Programmable Logic Controller

Jie LING

meejling@nuaa.edu.cn

Office: Room #423, Building 17, MC

Department of Mechatronics Engineering,
College of Mechanical and Electrical Engineering

Course Description

Overview

CH1 Introduction to PLCs	
CH2 CP1 PLC	
Review & Quiz 1	
CH3 Instructions	**3 sessions**
CH4 PLC programming	**3 sessions**
Review & Quiz 2	**2 session**

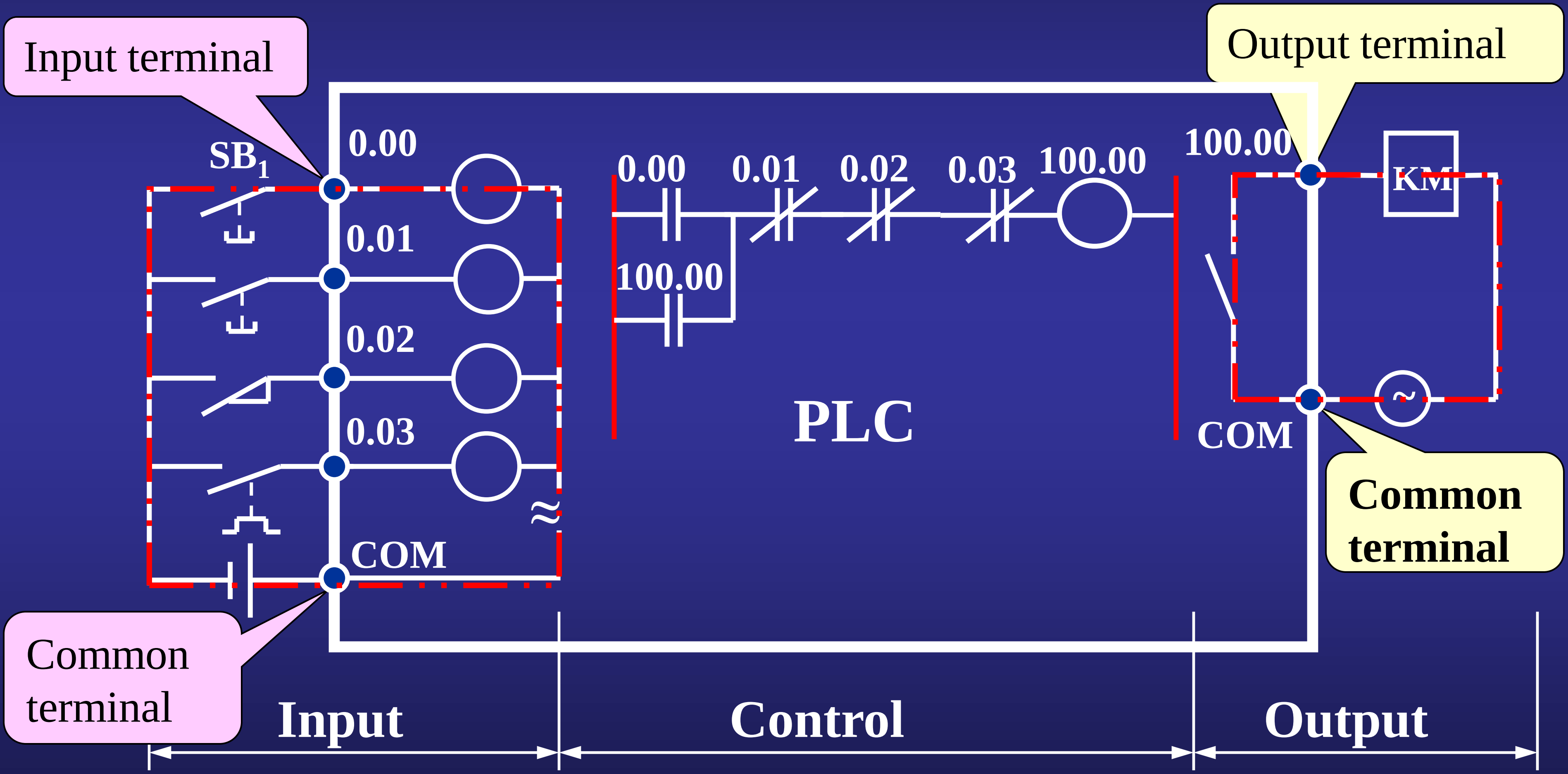
Course Description

Objectives:

- Understand the working principle of PLC
- List the components and specifications of the PLC CP1
- Use the instructions of CP1
- Design PLC control systems

Brief Review of Ch1 & 2

Architecture of PLC



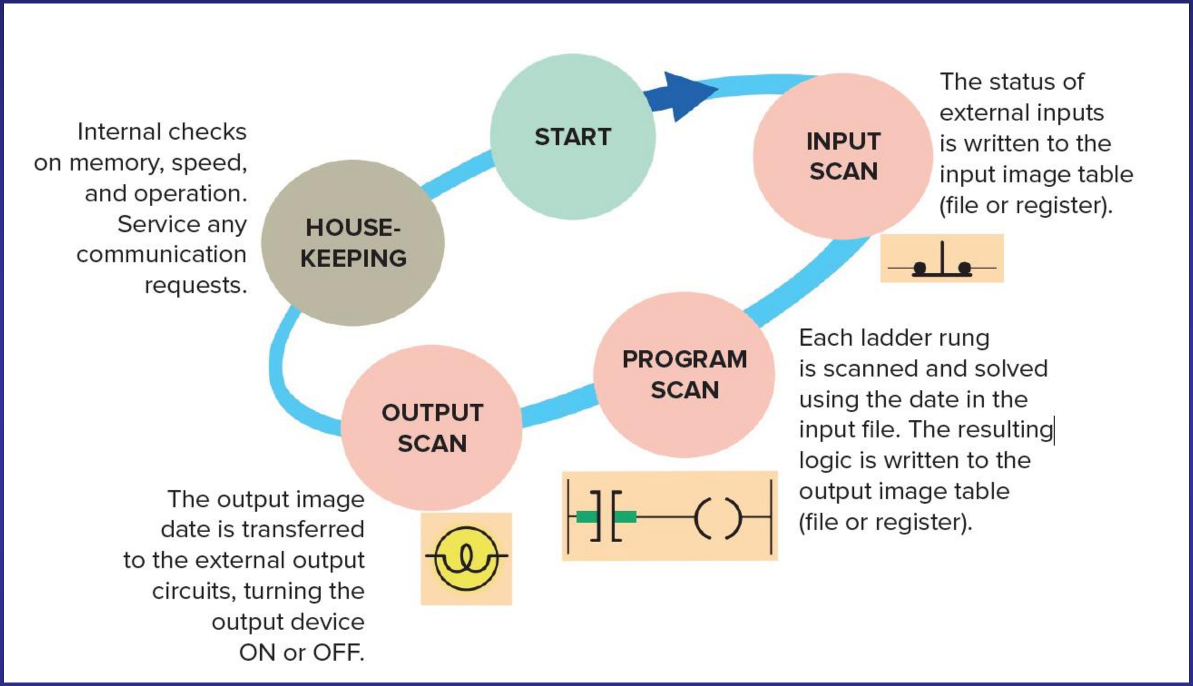
Brief Review of Ch1 & 2

Principle of Operation

Cycle by cycle scan

Program Scan Cycle

- ✓ Input scan
- ✓ Program scan
- ✓ Output scan
- ✓ Housekeeping duties



The status of external inputs is written to the input image table (file or register).

Each ladder rung is scanned and solved using the date in the input file. The resulting logic is written to the output image table (file or register).

The output image date is transferred to the external output circuits, turning the output device ON or OFF.

Internal checks on memory, speed, and operation. Service any communication requests.

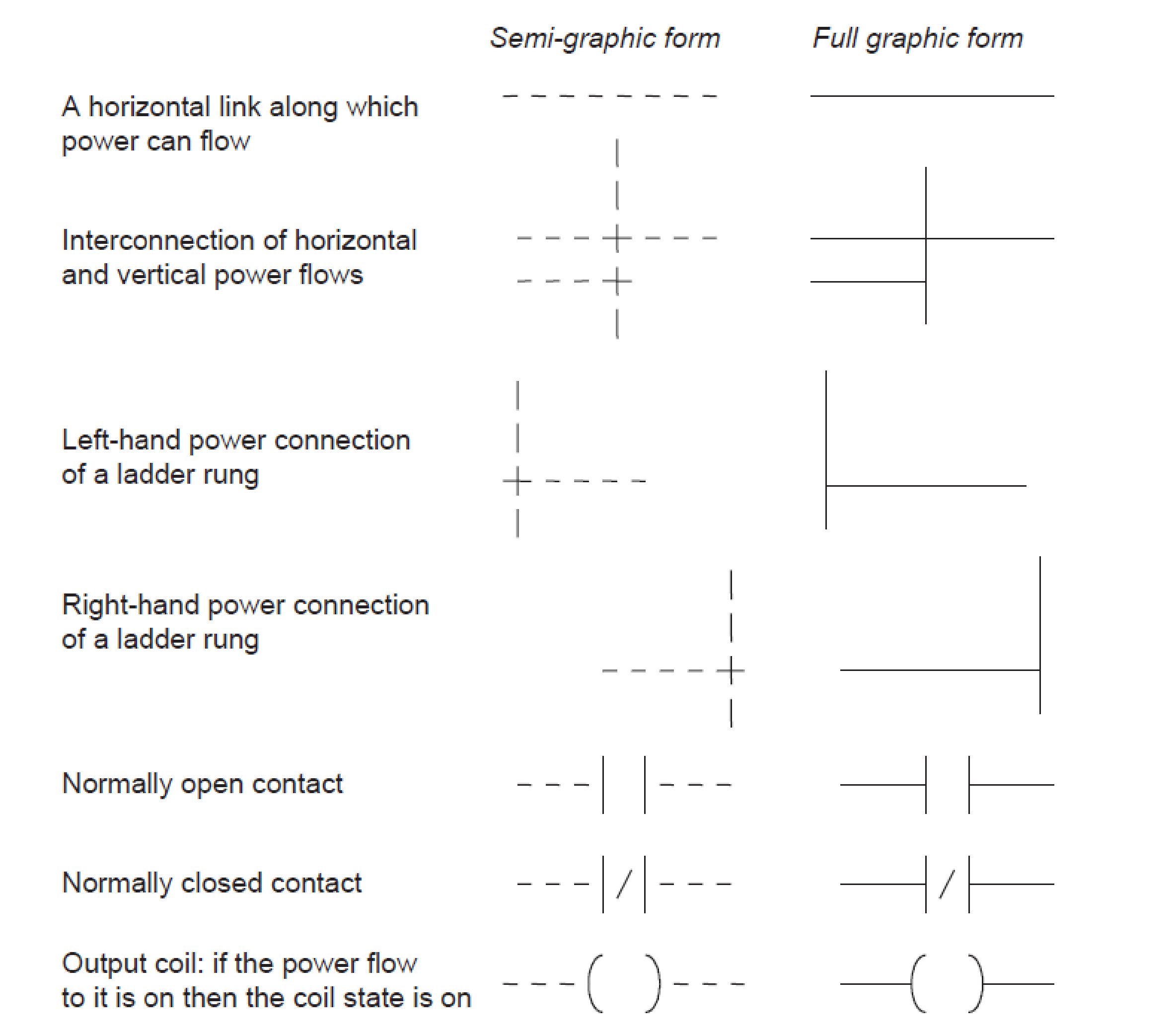
Scan Cycle Time

- ✓ The speed of the processor module
- ✓ **The length of the ladder program**
- ✓ **The type of instructions executed**
- ✓ The actual ladder true/false conditions

Brief Review of Ch1 & 2

Ladder diagram - Standard

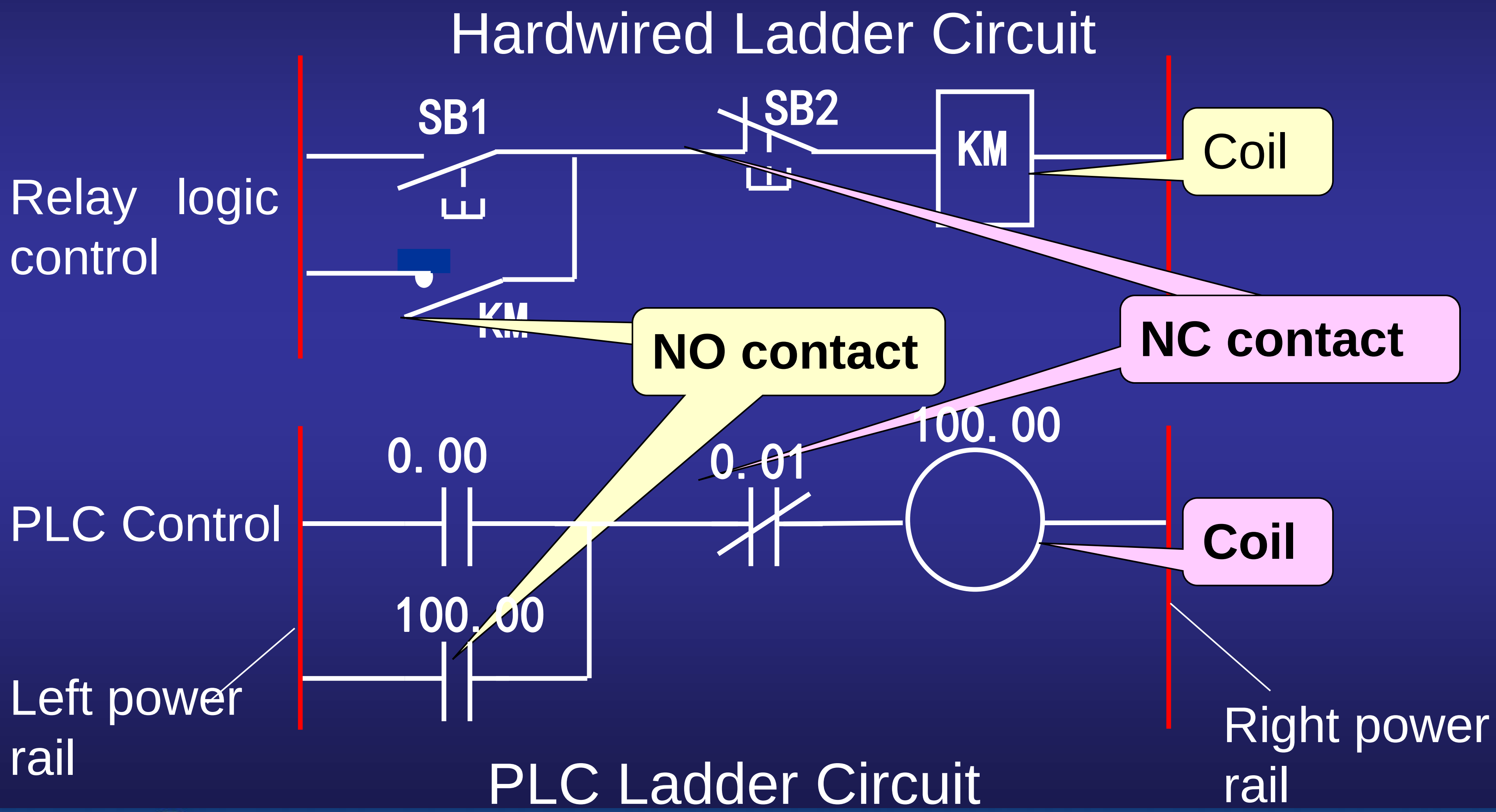
Standard IEC 61131-3 symbols



	Semi-graphic form	*Full graphic form*
A horizontal link along which power can flow		
Interconnection of horizontal and vertical power flows		
Left-hand power connection of a ladder rung		
Right-hand power connection of a ladder rung		
Normally open contact		
Normally closed contact		
Output coil: if the power flow to it is on then the coil state is on		

