဲခဲ့ကြသည့် Control language တစ်မျိုးမှာ Ladder Diagram ဖြစ်သည်။ Ladder Diagram များကို contact symbology ဟုခေါ်သည်။ Ladder Diagram ၌ ပါရှိသည့် instruction များသည် Normally Open နှင့် Normally Close စသည့် relay equipment contact symbols များဖြစ်သည်။

Contact symbology သည် control logic များကိုဖော်ပြရန်အတွက် အရိုးရှင်းဆုံးနည်းတစ်ခု ဖြစ်သည်။ relay control schematic များကို သင်္ကေတ(symbol)များဖြင့်ဖော်ပြခြင်းဖြစ်သည်။

၃.၄ PLC circuit နှင့် logic contact သင်္ကေတများ



ပုံ 3-10a Hardwired relay logic circuit.

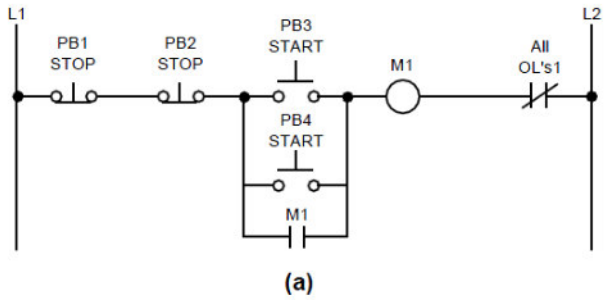


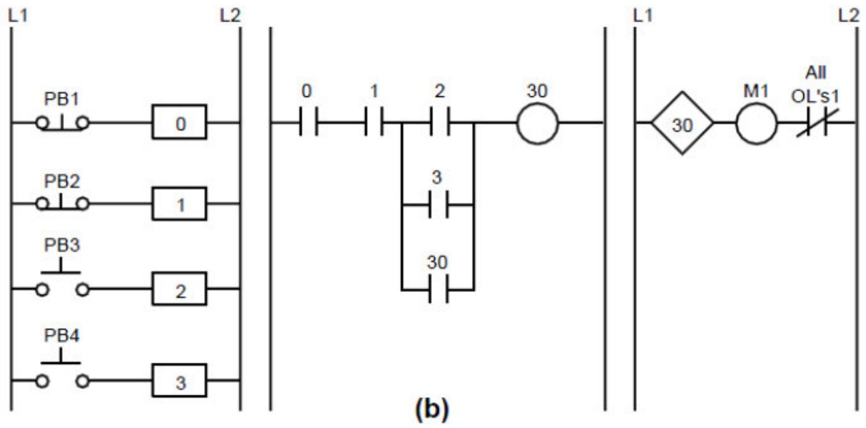
ပုံ 3-10b PLC ladder diagram implementation of Figure 3-10a.

အကယ်၍ controller language သည် Ladder Diagram ဖြစ်လျှင် ပထမအဆင့်အဖြစ် ရှိပြီးသား relay logic မှ programming (CS) အဖြစ်သို့ ပြောင်းရန်လိုသည်။ အကယ်၍ controller language သည် Boolean mnemonics, ဖြစ်လျှင် contact symbology ပြောင်းရန်မလိုပေ။ သို့သော် ပိုမိုရှင်းလင်းစွာ နားလည်နိုင်ရန်နှင့် မှတ်တမ်းများအဖြစ် စနစ်တကျ သိမ်းဆည်းရန်အတွက် Relay Logic နှင့် PLC logic သို့ပြောင်းလေ့ရှိသည်။ ဇယား ၃-၆ a တွင် Hardwired logic (Relay Logic) မှ (PLC logic) သို့ ပြောင်းထားပုံကို နမူနာအဖြစ် ဖော်ပြထားသည်။

ပုံ၃-၁၀ သည်လုံးဝပြည့်စုံသည် Ladder circuit ဖြစ်သည်။ circuitတစ်ခုချင်းစီအဖြစ်လည်းခွဲထုတ်ယူနိုင်သည်။ circuit တစ်ခုချင်းစီ၌ output တစ်ခုစီပါရှိသည်။ ထို circuit တစ်ခုချင်းစီကို rung (သို့မဟုတ် network) သတ်မှတ်လေ့ရှိသည်။ ထို့ကြောင့် rung သည် contact symbology ဖြစ်သည်။ (PLC ဖြင့် output တစ်မျိုးမျိုးကို control လုပ်သည့်) အချို့သော controller များ၌ rung မှ တစ်ခုထက်ပိုများသော multiple output များထုတ်ပေးနိုင်သည်။ သို့သော် အစဉ်အလာအားဖြင့် rung တစ်ခုလျှင် output ၁ ခုကိုသာထုတ်ပေးလေ့ရှိသည်။ ပုံ ၃-၁၁ a သည် hardwired circuit ဖြစ်သည့် ပုံ၃-၁၀ မှ top rung ကိုဖော်ပြထားသည်။ ပုံ ၃-၁၁b သည် ပြောင်းထားသည့် PLC circuit မှ top rung ကိုဖော်ပြထားသည်။

PLC diagram တွင် PLC ၏ I/O interface တွင်တပ်ဆင်ထားသည့် field input output device များပါဝင်သည် ပြည့်စုံသည့် (complete) PLC Ladder Diagram တစ်ခုတွင် rung ပေါင်းများစွာပါဝင်သည်။ rung တစ်ခုတိုင်းသည် output interface တစ်ခုကို control လုပ်သည်။ output interface သည် output device တစ်ခုနှင့်ချိတ်ဆက်(connect)ထားသည်။ output device သည် equipment တစ်မျိုးမျိုးဖြစ်ပြီး PLC မှ ပေးပို့သည့် information အတိုင်း (သို့)ထုတ်ပေး output signal အတိုင်း(သို့) ခိုင်းသည့် instruction အတိုင်း ပြုလုပ်ဆောင်ရွက်ပေးသည်။





ပုံ 3-11(a) **(a)** Top rung of the hardwired circuit from Figure 3-10 and **(b)** its equivalent PLC circuit.

Input များနှင့် သက်ဆိုင်သည့် symbol များကို အလိုရှိသည့် (Desired) Logic များရရန်အ တွက် အတန်းလိုက် (series)သော်လည်းကောင်း ၊ အပြိုင်ပုံစံမျိုး(parallel) သော်လည်းကောင်း၊ သို့မဟုတ် နှစ်မျိုးလုံးပေါင်း၍ သော်လည်းကောင်းချိတ်ဆက်(connect)လေ့ရှိကြသည်။ ထို input symbol များသည် PLC ၏ input interface ၌လာရောက်ချိတ်ဆက်(connect) ထားသည့် input device များကို ကိုယ်စားပြုသည်။ input device များသည် PLC ထံသို့ field data များကို ပေးပို့သည်။

Programmed rung concept သဘောတရားသည် hardwired relay Ladder rung များ ၏ အလုပ်လုပ်ပုံကိုအတုယူကာ လက်ခံသုံးစွဲခြင်းဖြစ်သည်။ Hardwired Relay Logic Circuit များ၌ အမျိုးမျိုးသော output များကို control လုပ်ရန်အတွက် input device များကိုချိတ်ဆက် (connect) ထားခြင်း ဖြစ်သည်။ Activate ဖြစ်သည့်အခါ ထို input device များသည် circuit အတွင်းသို့ လျှပ်စစ် current ကိုဆက်စီးခွင့်ပြုလိုက်သည်။ သို့မဟုတ် စီးနေသည့် current ကိုဖြတ်ချ(Break)လိုက်သည်။ထို့ကြောင့် output device သည် ပွင့်ခြင်း (ON) သို့မဟုတ် ပိတ်ခြင်း (OFF) ဖြစ်ပေါ်လာသည်။

Ladder rung ၌ပါရှိသော input symbol သည် ချိတ်ဆက် (connect)ထားသည့် input device မှထုတ်ပေး သည့် input signal ကို ကိုယ်စားပြုသည်။ controller မှာထုတ်လိုက်သည့် output signal သည်ချိတ်ဆက် ထားသည့် (connect) output device ထံသို့ ရောက်သွားသည်။

Input Devices	Output Devices
Push button	Pilot light
Selector switch	Solenoid valve
Limit switch	Horn
Proximity switch	Control relay
Timer contact	Timer

Table 3-4. ON/OFF input and output devices.

၃.၅ PLC ၌အသုံးပြုသည့် Address များ

Rung တစ်ခု၌ ရှိသည့် Symbol တိုင်းအတွက် reference number များရှိရန်လိုအပ်သည်။ reference number ဆိုသည်မှာ memory ပေါ်၌ရှိသည့် address ဖြစ်သည်။ memory ပေါ်ရှိနေရာတစ်ခု၏ address ဖြစ် သည်။ လက်ရှိတန်ဘိုး (current status) ကို memory ပေါ်ရှိနေရာတစ်ခု၌ reference number တစ်ခုဖြင့် သိမ်းဆည်းထားသည်။ ထိုနေရာကို address ဖြင့်သတ်မှတ်ဖော်ပြသည်။ Field signal တစ်ခုသည် input (သို့) output interface တစ်ခုခုနှင့် ချိတ်ဆက်(Connect)လိုက်လျှင် address သည် signal wire တပ်ဆင်ထားသည့် terminal နှင့်သက်ဆိုင်သည်။ address သည် ထို input/output terminal ဖြစ်သည်။ input (သို့) output တစ်ခု၏ address ကို control logic program ၌ ကြိုက်သလောက် အကြိမ်ပေါင်းများစွာသုံးနိုင်သည်။ control logic က အလိုရှိသလောက်သုံးနိုင်သည်။ ထိုသို့သုံးနိုင်ခြင်းသည် PLC ၏ ကောင်းမွန်သည့် feature တစ်ခုဖြစ်သည်။