Brief Review of Ch1 & 2

- Ladder diagram – Example



Hardwired Ladder Circuit

Relay logic control

PLC Control

Left power rail

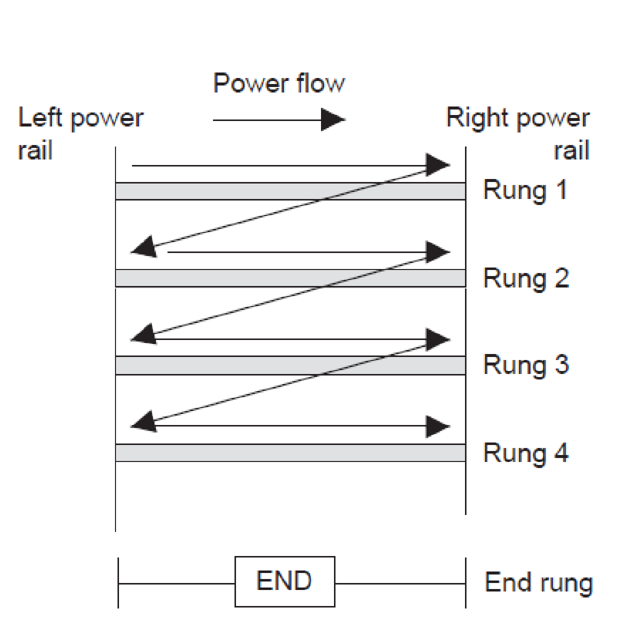
Right power rail

PLC Ladder Circuit

Brief Review of Ch1 & 2

Ladder diagram - Conventions

- The vertical lines of the diagram represent the power rails between which circuits are connected. The power flow is taken to be from the left-hand vertical across a rung.



- Each rung on the ladder defines one operation in the control process.
- A ladder diagram is read from left to right and from top to bottom.
- When the PLC is in its run mode, it goes through the entire ladder program to the end, the end rung of the program being clearly denoted, and then promptly resumes at the start

Brief Review of Ch1 & 2

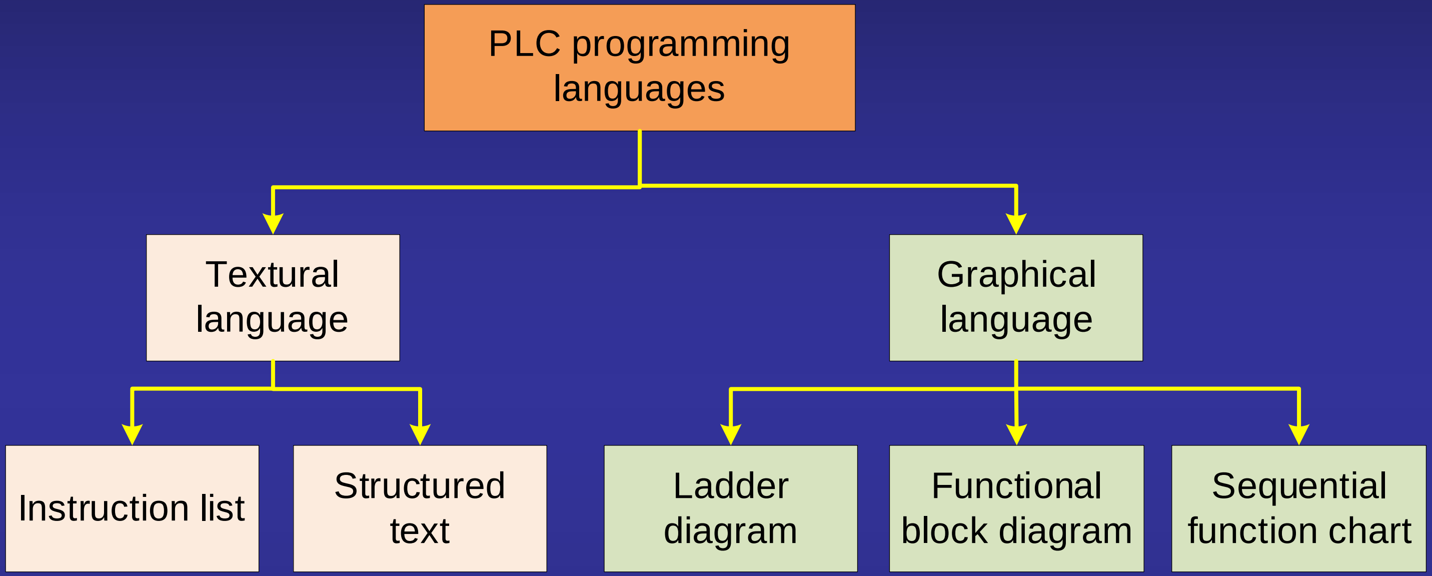
- This procedure of going through all the rungs of the program is termed a cycle.
- The end rung might be indicated by a block with the word END or RET, for return, since the program promptly returns to its beginning. The scan time depends on the number of runs in the
- program, taking about 1 ms per 1000 bytes of program and so typically ranging from about 10 ms up to 50 ms.
- Each rung must start with an input or inputs and must end with at least one output. The term input is used for a control action, such as closing the contacts of a switch. The term output is used for a device connected to the output of a PLC, such as a motor.
- As the program is scanned, the outputs are not updated instantly, but the results stored in memory and all the outputs are updated simultaneously at the end of the program scan.

Brief Review of Ch1 & 2

- **Electrical devices are shown in their normal condition.**
  - □ A switch, that is normally open until some object closes it, is shown as open on the ladder diagram
  - □ A switch that is normally closed is shown closed
- **A particular device can appear in more than one rung of a ladder.**
  - □ A relay that switches on one or more devices
  - □ The same letters and/or numbers are used to label the device in each situation
- **The inputs and outputs are all identified by their addresses, the notation used depends on the PLC manufacturer**

What we do in Ch3 & 4

Standard languages associated with PLC programming (Standard IEC 61131)



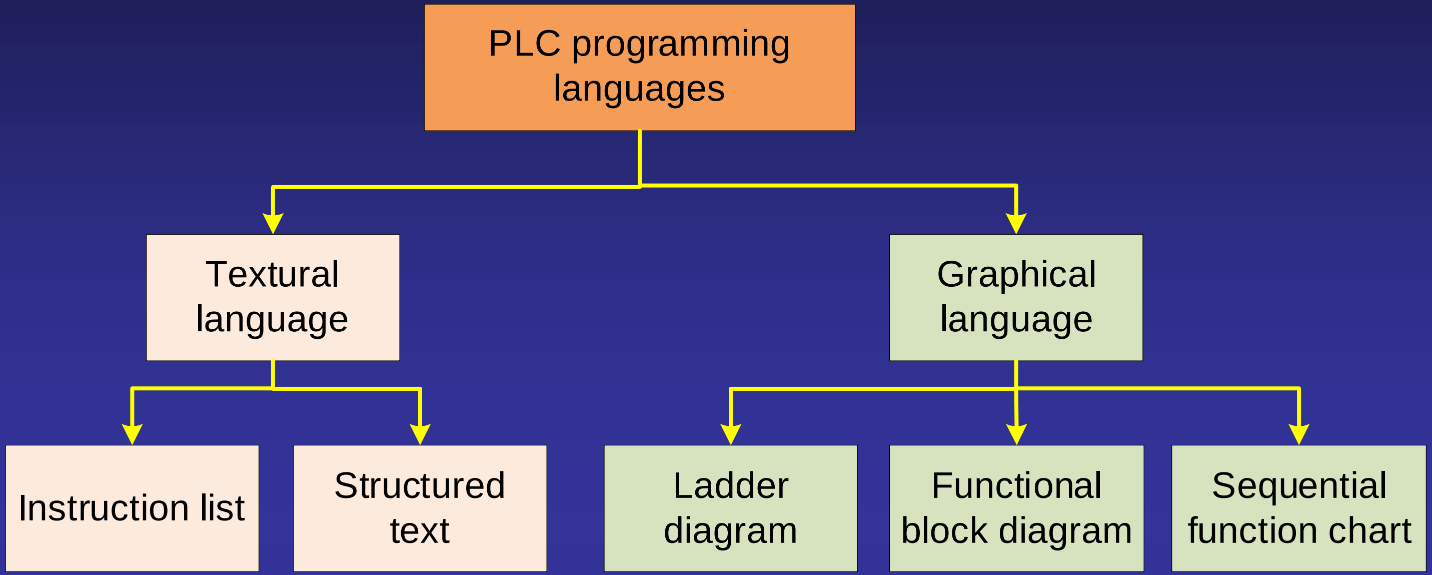
- **Instruction List (IL)**

a low-level, text-based language that uses mnemonic instructions.

- **Structured Text (ST)**

a high-level, text-based language such as BASIC, C, or PASCAL specifically developed for industrial control applications.

What we do in Ch3 & 4



- **Ladder Diagram (LD)**

a symbolic depiction of instructions arranged in rungs similar to ladder formatted schematic diagrams.

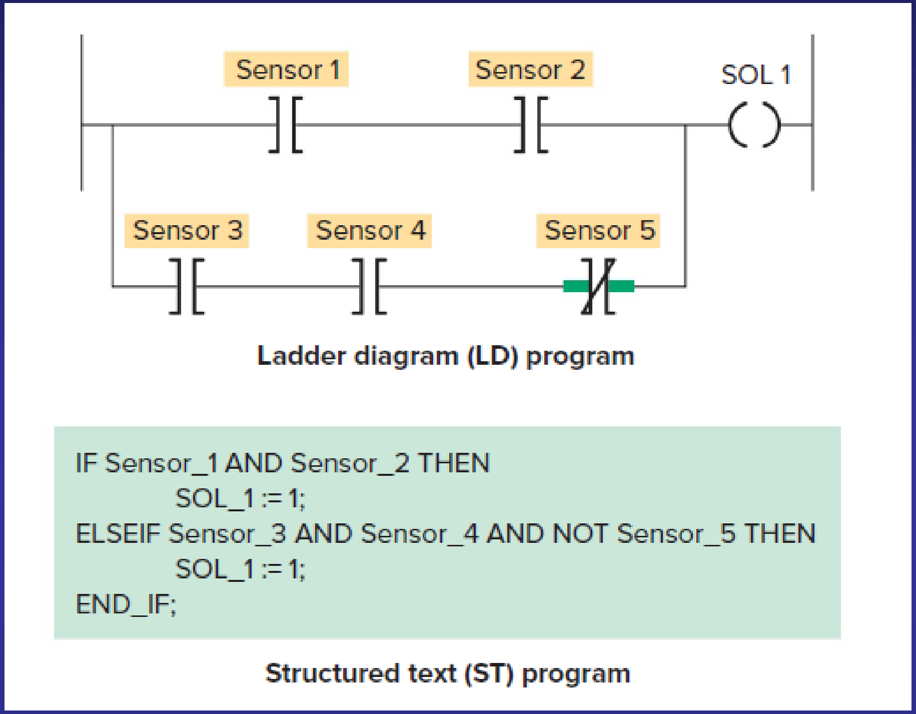
- **Function Block Diagram (FBD)**

a graphical depiction of process flow using simple and complex interconnecting blocks. (i.g., Boolean operations: Y = AB+C)

- **Sequential Function Chart (SFC)**

a graphical depiction of interconnecting steps, actions, and transitions.

What we do in Ch3 & 4



Ladder diagram (LD) program

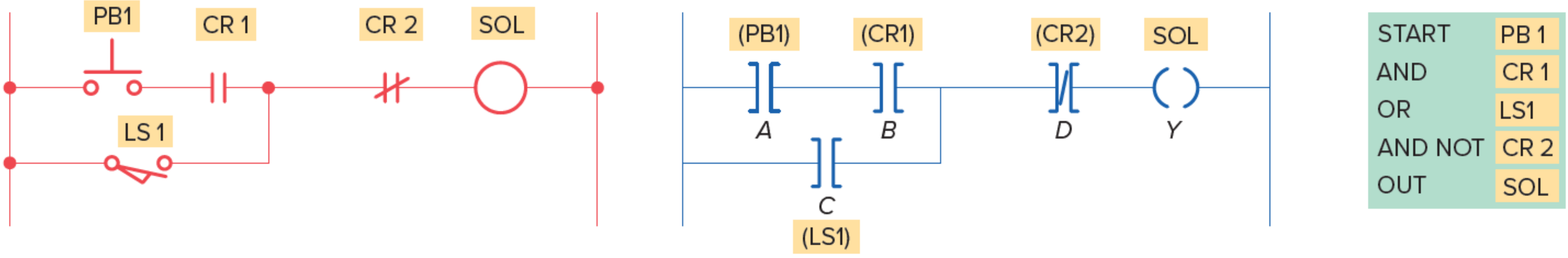
```
IF Sensor_1 AND Sensor_2 THEN
          SOL_1 := 1;
ELSEIF Sensor_3 AND Sensor_4 AND NOT Sensor_5 THEN
          SOL_1 := 1;
END_IF;
```

Structured text (ST) program

- A high-level text language primarily used to implement complex procedures that cannot be easily expressed with graphical languages
- Structured text uses statements to define what to execute
- For this application, the objective is to energize SOL 1 whenever either one of the two following circuit conditions exists:
  - Sensor 1 and Sensor 2 switches are both closed.
  - Sensor 3 and Sensor 4 switches are both closed and
  - Sensor 5 switch is open.