အားသာချက်

Relay type hardwire များ၌ ထိုကဲ့သို့ additional contact တစ်ခုလိုအပ်ပါက additional Hardwire တစ်ခု ထပ်ထည့်ရန်လိုသည်။ PLC ၌အခမဲ့ လိုသလောက်ရနိုင်သည်။ Section 5-4 နှင့် 6-2 တို့တွင် I/O များနှင့် ပတ်သက်သည့် PLC memory များအကြောင်းဖော်ပြထားသည်။ ပုံ 3-12 တွင်ရိုးရှင်းသည့် electional Ladder circuit နှင့် ယင်း circuit ကို PLC logic အဖြစ်ပြောင်းထားပုံကိုဖော်ပြထားသည်။

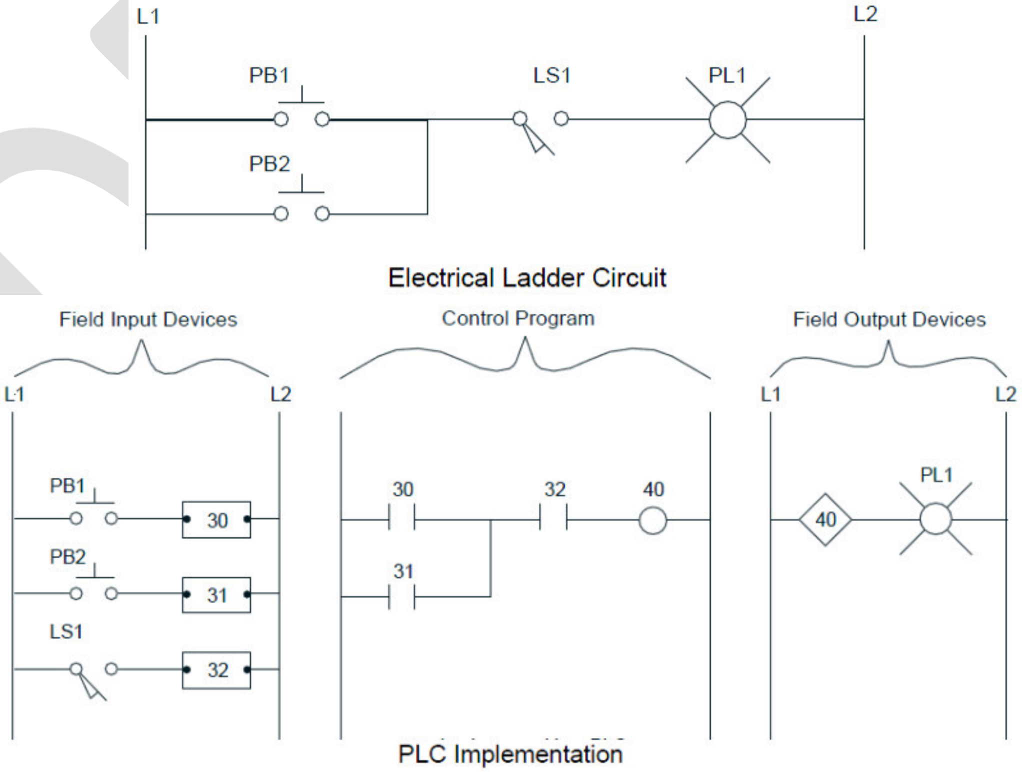
"real" field device ဟုခေါ်သည့် push button (PB1)၊ push button (PB2)၊ Limit switch (LS1) နှင့် pilot light (PL1) တို့သည် PLC ၏ input နှင့် output module တို့၌ ချိတ်ဆက် (connect) တပ်ဆင်ထားသည်။ ပုံ 3-13 ကိုကြည့်ရန်။

ထို real field device များအားလုံးအတွက် reference number ကိုယ်စီရှိကြသည်။ controller အများစု သည် reference number အဖြစ်Octal numbering (Base 8) နှင့် Decimal numbering (Base 10) တို့ကို numeric address များအဖြစ် အသုံးပြုထားသည်။

Ladder circuit များ၌ contact များအားလုံး close ဖြစ်သည့်အခါတွင် complete electrical path ဖြစ်ပေါ်က ထိုcircuit (line) တွင်ရှိသည့် output ကို energize ဖြစ်စေသည်။ output (pilot light PL1) ကို energize ဖြစ်စေသည်။ "ON" စေသည်။ ဥပမာ- Pilot light PL1 "ON" စေရန် အောက်ပါ condition ၂ ခုပြည့်စုံရမည်။