What we do in Ch3 & 4

Comparison of ladder diagram and instruction list programming



START	PB 1
AND	CR 1
OR	LS1
AND NOT	CR 2
OUT	SOL

(*a*) Hardwired relay control circuit

(*b*) Equivalent ladder diagram (LD) program

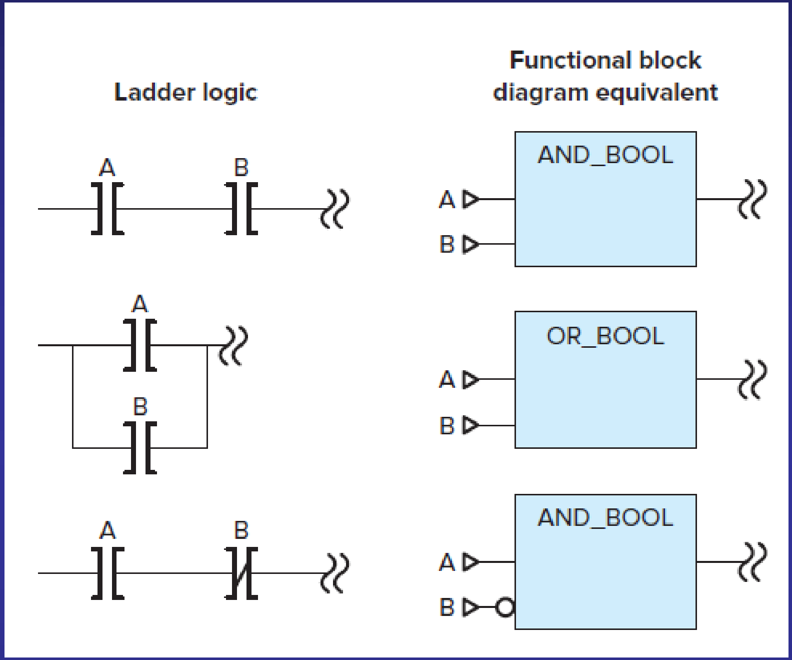
(c) Equivalent instruction list (IL) program

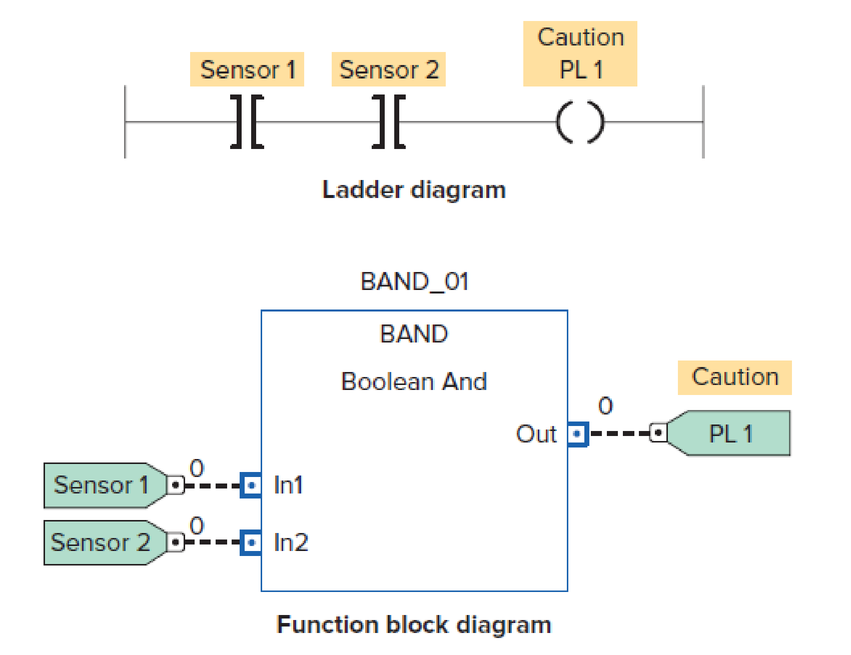
- **The original relay hardwired control circuit**
- **The equivalent logic ladder diagram programmed into a controller**
- **The program using the instruction list programming language**

- **Ladder diagram language is the most commonly used PLC language and is designed to mimic relay logic.**
- **The ladder diagram is popular for those who prefer to define control actions in terms of relay contacts and coils, and other functions as block instructions.**
- **The instructional list consists of a series of instructions that refer to the basic AND, OR, and NOT logic gate functions.**

What we do in Ch3 & 4

Function block diagram equivalents to ladder logic contacts

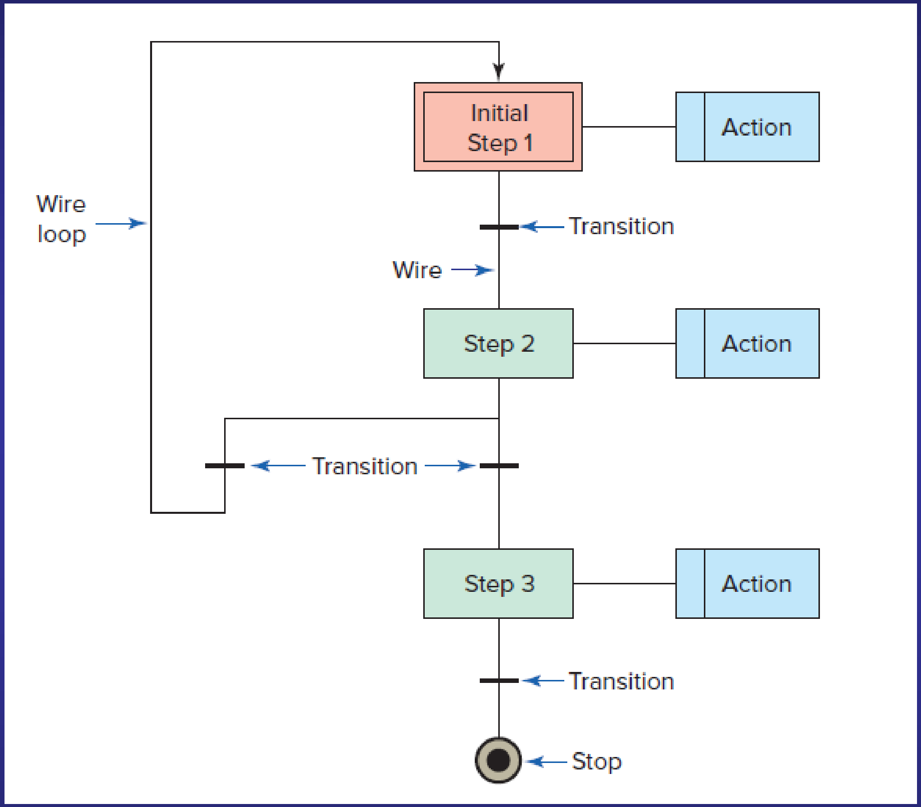


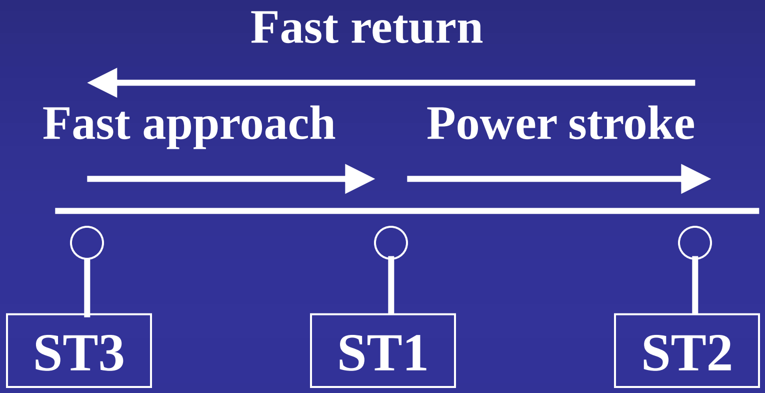


- The instructional list consists of a series of instructions that refer to the basic AND, OR, and NOT logic gate functions.
- Typical types of function block: logic, timers, and counters.
- Complex system → Blocks of functionality: Electrical/electronic block diagrams
- Primary concept: data flow. Data flow on a path from inputs, through function blocks or instructions, and then to outputs.

What we do in Ch3 & 4

Major elements of a sequential function chart program.





- Accommodate the programming of more advanced processes
- Steps with multiple operations happening in parallel branches

Ch3

Instructions of CP1

Ch3 Instruction of CP1

- Overview
- Basic instructions ★
- Application instructions ★

3.1 Overview

3.1.1 Types of instruction

✓ Basic instructions

- I/O
- Timing
- Counting

✓ Application instructions

- Data Comparison
- Data Transmission
- Data Shifting
- Data Conversion
- Arithmetical operation
- Communication

3.1 Overview

3.1.2 Format of instruction

Operator (function code) operand1

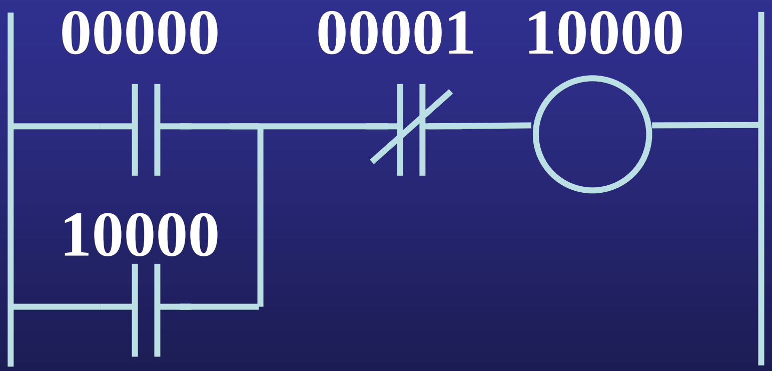
operand2

operand3

Function

A three-digit number

Objects executed by the instruction. The number of this is determined by the instruction



LD	00000
OR	10000
AND NOT	00001
OUT	10000

3.1 Overview

3.1.2 Format of instruction

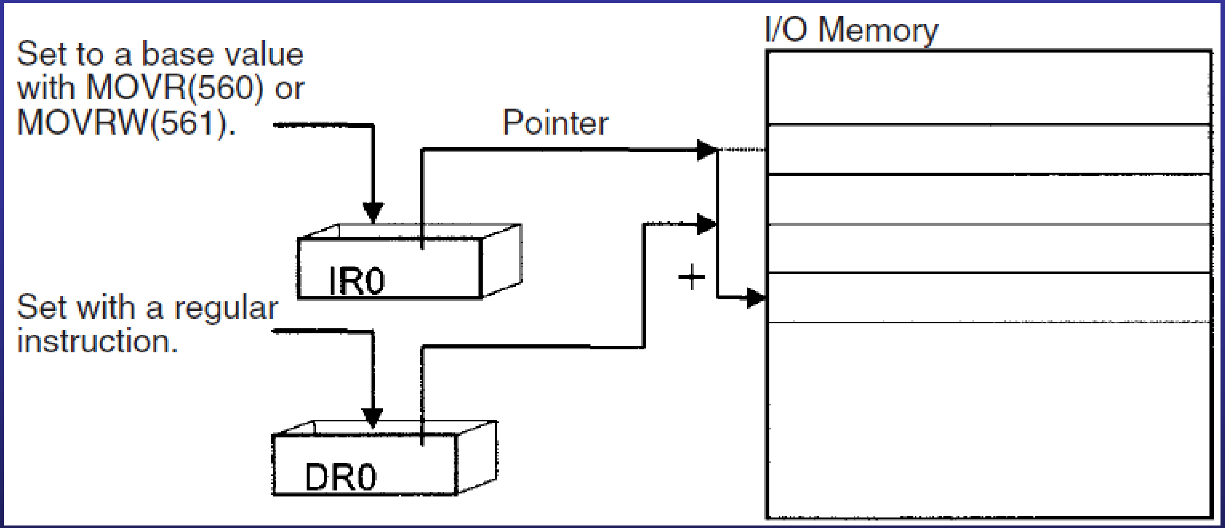
Operand

- Most instructions have at least one or more operands associated with them
- These are sometimes input as the actual numeric values (i.e., as constants), but are usually the addresses of data area words or bits that contain the data to be use
- A bit whose address is designated as an operand is called an operand bit; a word whose address is designated as an operand is called an operand word.

3.1 Overview

➢operand ✓Can be Word address, Relay or Constant

- 16 Index Registers (IR0 to IR15), used for indirect addressing.
- 16 Data Registers (DR0 to DR15) are used to offset the PLC memory addresses in IR when addressing words indirectly.


I/O Memory
Set to a base value with MOVR(560) or MOVRW(561).
Pointer
IR0
+
Set with a regular instruction.
DR0

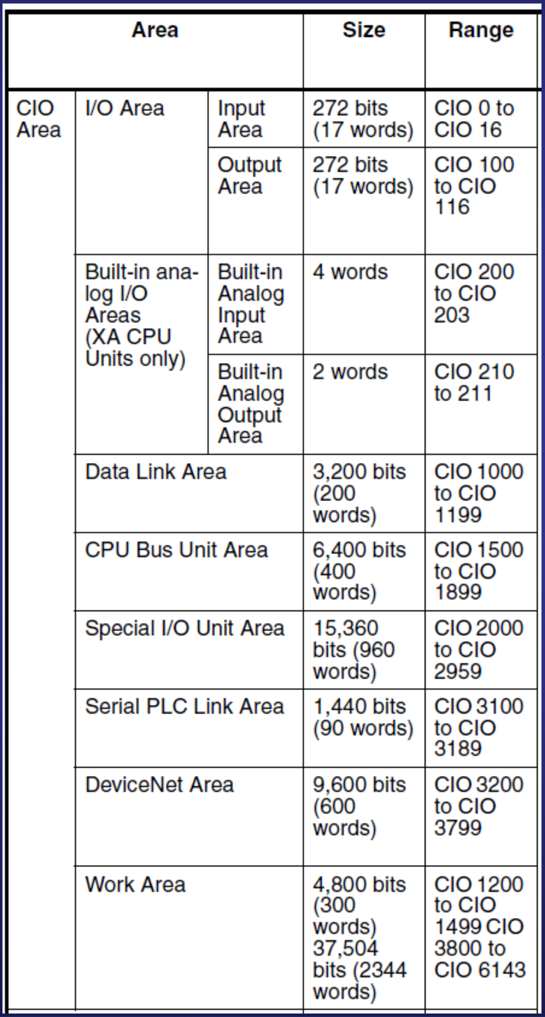
MOV(021) #0002
IR0

MOV(021) #0001
DR0, IR0

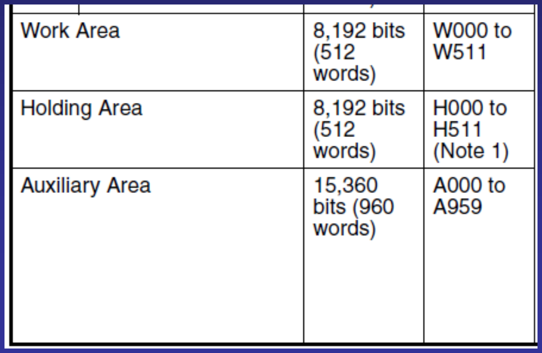
3.1 Overview

➢operand

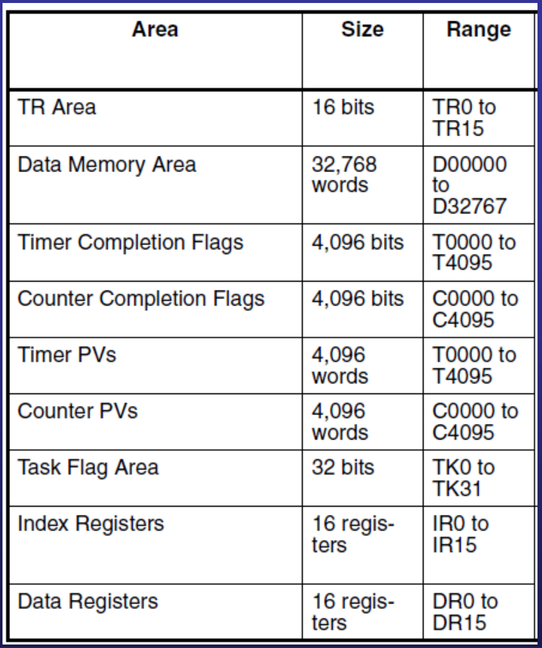
✓Can be Word address, Relay or Constant



Area			Size	Range
CIO Area	I/O Area	Input Area	272 bits (17 words)	CIO 0 to CIO 16
		Output Area	272 bits (17 words)	CIO 100 to CIO 116
	Built-in analog I/O Areas (XA CPU Units only)	Built-in Analog Input Area	4 words	CIO 200 to CIO 203
		Built-in Analog Output Area	2 words	CIO 210 to 211
	Data Link Area		3,200 bits (200 words)	CIO 1000 to CIO 1199
	CPU Bus Unit Area		6,400 bits (400 words)	CIO 1500 to CIO 1899
	Special I/O Unit Area		15,360 bits (960 words)	CIO 2000 to CIO 2959
	Serial PLC Link Area		1,440 bits (90 words)	CIO 3100 to CIO 3189
	DeviceNet Area		9,600 bits (600 words)	CIO 3200 to CIO 3799
	Work Area		4,800 bits (300 words) 37,504 bits (2344 words)	CIO 1200 to CIO 1499 CIO 3800 to CIO 6143