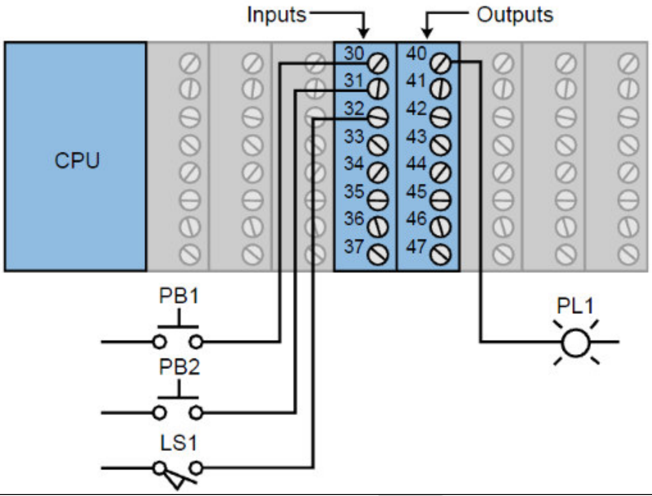
(၁) PB1 ကို ဖိထားရမည်။
(၂) LS1 သည် "CLOSE" ဖြစ်နေရမည်။ သို့မဟုတ်
(၁) PB2 ကိုဖိထားရမည်။
(၂) LS1 သည် "CLOSE" ဖြစ်နေရမည်။

အထက်ပါအခြေအနေ ၂ မျိုးအနက် ၁ မျိုးမျိုးသည် complete circuit ကို ဖြစ်စေသည်။ complete electrical path ကိုဖြစ်စေကာ လိုအပ်သည့် လျှပ်စစ်စွမ်းအားကို စီးဆင်းစေပြီး Pilot light ကိုလင်း စေသည်။



ပုံ 3-12 . Electrical ladder circuit and its equivalent PLC implementation.

DRAFT



ပုံ 3-13. Field devices from Figure 3-12 connected to I/O module.

ထို electrical ladder circuit တွင်သုံးထားသည့် logic ကိုပင် PLC circuit တွင်အသုံးပြုနိုင်သည်။ PLC control program တွင် လျှပ်စစ်စွမ်းအား (power) သည် address 30 (PB1) နှင့် address 32 (LS1) တို့မှ ၎င်း၊ address 31 (PB2) နှင့် address 32 (LS1) တို့မှ၎င်း စီးဆင်းကာ output 40 (PL1) ကို "ON" စေသည်။ output 40 သည် light PL1 ဖြစ်ပြီး Address 40 ရှိ interface terminal နှင့် ချိတ်ဆက်(connect) လုပ်ထားသည်။ Address 30 ၊ 31 (သို့) 32 တို့ကို power ပေးရန်အတွက် ယင်းတို့သည် interface terminal address 30 ၊ 31 နှင့် 32 နှင့် ချိတ်ဆက် (connect) ထားရမည်။

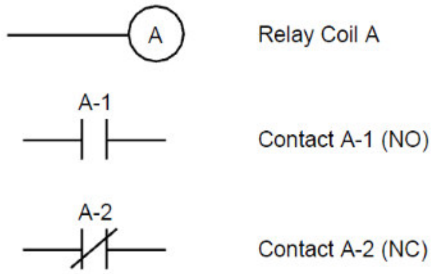
[ပြန်ရေးရန်]

၃.၆ Contact symbols used in PLC

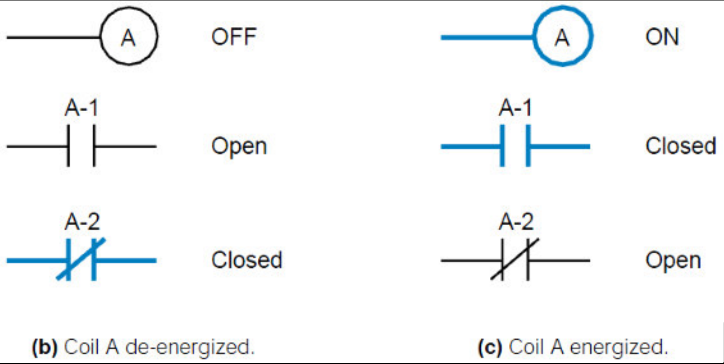
PLC တွင် အသုံးပြုသည့် Contact Symbol များ PLC ၌ ရှိသော Contact များနှင့် electromechanical relay များ၏ contact များ၏ အလုပ်လုပ်(operation) မှာ ခပ်ဆင်ဆင်တူညီကြသည်။ ဥပမာပုံ 3-14 a တွင် ပြထားသည့် relay A တွင် Contact ၂ ခုပါဝင်သည်။ A-1 သည် Normally Open ဖြစ်၍ A-2 သည် Normally Close ဖြစ်သည်။ အကယ်၍ relay အတွင်း၌ရှိသော Coil A သည် energize ဖြစ်မနေလျှင် (OFF ဖြစ်နေလျှင်) Contact A-1 သည် Normally Open ဖြစ်မြဲအတိုင်း Normally Open ဖြစ်နေလိမ့်မည်။ Contact A-2 သည် Normally Close ဖြစ်နေလိမ့်မည်။