Work Area	8,192 bits (512 words)	W000 to W511
Holding Area	8,192 bits (512 words)	H000 to H511 (Note 1)
Auxiliary Area	15,360 bits (960 words)	A000 to A959



Area	Size	Range
TR Area	16 bits	TR0 to TR15
Data Memory Area	32,768 words	D00000 to D32767
Timer Completion Flags	4,096 bits	T0000 to T4095
Counter Completion Flags	4,096 bits	C0000 to C4095
Timer PVs	4,096 words	T0000 to T4095
Counter PVs	4,096 words	C0000 to C4095
Task Flag Area	32 bits	TK0 to TK31
Index Registers	16 registers	IR0 to IR15
Data Registers	16 registers	DR0 to DR15

- **CIO Area (CIO)**
- **Work Area (W)**
- **Holding Area (H)**
- **Auxiliary Area (A)**
- **Temporary Relay Area (TR)**
- **Data Memory Area (D)**
- **Timer Area (T)**
- **Counter Area (C)**
- **Task Flag Area (TK)**
- **Data Registers (DR)**
- **Index Registers (IR)**
- **Condition Flag Area**
- **Clock Pulse Area**

MOV(021) #3560
A448.02

3.1 Overview

➢operand ✓Can be Word address, Relay or Constant

✓**Signed Decimal**
✓**Unsigned Decimal**
✓**BCD** (Binary Coded Decimal)
✓**BIN** (Binary Number)

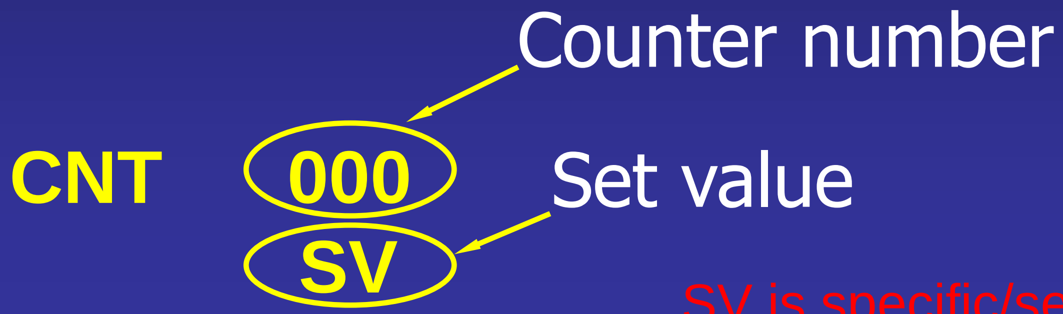
Format	Sign	Range
Unsigned Decimal	&	&0~&65535
Signed Decimal	±	-32768~+32767
BIN	#	#0000~#FFFF
BCD	#	#0~#9999

```
MOV(021)   #3560
           A448.02
```

3.1 Overview

➢operand

✓The number of operands depends on instruction



SV is specific/set value
PV is present value

if SV=200

The set value is the data in CIO 2.00

If SV=#0200

The set value is 200

3.1 Overview

➢operand

✓When operand is constant, place Sign before constant

✓Indirect DM addressing is specified by placing an * before D, i.e. *D × × × × ×

✓When an indirect DM address is specified, the designated DM word will contain the address of the DM word that contains the data that will be used as the operand of the instruction.

✓When using indirect addressing, the address of the desired word must be in BCD, and it must be within the DM area.

3.1 Overview

CNT 0000

D01000

Direct DM addressing

✓If the data in D01000 is 00010, then the set value is 00010

CNT 0000

***D01000**

Indirect DM addressing

✓If the data in D01000 is 00010 and the data in DM00010 is 2500, then the set value is 2500

3.1 Overview

➢Differentiated and non-differentiated form of instructions

✓**Differentiated instructions**

an @ in front of the mnemonic instruction**.**

✓Difference

- A non-differentiated instruction is executed each time it is scanned as long as its execution condition is ON.
- A differentiated instruction is executed only once after its execution condition goes from OFF to ON. If the execution condition has not changed or has changed from ON to OFF since the last time the instruction was scanned, the instruction will not be executed.

3.1 Overview

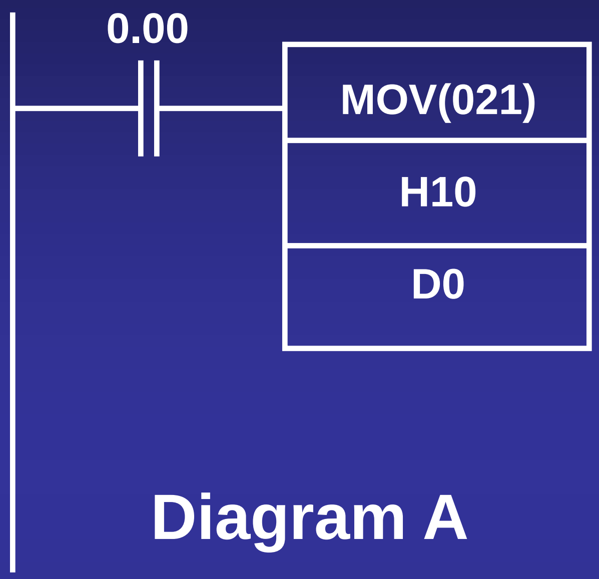


Diagram A

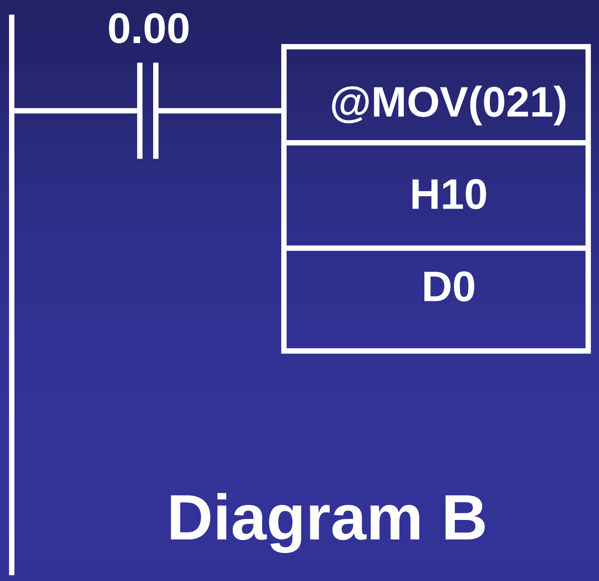


Diagram B

✓In diagram A, the non-differentiated MOV(21) will move the content of HR 10 to DM 0000 whenever it is scanned with 00000.

✓If the cycle time is 80 ms and 00000 remains ON for 2.0 seconds, this move operation will be performed 25 times and only the last value moved to DM 0000 will be preserved there.

3.1 Overview

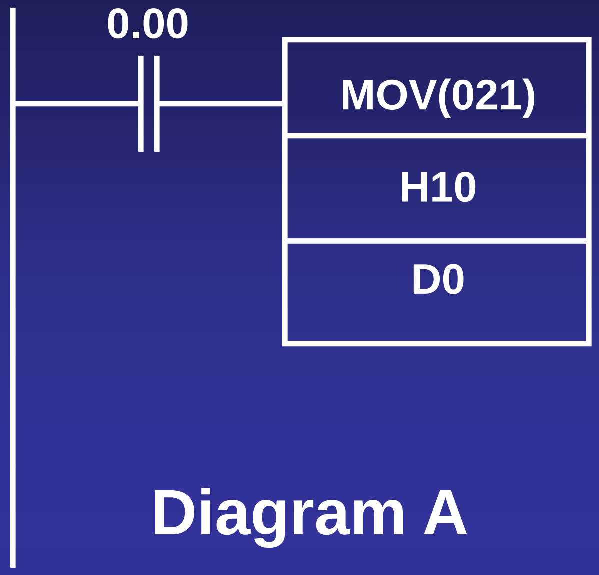


Diagram A

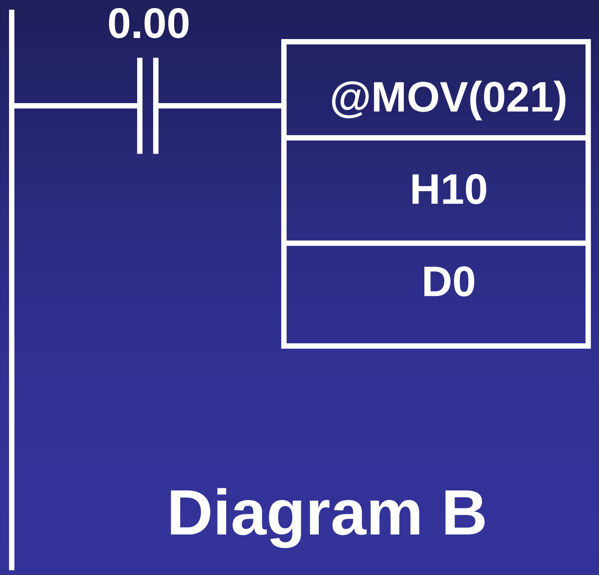


Diagram B

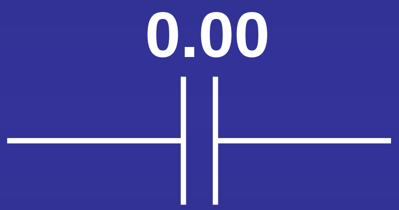
✓In diagram B, the differentiated @MOV(21) will move the content of HR 10 to DM 0000 only once after 00000 goes ON.

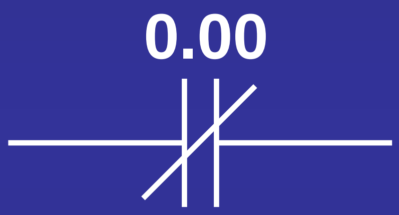
✓Even if 00000 remains ON for 2.0 seconds with the same 80 ms cycle time, the move operation will be executed only once during the first cycle in which 00000 has changed from OFF to ON. Because the content of HR 10 could very well change during the 2 seconds while 00000 is ON, the final content of DM 0000 after the 2 seconds could be different depending on whether MOV(21) or @MOV(21) was used.

3.1 Overview

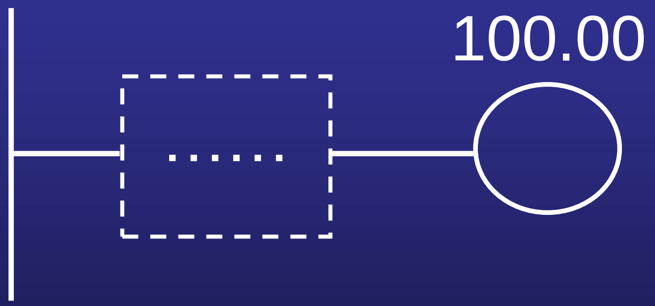
3.1.3 Basic terms

- ➢Normally Open (NO) condition
- ➢Normally closed (NC) condition





- ➢Execution Conditions



- ➢Operand Bits

3.2 Basic instructions

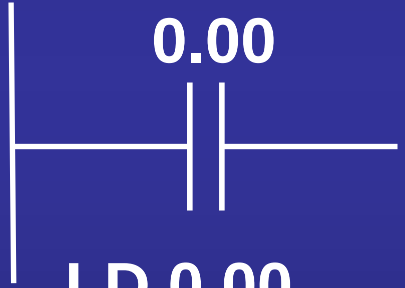
➢LD	LD NOT
➢AND	AND NOT
➢OR	OR NOT
➢OUT	OUT NOT
➢OR LD	AND LD
➢END	
➢NOP	
➢SET	RESET
➢KEEP	
➢DIFU	DIFD

3.2 Basic instructions

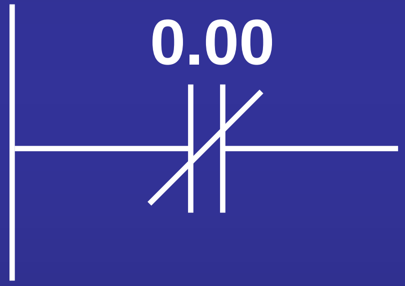
➢LD , LD NOT

✓LD—Start a rung with NO contact

✓LD NOT—Start a rung with NC contact



LD 0.00



LD NOT 0.00

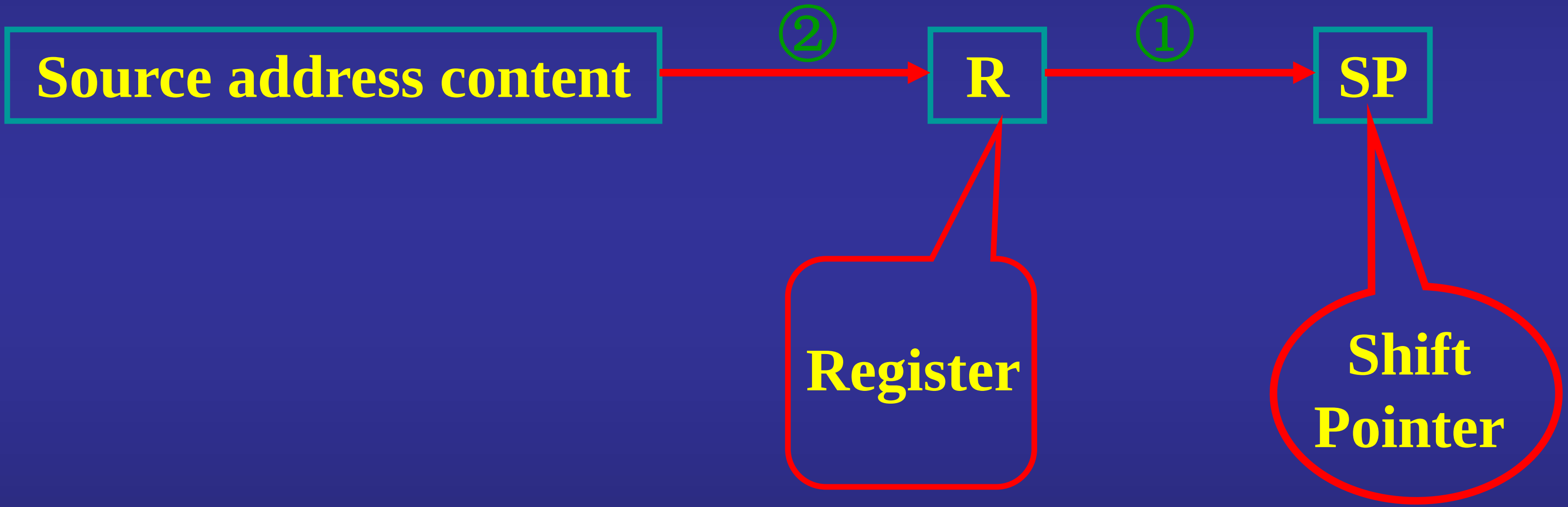
✓Operand Data Areas:

Input/output relay CIO, W, H, A, TR, T, C, TK, Condition Flag Area, Clock Pulse Area, IR

3.2 Basic instructions

➢LD , LD NOT

LD Instruction flow chart:

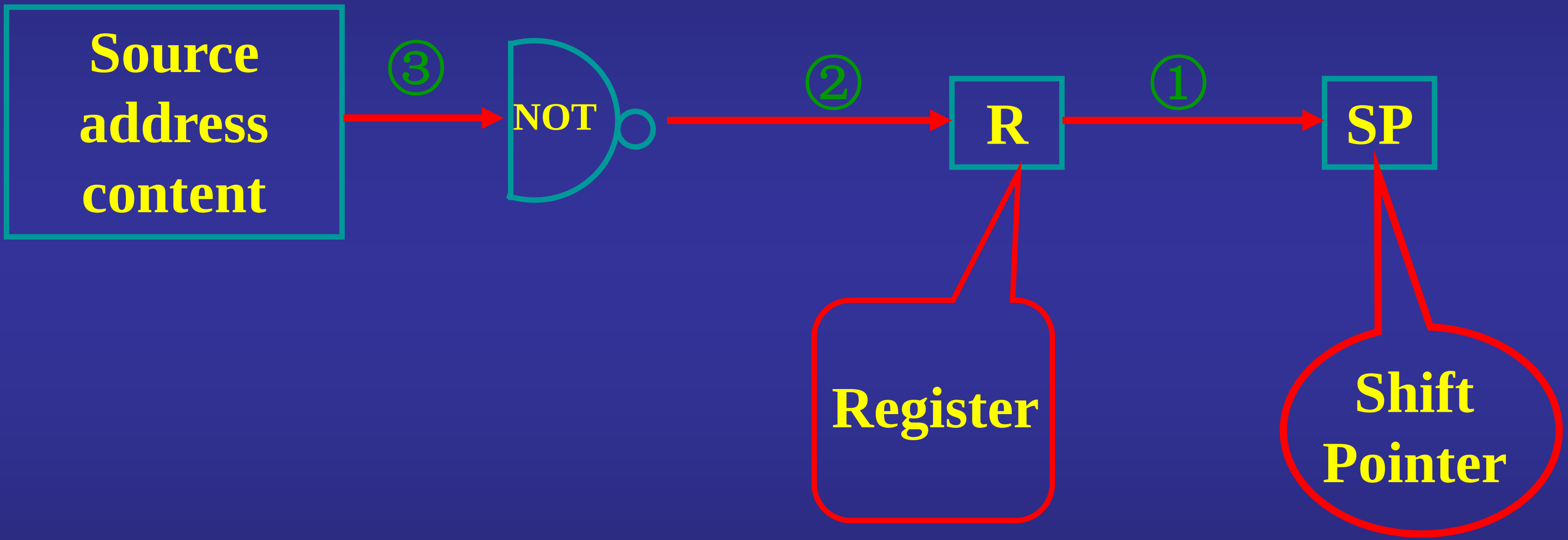


➢The source address specified by the LD operand remains unchanged.

3.2 Basic instructions

➢LD , LD NOT

LD Instruction flow chart:



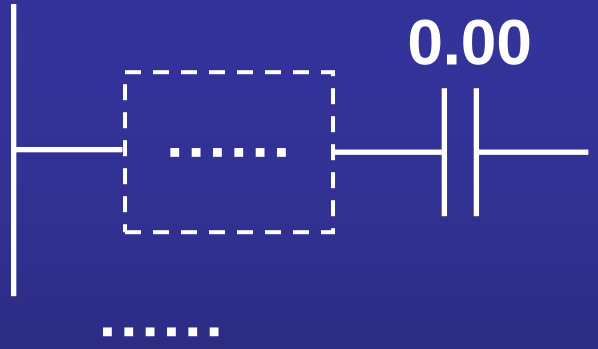
➢The source address specified by the LD NOT operand remains unchanged.

3.2 Basic instructions

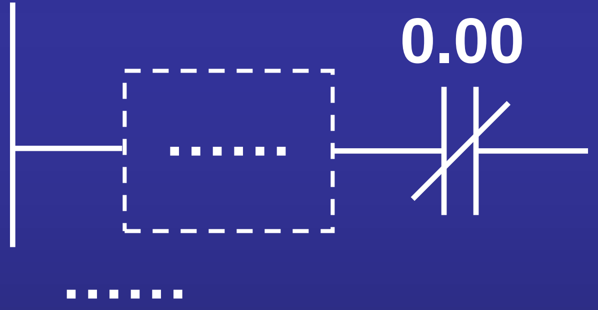
➢AND , AND NOT

✓AND— A series element with a NO contact

✓AND NOT— A series element with a NC contact



AND 0.00



AND NOT 0.00

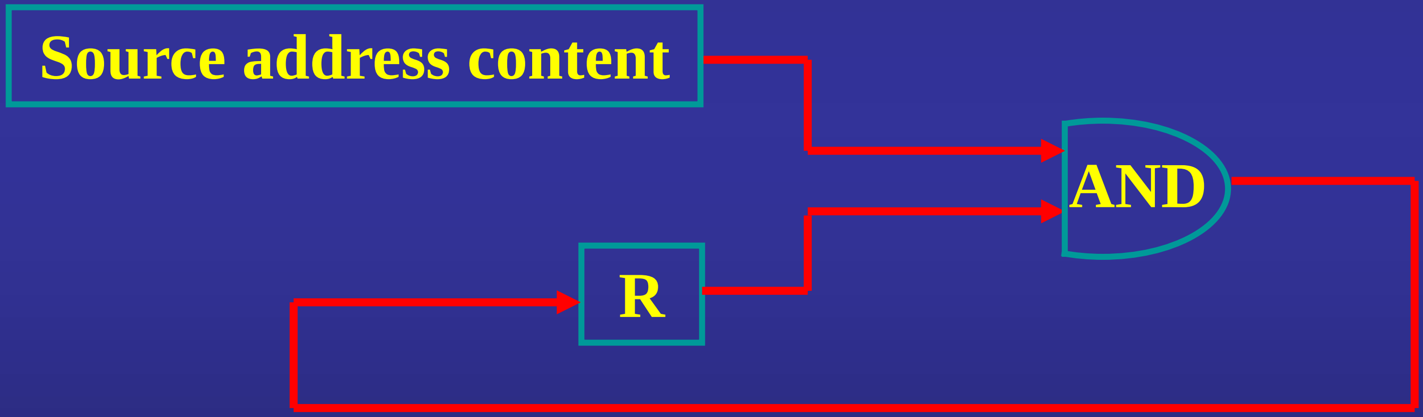
✓Operand Data Areas:

CIO, W, H, A, TR, T, C, TK, Condition Flag Area, Clock Pulse Area, IR

3.2 Basic instructions

- **AND , AND NOT**

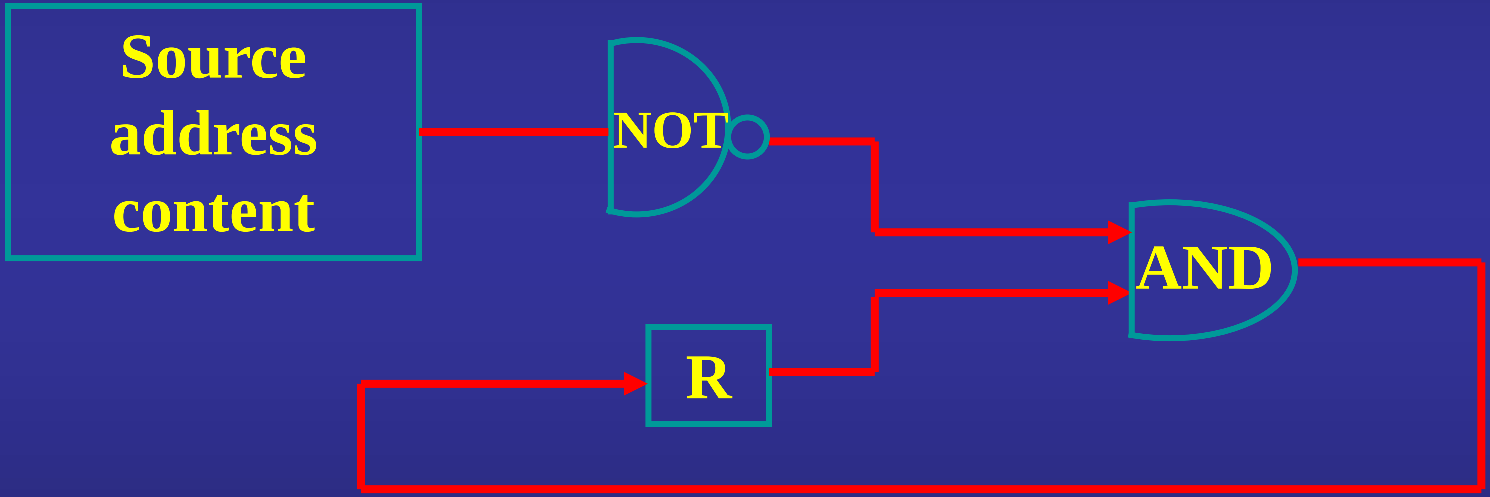
AND Instruction flow chart:



3.2 Basic instructions

- **AND , AND NOT**

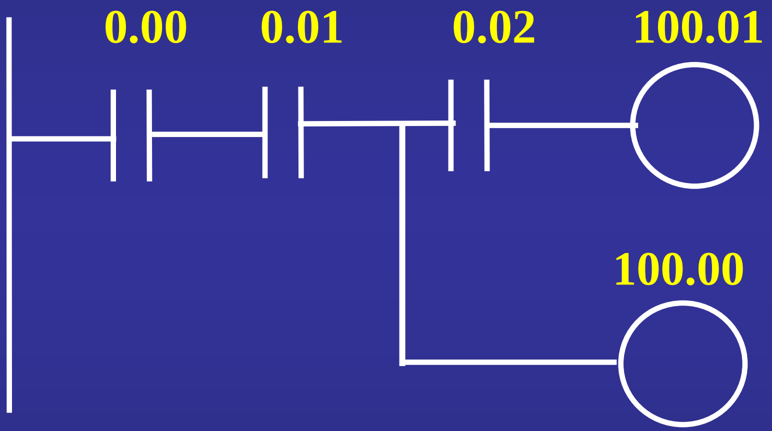
AND NOT Instruction flow chart:

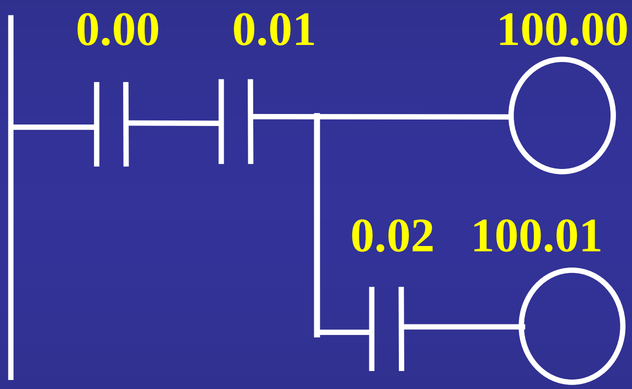


3.2 Basic instructions

➢AND , AND NOT

✓Continuous Output





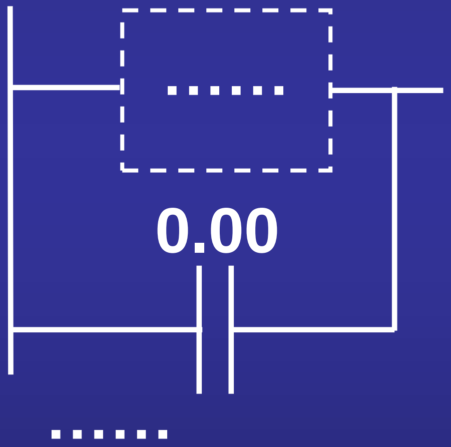
```
LD       0.00
AND      0.01
OUT      100.00
AND      0.02
OUT      100.01
```

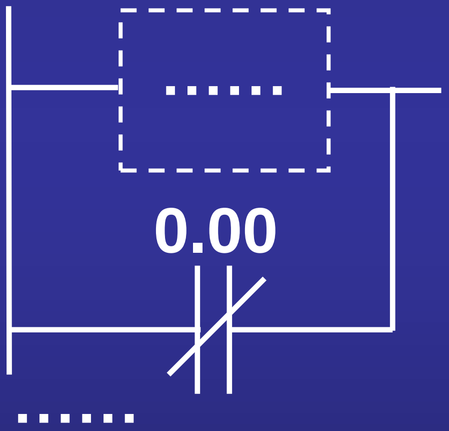
3.2 Basic instructions

➢OR , OR NOT

✓OR— a parallel element with NO contact.