အကယ်၍ Coil A သည် energise ဖြစ်လျှင် (တနည်း ON ဖြစ်လျှင်) Contact A-1 သည် N.O မှ close အဖြစ်သို့ ပြောင်းသွားလိမ့်မည်။ Contact A-2 သည် N.C မှ open အဖြစ်သို့ပြောင်းသွားလိမ့်မည်။ အပြာရောင်လိုင်းအထူများသည် coil နှင့် contact "ON" နှင့် ပြောင်းသွားသည့် အခြေအနေကို ဖော်ပြထားသည်။

DRAFT

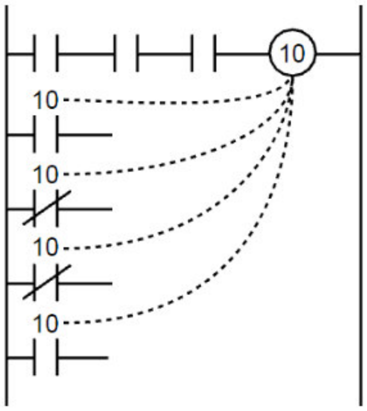


(a) Standard configuration for relay coil A with normally open contact A-1 and normally closed contact A-2.



ပုံ 3-14. Relay and PLC contact symbols showing a relay coil and normally open and normally closed contacts.

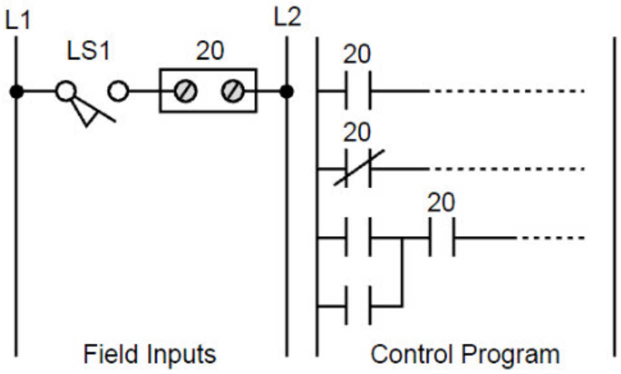
Contact close ဖြစ်လျှင် လျှပ်စစ်ကို စီးစေသည်။ Power flow ဖြစ်သည်။ (သို့) Continuity ဖြစ်ပေါ်စေသည်။ လျှပ်စစ်ပတ်လမ်း (Circuit) တွင် PLC ရှိ Coil များနှင့် သက်ဆိုင်သည့် Contact များအားလုံးအတွက် unique reference address များရှိကြသည်။ RA များသည် တစ်ခုနှင့်တစ်ခု မတူညီကြပေ။ unique ဖြစ်ကြသည်။ ပုံ 3-15 တွင်ဖော်ပြထားသည့် Coil နံပါတ် 10 တွင် Coil နံပါတ်ကိုလိုက်၍ Normally Open နှင့် Normally Close ရှိကြသည်။ Coil 10 အတွက် Normally Open နံပါတ် ၁၀ နှင့် Normally Close နံပါတ် ၁၀ တို့ဖြစ်သည်။ သတိပြုရန်အချက်မှာ PLC ၌ Normally Open နှင့် Normally Close Contact များ အလိုရှိသလောက် ရရှိနိုင်သည်။ သို့သော် electromechanical relay များ၌ Contact များ လိုသလောက် မရနိုင်ပေ။ အကန့်အသတ်ဖြင့်သာရနိုင် သည်။



ပုံ 3-15. Multiple contacts from a PLC output coil.

DRAFT

PLC များ၌ input device reference number များကို ထပ်ခါထပ်ခါအသုံးပြုနိုင်သည်။ multiple use ဟုခေါ်သည်။ ပုံ 3-16 တွင် limit switch (LS1) သည် input module – reference number 20 ဖြင့် connect လုပ်ထားသည်။ ထို ref no 20 PLC program ထဲတွင် Normally Open နှင့် Normally Close များ အခါ များစွာသုံးနိုင်သည်။



ပုံ 3-16 Input 20 has multiple contacts in the PLC control program.