✓OR NOT—Parallel element with NC contacts





OR 0.00 **OR NOT 0.00**

✓Operand Data Areas:

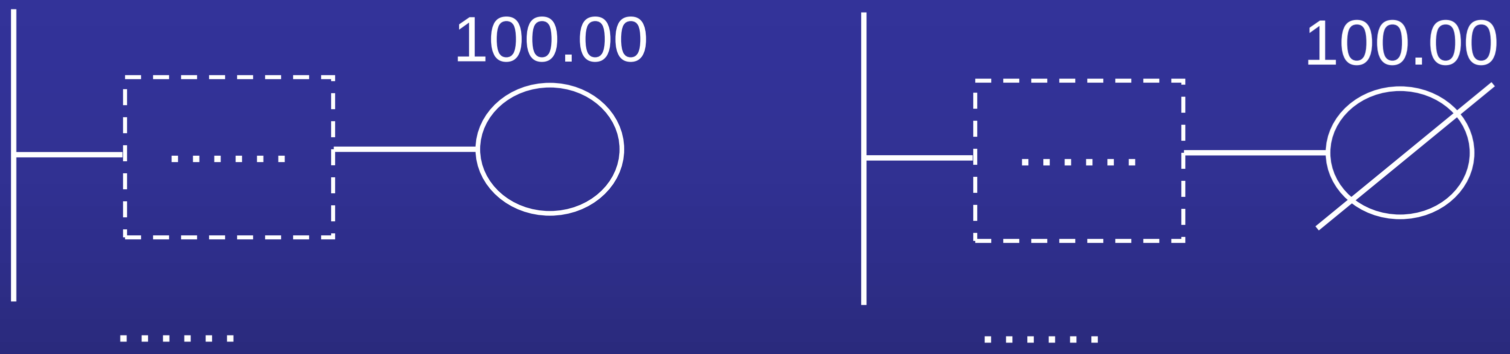
CIO, W, H, A, TR, T, C, TK, Condition Flag Area, Clock Pulse Area, IR

3.2 Basic instructions

➢OUT, OUT NOT

✓OUT—Output the execution condition to operand bit

✓OUT NOT—Output the converse of execution condition to operand bit



OUT 100.00

OUT NOT 100.00

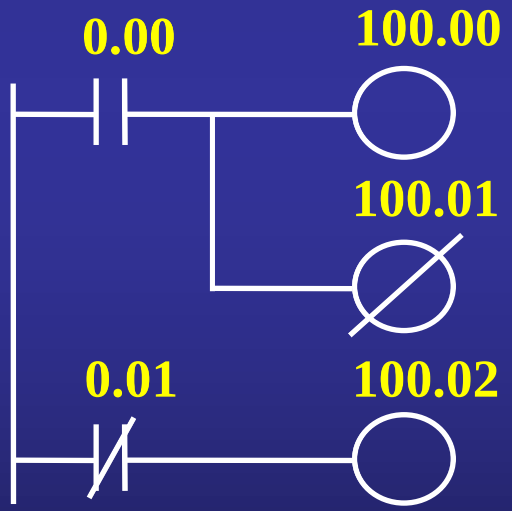
✓Operand Data Areas:

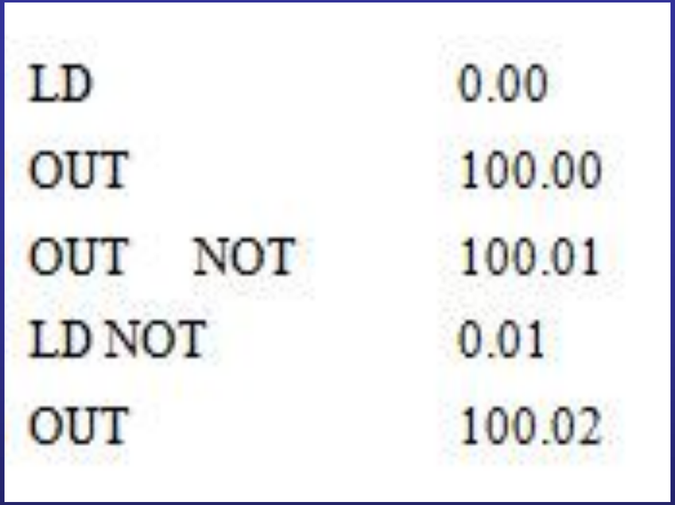
CIO, W, H, A, TR, IR

3.2 Basic instructions

➢OUT, OUT NOT

✓Parallel Coil Output





LD	0.00
OUT	100.00
OUT NOT	100.01
LD NOT	0.01
OUT	100.02

3.2 Basic instructions

➢OUT, OUT NOT

- LD LD NOT AND AND NOT
  OR OR NOT OUT OUT NOT

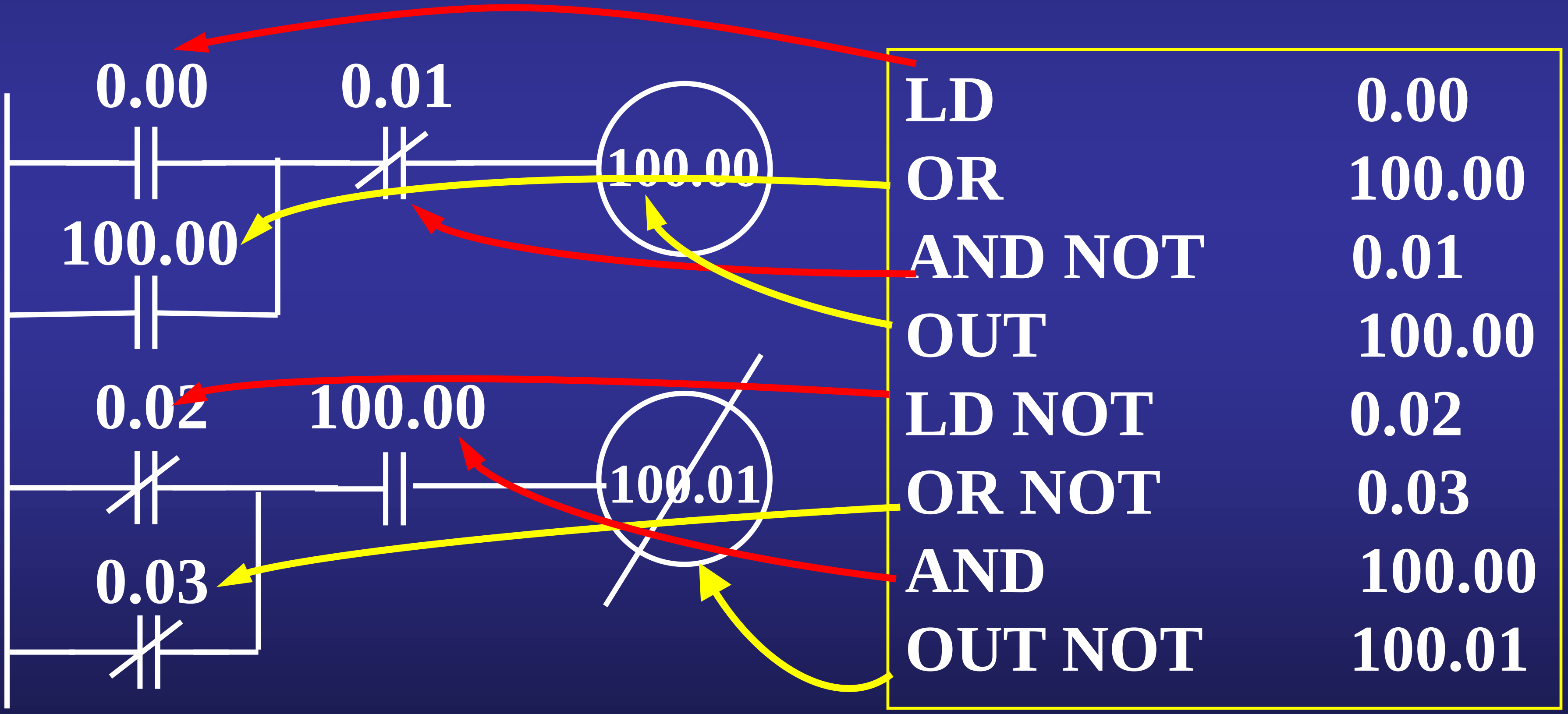
- LD: NO Contact and Left Power Rail
- AND: Series connection of NOs
- OR: Parallel connection of NOs
- OUT: Output the logic operation results

3.2 Basic instructions

Example

ladder diagram

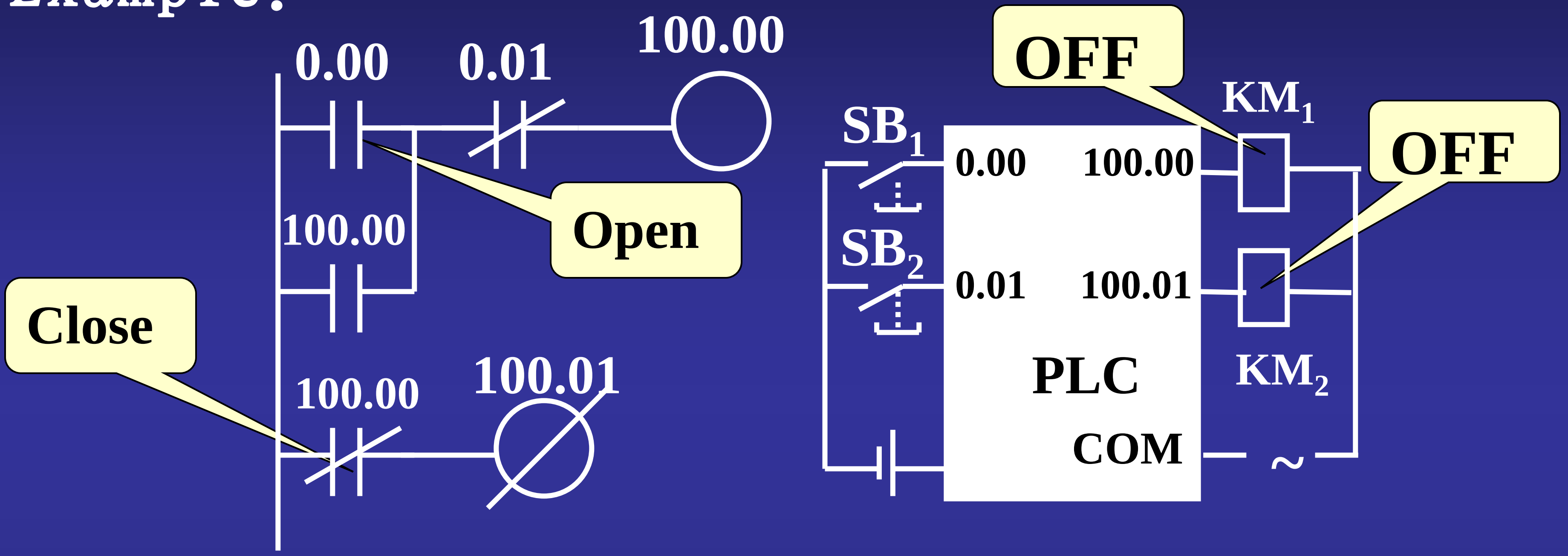
mnemonic program



```
LD            0.00
OR            100.00
AND NOT       0.01
OUT           100.00
LD NOT        0.02
OR NOT        0.03
AND           100.00
OUT NOT       100.01
```

3.2 Basic instructions

Example:

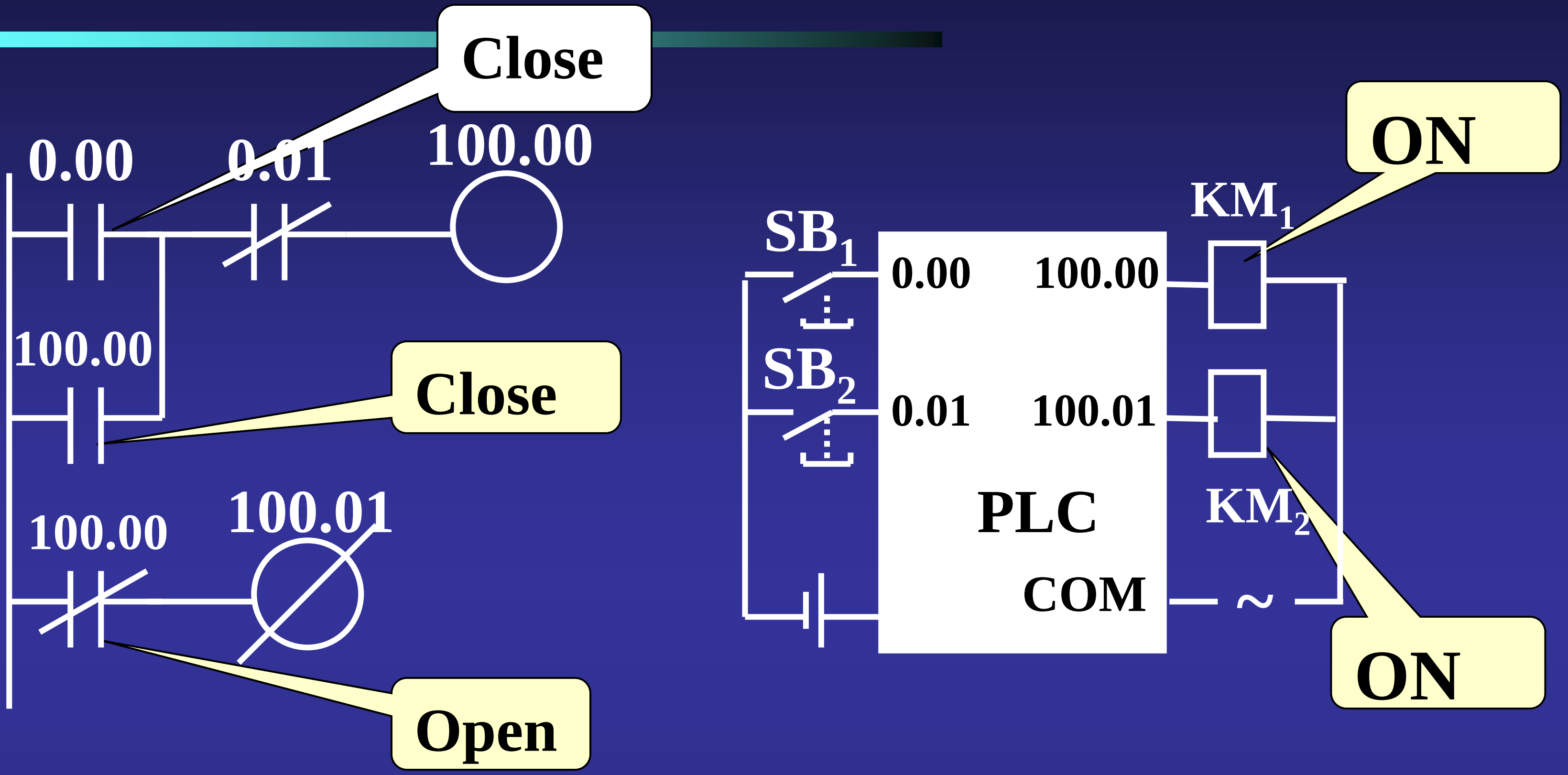


Release SB_1 → Coil 0.00 OFF→ Contact 0.00 open→

Contact 0.01 close → Coil 100.00 OFF→ KM_1 OFF

Contact 100.00 close→ Coil 100.01 OFF→ KM_2 OFF

3.2 Basic instructions

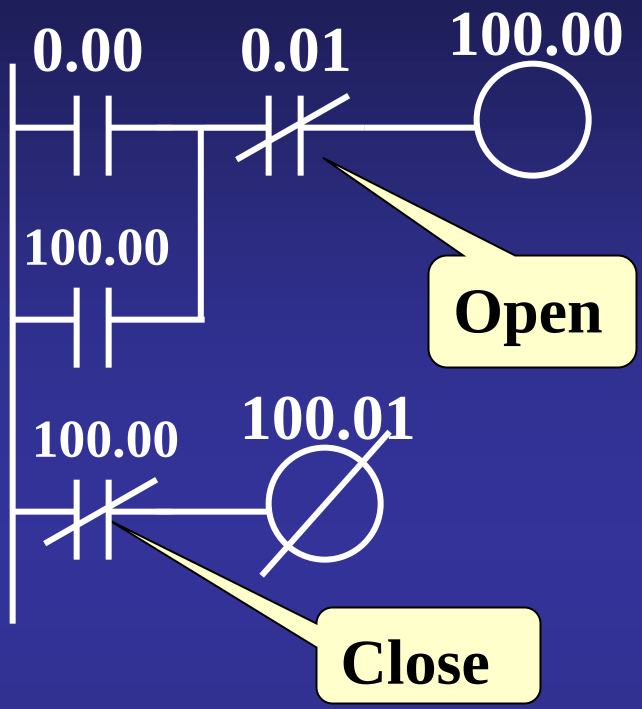


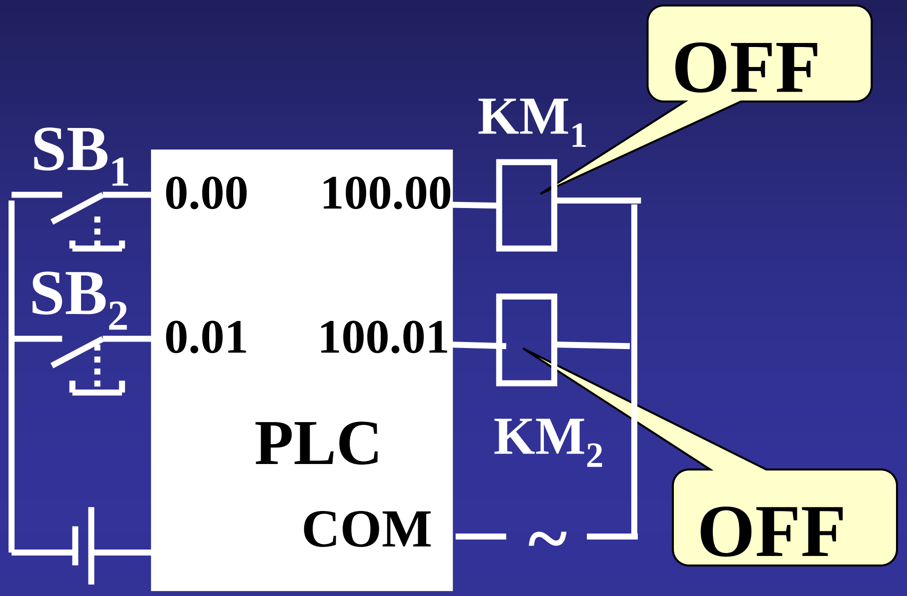
Press SB_1 → Coil 0.00 ON→ Contact 0.00 close→

Contact 0.01 close→ Coil 100.00 ON→ KM_1 ON

Contact 100.00 open→ Coil 100.01 ON→ KM_2 ON

3.2 Basic instructions





Press SB$_2$ → Coil 0.01 ON→ Contact 0.01 Open→

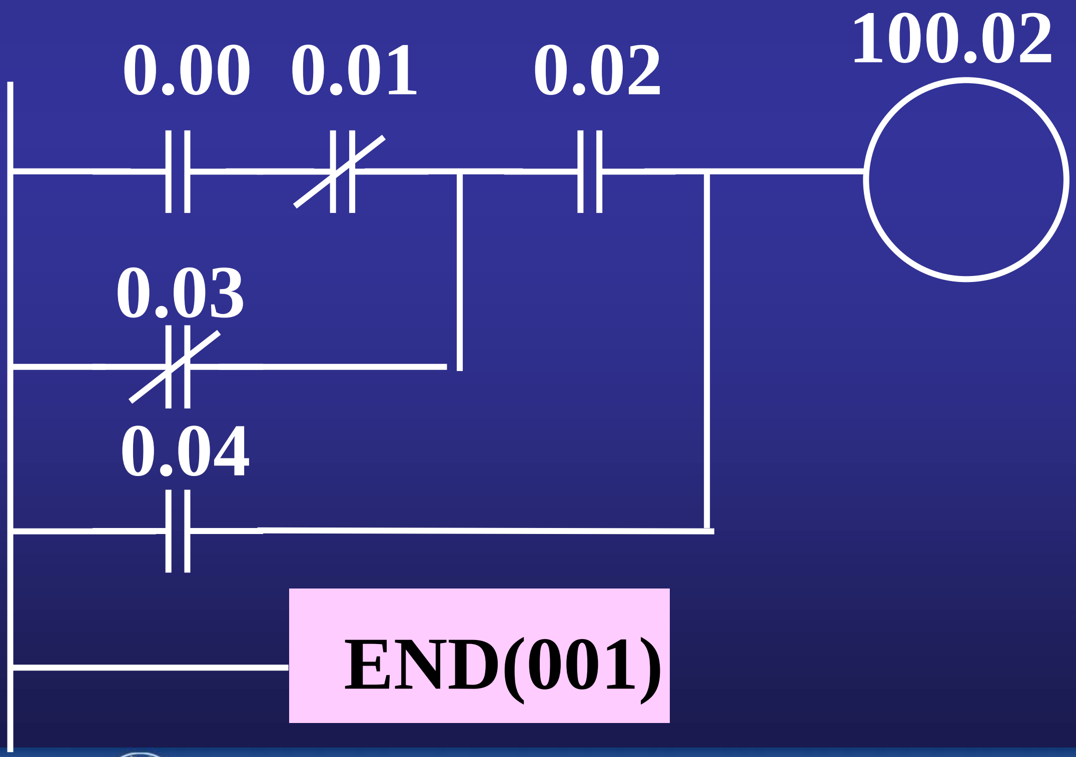
Coil 100.00 OFF→ KM$_1$ OFF

Contact 100.00 close→ Coil 100.01 OFF→ KM$_2$ OFF

3.2 Basic instructions

➢END

—The last instruction required to complete a simple program

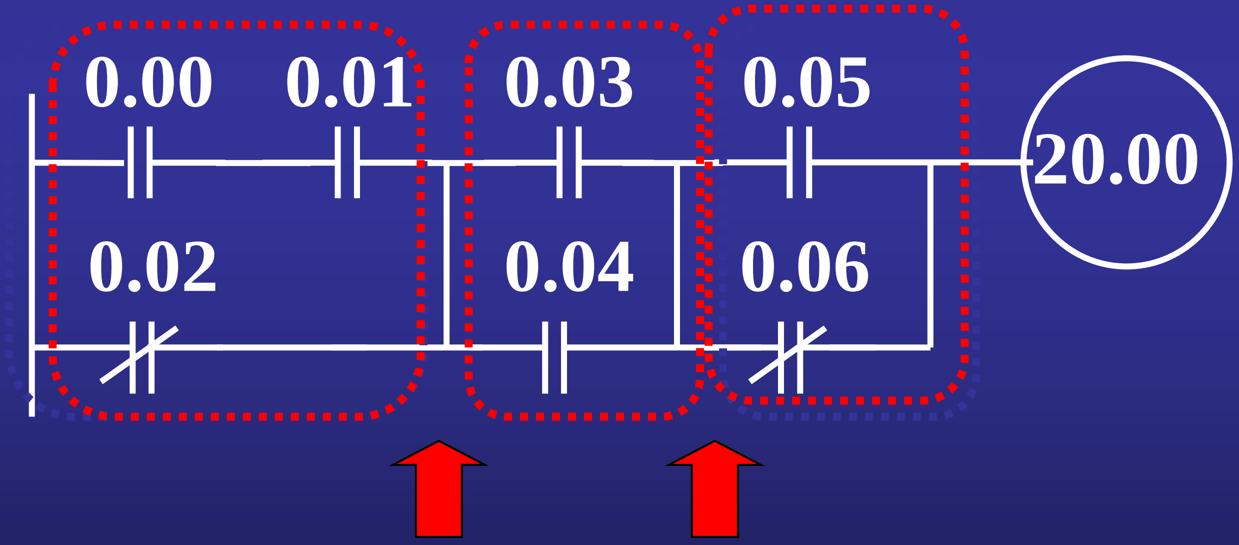


LD	0.00
AND NOT	0.01
OR NOT	0.03
AND	0.02
OR	0.04
OUT	100.02
END (001)	

3.2 Basic instructions

Logic block instructions

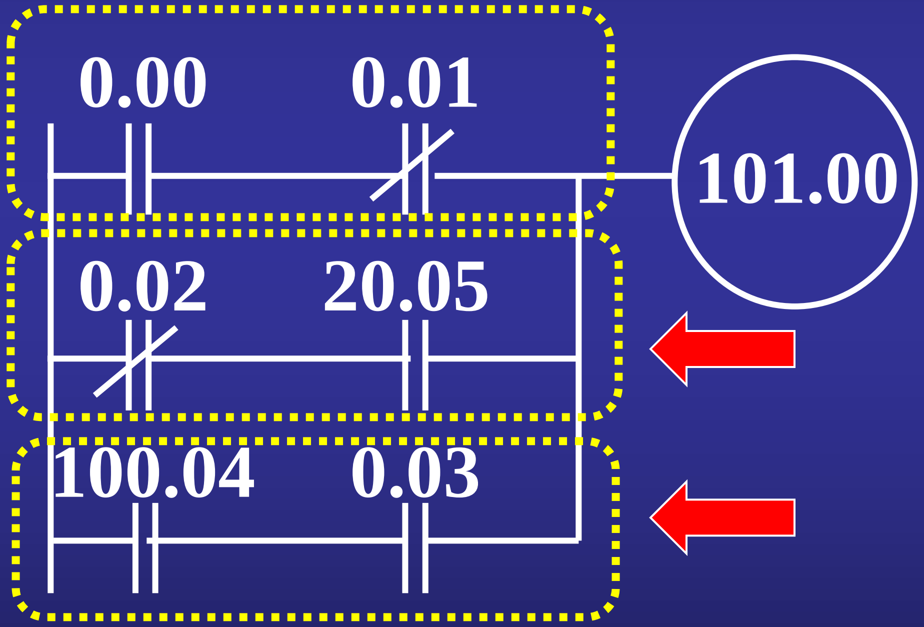
- ➢AND LD —— And the execution conditions produced by two logic blocks



```
LD          0.00
AND         0.01
OR NOT      0.02
LD          0.03
OR          0.04
AND LD
LD          0.05
OR NOT      0.06
AND LD
OUT        20.00
```

3.2 Basic instructions

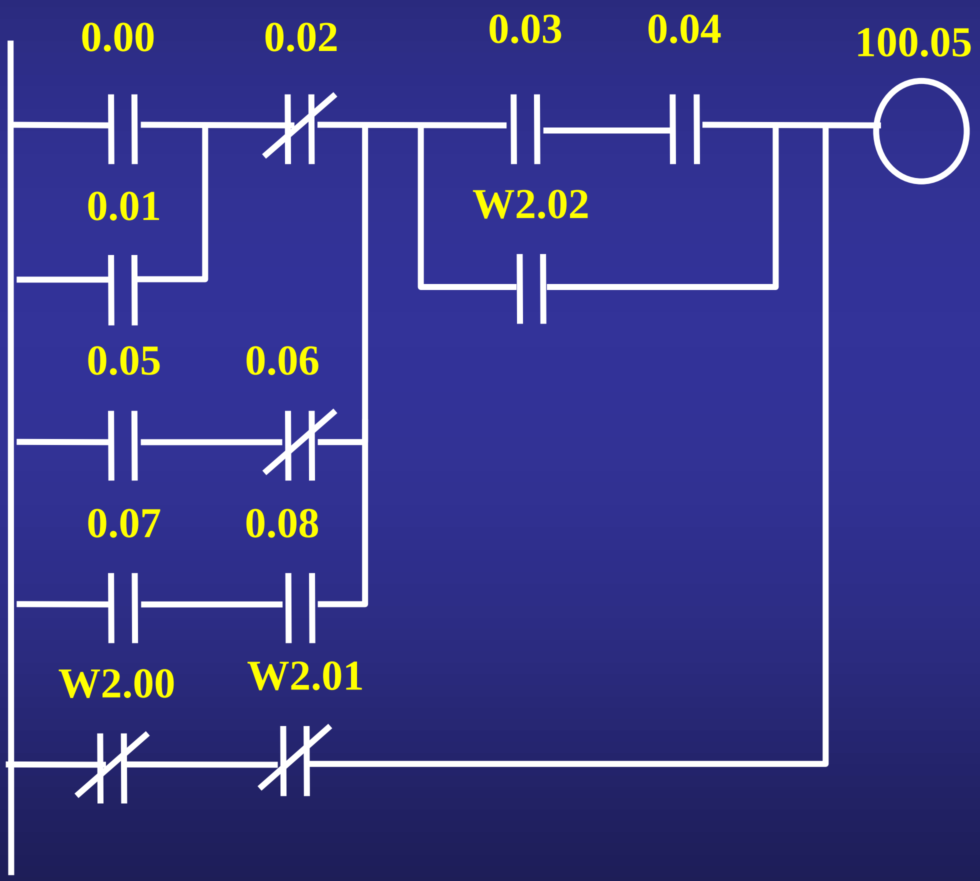
- **OR LD** ——And the execution conditions produced by two logic blocks