Table 3-5 ၌ပါရှိသော Symbol များသည် Relay Control Logic မှ Contact Symbolic Logic သို့ပြောင်းရာတွင် အသုံးပြုသည်။ ထို Symbol များသည် Ladder Diagram များ၏ basic instruction များဖြစ်သည်။ (သို့သော် timer နှင့် counter instruction များမပါဝင်ပေ။) Chapter 9 တွင် ပို၍ ခက်ခဲ အဆင့်မြင့်သည့် (Advanced) instruction များကို ဖော်ပြထားသည်။

Normally open contact ဖြစ်သည်။ control logic ၏ မည်သည့် input ကိုမဆို ဤ symbol ဖြင့် ဖော်ပြနိုင်သည်။ Input တစ်ခုသည် connect လုပ်ထားသည့် switch တစ်ခုလည်းဖြစ်နိုင်သည်။ connect လုပ်ထားသည့် output တစ်ခု၏ contact လည်းဖြစ်နိုင်သည်။ Internal output တစ်ခုလည်းဖြစ်နိုင်သည်။ အကယ်၍ status သည် 1 ဖြစ်လျှင် contact သည် close ဖြစ်သွားပြီး လျှပ်စစ် current သည် contact ကိုဖြတ်၍ စီးဆင်းသွားလိမ့်မည်။ အကယ်၍ reference input/output သည် ဝဖြစ်လျှင် contact သည် open ဖြစ်မြဲဖြစ်နေလိမ့်မည်။ Open ဖြစ်နေသောကြောင့် Current သည် Contact ကို ဖြတ်၍ စီးဆင်းသွား လိမ့်မည်မဟုတ်ပေ။

Normally Closed Contact:

Output: input logic များပေါင်း၍ ရရှိသည့် output ကိုဆိုလိုသည်။ Output များကို Device သို့မဟုတ် internal input ဖြင့် ချိတ်ဆက် (connect) လုပ်နိုင်သည်။ အကယ်၍ ဘယ်ဘက်မှ ညာဘက်သို့ (left to right path) ရှိ condition များအားလုံးမှန်လျှင် output သည် energized ဖြစ်လိမ့်မည်။ တနည်း ON လိမ့်မည်။

အကယ်၍ ဘယ်ဘက်မှ ညာဘက်သို့ (left to right path) ရှိ condition များ အားလုံးမှန်လျှင် (True ဖြစ်လျှင်) output သည် De-energized ဖြစ်လိမ့်မည်။ OFF ဖြစ်လိမ့်မည်။ Hardwired logic မှ PLC logic သို့ပြောင်းရန်အတွက် Guide Line 7 ခုကို အောက်တွင်ဖော်ပြထားသည်။ (PLC Contact Symbol 7 ခုကို အောက်တွင်ဖော်ပြထားသည်။)

Normally Open Contact: Program ရှိ Symbol သည် 1 ဖြစ်နေလျှင် Contact ကို Close အဖြစ်သို့ပြောင်းသွားလိမ့်မည်။ Symbol ကို refer လုပ်သည့် signal သည် Activated ဖြစ်လိမ့်မည်။ (သို့) ON ဖြစ်လိမ့်မည်။ (သို့) Close ဖြစ်လိမ့်မည်။

Normally Closed Contact

Output: Rung တစ်ခုရှိ output သည် energized ဖြစ်လိမ့်မည်။ ဘယ်မှညာသို့ လမ်းကြောင်းတစ်လျှောက်ရှိ (left to right path) contact များအားလုံး close ဖြစ်လျှင်၊ ချွင်းချက်အဖြစ် Power သည် ညာမှဘယ်သို့ စီးဆင်းသည့်အခါမှလွဲလျှင်၊ Output တစ်ခုသည် ချိတ်ဆက်ထားသည့် device ကို control လုပ်သည်။ သို့မဟုတ် program အတွင်းရှိ internal point ကို control လုပ်သည်။ Internal output သည် field device ကို control မလုပ်နိုင်ပေ။ PLC အတွင်း၌သာ interlocking ပြုလုပ်ရန်အတွက် internal output ကိုအသုံးပြုသည်။ Input: Input Contact Symbol သည် ချိတ်ဆက်ထားသည့် (connect လုပ်ထားသည့်) input သော်၎င်း၊ internal output သော်၎င်း၊ ချိတ်ဆက်ထားသည့် connect ကိုသော်၎င်း input အဖြစ် သတ်မှတ်သည်။

Contact address: Program Symbol တိုင်းကို address များသတ်မှတ်ပေးထားသည်။ address များဖြင့် reference လုပ်ထားသည်။ Symbol သည် ချိတ်ဆက်ထားသည့် input/output ကို reference လုပ်ထားလျှင် ထို Symbol ၏ address သည် ချိတ်ဆက်ထားသည့် terminal ၏ address ဖြစ်သည်။

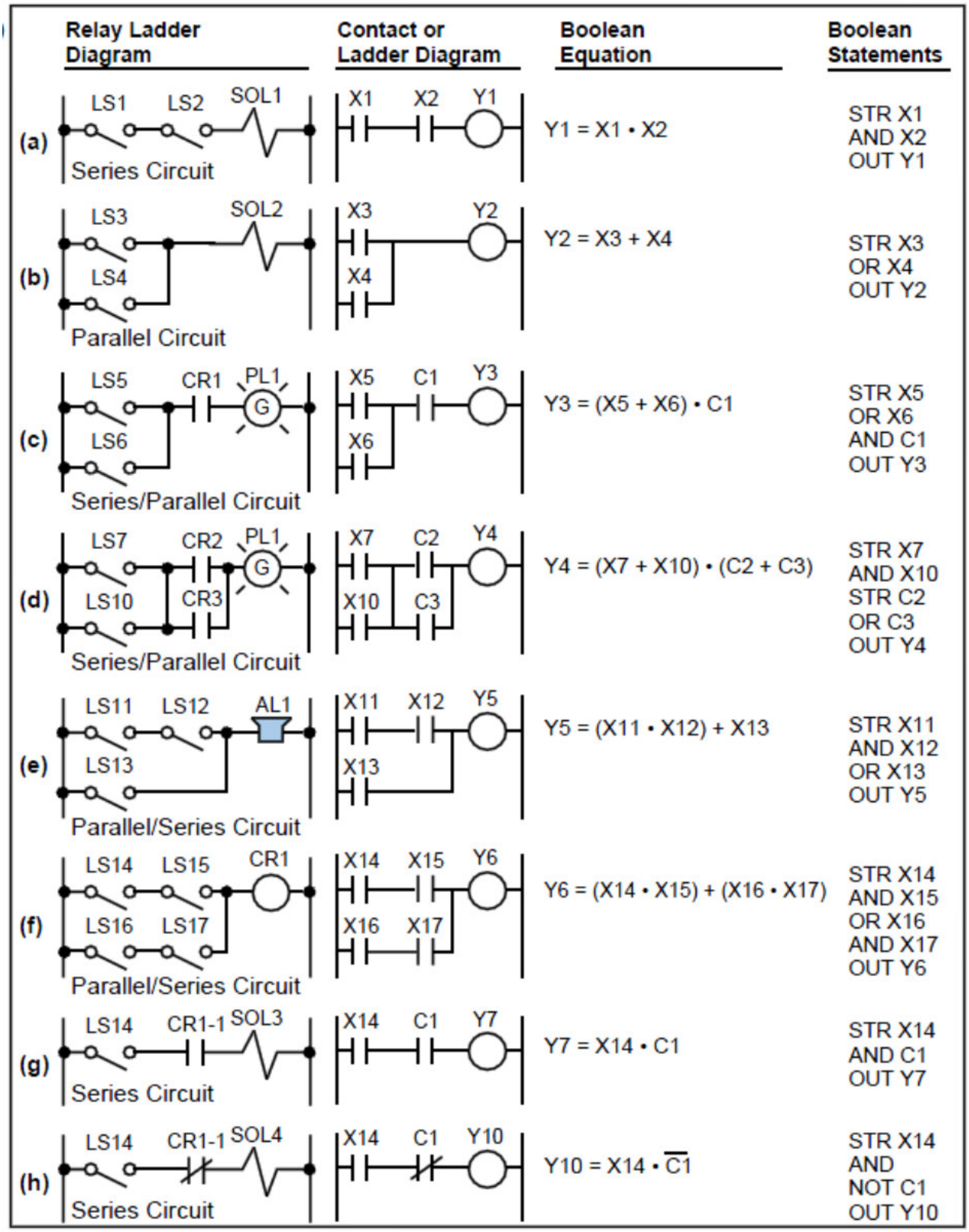
Repeated use of contact: program အတွင်း၌ input/output နှင့် internal output တို့ကို အလိုရှိသလောက်အခါများစွာ အသုံးပြုနိုင်သည်။

Logic format: အလိုရှိသည့် output logic အတိုင်းရစေရန်အတွက် contact များကို အတန်းလိုက် (series) သို့မဟုတ် အပြိုင်(parallel) ချိတ်ဆက်နိုင်သည်။ အတန်းလိုက်ရှိနေသည့် contact အရေအတွက် (no of series contacts) နှင့် အပြိုင်ရှိနေသည့် Branch အရေအတွက် (no of parallel Branch) တို့သည် PLC အမျိုးအစားနှင့် အရွယ်အစားကိုလိုက်၍ ကွဲပြားသည်။

ဇယား 3-6a တွင် hardwired series နှင့် parallel circuit တို့ကို program logic အဖြစ်သို့ ပြောင်းလဲပုံကို ဖော်ပြထားသည်။ Series circuit သည် Boolean AND operation နှင့်ညီသည်။ (Equivalent ဖြစ်သည်။) ထို့ကြောင့် output "ON" ဖြစ်စေရန်အတွက်) input အားလုံးသည် ON ရမည်။ Parallel circuit သည် Boolean OR operation နှင့် ညီသည်။ (Equivalent ဖြစ်သည်။) ထို့ကြောင့် output "ON" ဖြစ်စေရန်အတွက် (Activate (သို့) Energized) input အားလုံး အနက်မှ အနည်းဆုံးတစ်ခုသည် ON ဖြစ်ရမည်။

STR သည် New rung တစ်ခုအတွက် START ဟုအဓိပ္ပာယ်ရသည်။ Out သည် rung တစ်ခုအတွက် output ဟုဆိုလိုသည်။

ဇယား 3-6 b သည် ဇယား 3-6 aကို အသေးစိတ်ရှင်းလင်းဖော်ပြထားသည်။


Relay Ladder Diagram
Contact or Ladder Diagram
Boolean Equation
Boolean Statements
(a)
LS1 LS2 SOL1
Series Circuit
X1 X2 Y1
Y1 = X1 • X2
STR X1
AND X2
OUT Y1
(b)
LS3 SOL2
LS4
Parallel Circuit
X3 Y2
X4
Y2 = X3 + X4
STR X3
OR X4
OUT Y2
(c)
LS5 CR1 PL1
G
LS6
Series/Parallel Circuit
X5 C1 Y3
X6
Y3 = (X5 + X6) • C1
STR X5
OR X6
AND C1
OUT Y3
(d)
LS7 CR2 PL1
G
LS10 CR3
Series/Parallel Circuit
X7 C2 Y4
X10 C3
Y4 = (X7 + X10) • (C2 + C3)
STR X7
AND X10
STR C2
OR C3
OUT Y4
(e)
LS11 LS12 AL1
LS13
Parallel/Series Circuit
X11 X12 Y5
X13
Y5 = (X11 • X12) + X13
STR X11
AND X12
OR X13
OUT Y5
(f)
LS14 LS15 CR1
LS16 LS17
Parallel/Series Circuit
X14 X15 Y6
X16 X17
Y6 = (X14 • X15) + (X16 • X17)
STR X14
AND X15
OR X16
AND X17
OUT Y6
(g)
LS14 CR1-1 SOL3
Series Circuit
X14 C1 Y7
Y7 = X14 • C1
STR X14
AND C1
OUT Y7
(h)
LS14 CR1-1 SOL4
Series Circuit
X14 C1 Y10
Y10 = X14 • C̄1
STR X14
AND
NOT C1
OUT Y10

(a) **Series Circuit.** In this circuit, if both switches LS1 and LS2 are closed, the solenoid SOL1 will energize.

(b) **Parallel Circuit.** In this circuit, if either of the two switches LS3 or LS4 closes, the solenoid SOL2 will energize.

(c) & (d) **Series/Parallel Circuits.** In a series/parallel circuit, the result of ORing two or more inputs is ANDed with one or more series or parallel inputs. In both of these examples, all of the relay circuit elements are normally open and must be closed to activate the pilot lights. Normally open contacts are used in the program.

(e) & (f) **Parallel/Series Circuits.** In a parallel/series circuit, the result of ANDing two or more inputs is ORed with one or more series inputs. In both of these examples, all of the relay circuit elements are normally open and must be closed to activate the output device. Normally open contacts are used in the program.

(f) **Internal Outputs.** Circuit **(f)** controls an electromechanical control relay. Control relays do not normally drive output devices, but rather drive other relays. They are used to provide additional contacts for interlocking logic. The internal output provides the same function in software; however, the number of contacts are unlimited and can be either normally open or normally closed.

(g) **Normally Open Contacts.** In the series circuit **(g)**, the solenoid will energize if LS14 closes and CR1-1 is energized. CR1-1 is a contact from the control relay CR1 in circuit **(f)** and closes whenever CR1 is energized. In the program, CR1 was replaced by the internal output C1; therefore, the program uses a normally open contact from the internal output C1. SOL3 will energize when LS14 closes and C1 is energized.

(h) **Normally Closed Contacts.** In circuit **(h)**, the solenoid will energize if LS14 closes and CR1-1 is not energized. The program uses a normally closed contact from the internal output C1. SOL3 will stay energized as long as the limit switch is closed and C1 is not energized.