LD	0.00
AND NOT	0.01
LD NOT	0.02
AND	20.05
OR LD	
LD	100.04
AND	0.03
OR LD	
OUT	101.00

3.2 Basic instructions

➢Example: Ladder diagram → Instructions

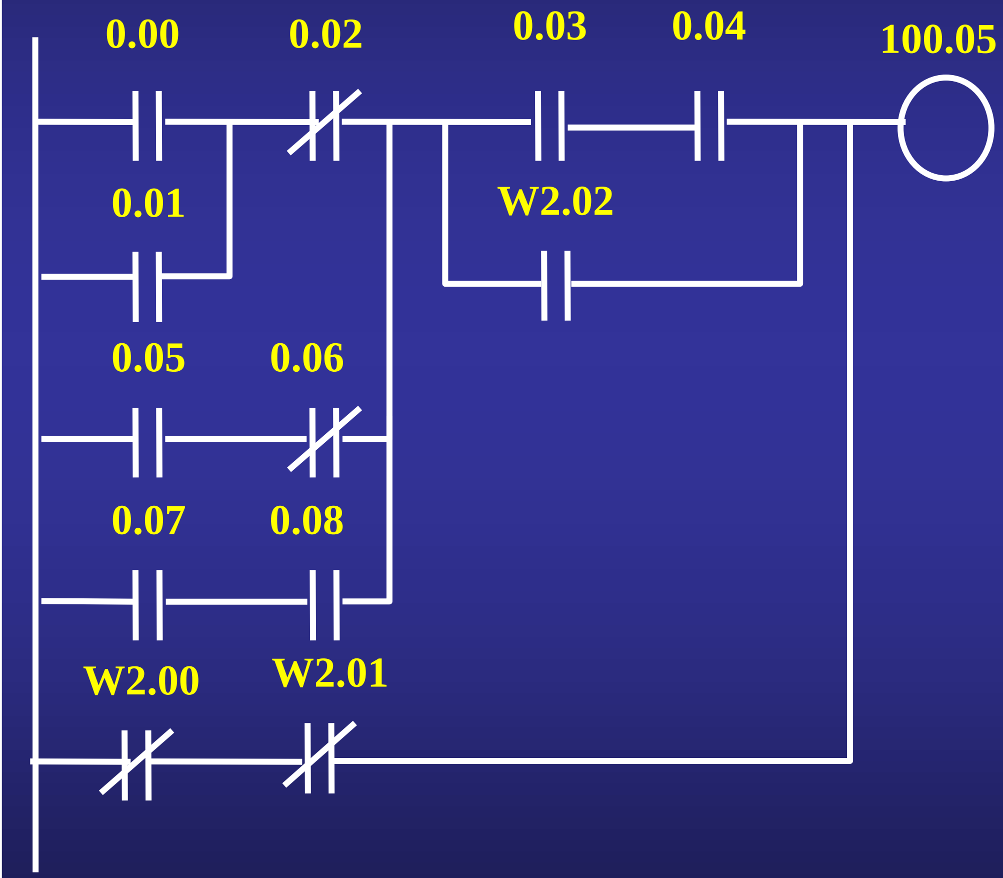


LD	0.00
OR	0.01
AND NOT	0.02
LD	0.05
AND NOT	0.06
OR LD	
LD	0.07
AND	0.08
OR LD	
LD	0.03
AND	0.04
OR	W2.02
AND LD	
LD NOT	W2.00
AND NOT	W2.01
OR LD	
OUT	100.05

3.2 Basic instructions

➢Example: Instructions → Ladder diagram

```
LD            0.00
OR            0.01
AND NOT       0.02
LD            0.05
AND NOT       0.06
OR LD
LD            0.07
AND           0.08
OR LD
LD            0.03
AND           0.04
OR            W2.02
AND LD
LD NOT        W2.00
AND NOT       W2.01
OR LD
OUT           100.05
```

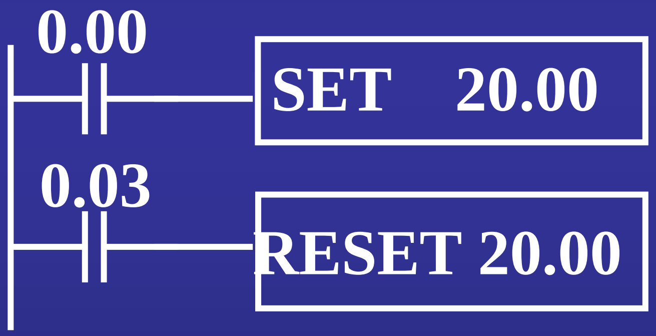


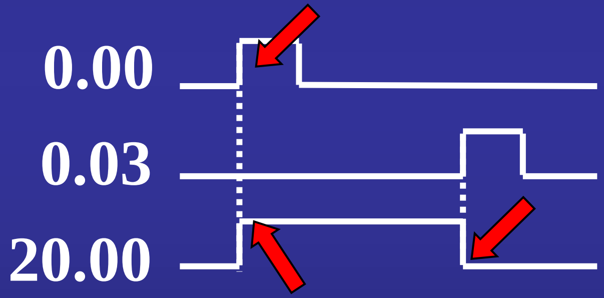
3.2 Basic instructions

3. 2. 3 SET, RESET

SET —— Turn bit ON when condition is ON

RESET —— Turn bit OFF when condition is ON





```
LD        0.00
SET       20.00
LD        0.03
RESET     20.00
```

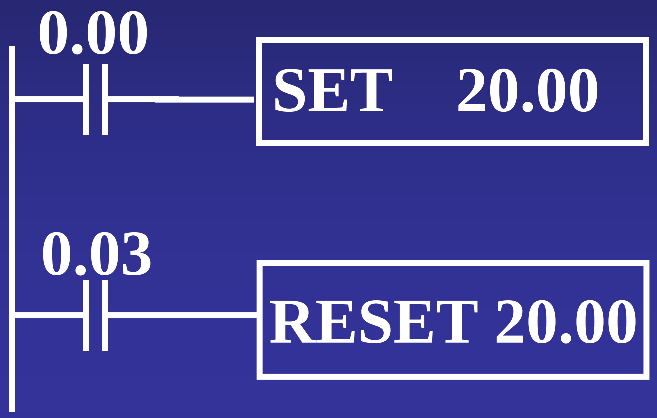
✓Different from OUTPUT and OUTPUT NOT

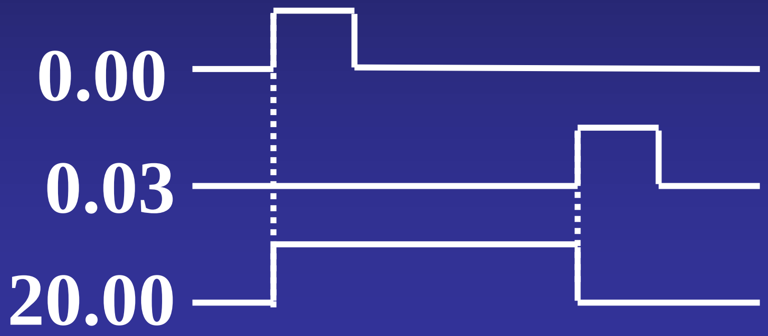
✓Execution condition is often short signal (pulse)

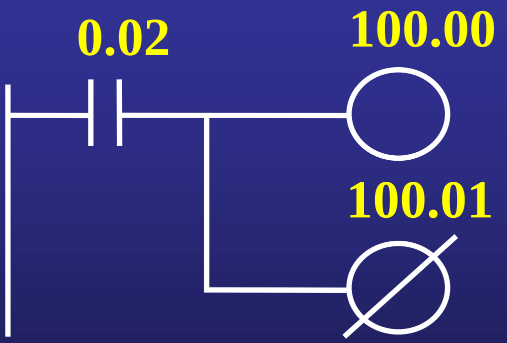
✓SET or RESET can be used separately

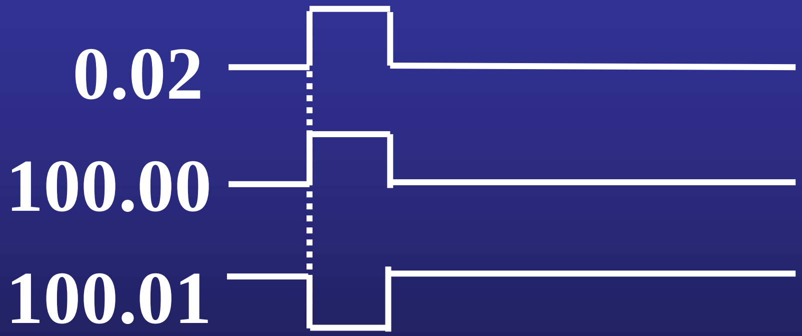
3.2 Basic instructions

3. 2. 3 SET， RESET







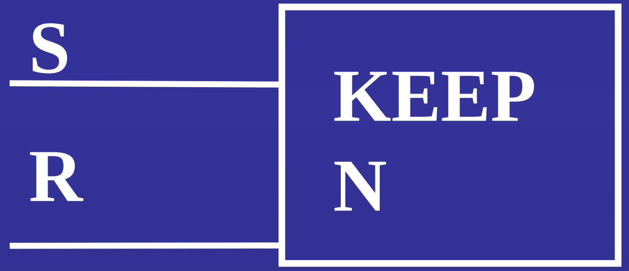


3.2 Basic instructions

3. 2. 4 KEEP

——used to maintain the status of the operand bit based on two execution conditions.

- **symbol**



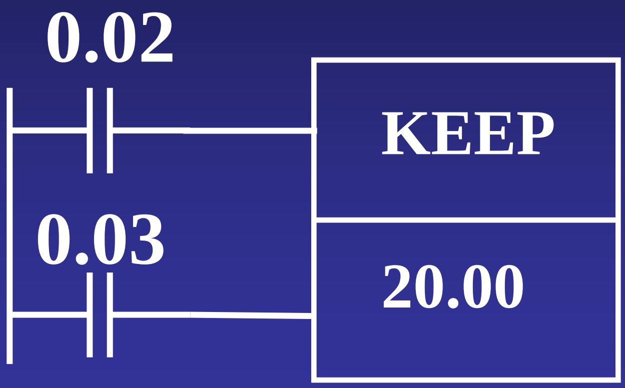
- **Function:**

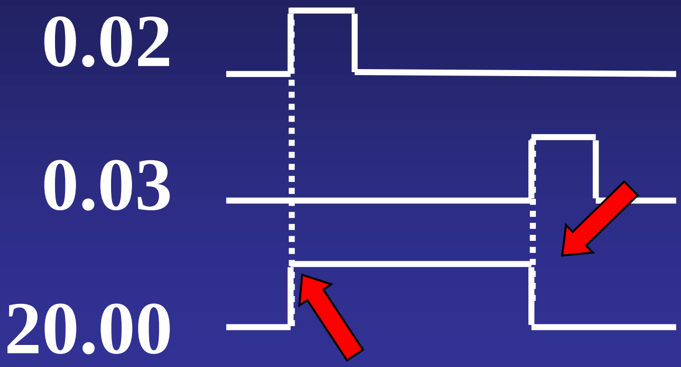
✓When S is ON， then N is ON until R is ON

✓When R is ON， N is OFF。

✓When R and S are ON, N is OFF

3.2 Basic instructions



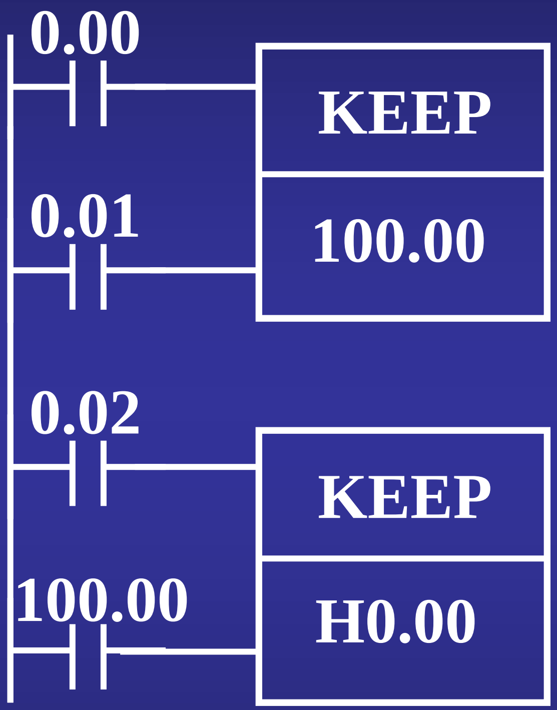


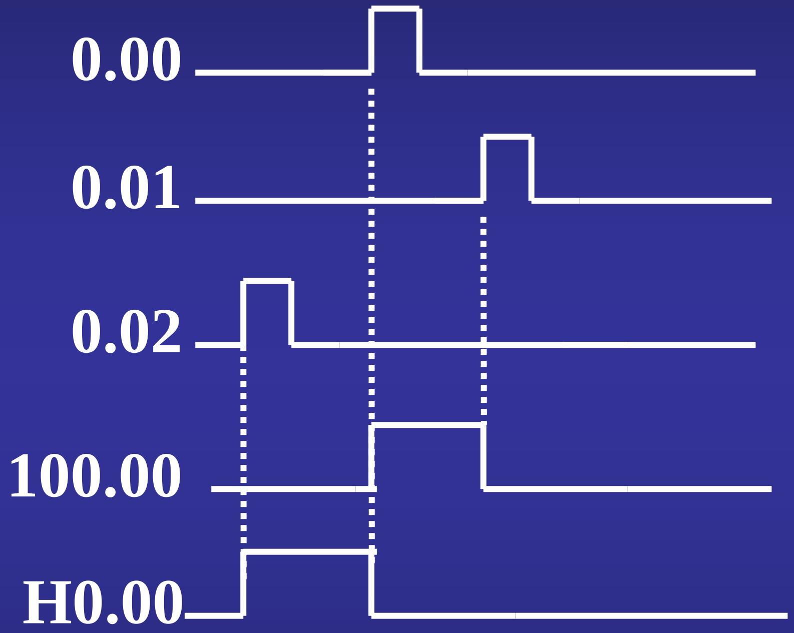
LD	0.02
LD	0.03
KEEP	20.00

Brief signal can be used as Set and reset input

3.2 Basic instructions

Example 3:

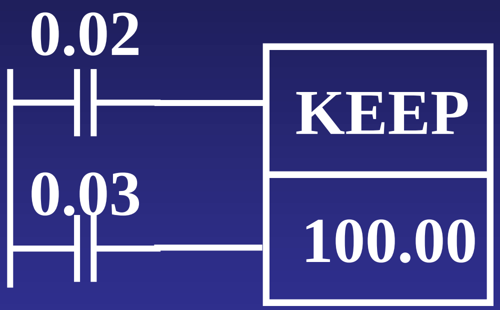




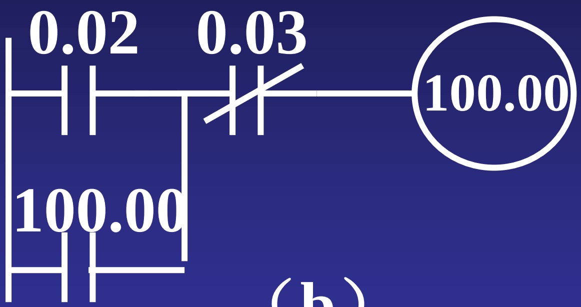
```
LD              0.00
LD              0.01
KEEP（011）     100.00
```

```
LD              0.02
LD              100.00
KEEP（011）     H0.00
```

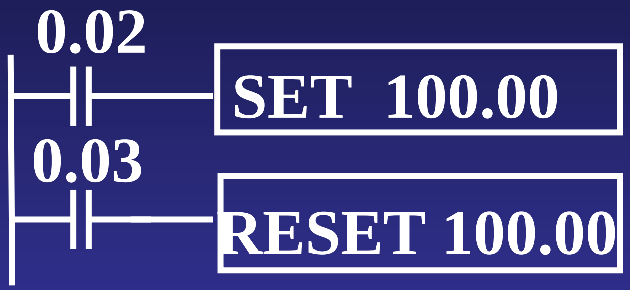
3.2 Basic instructions



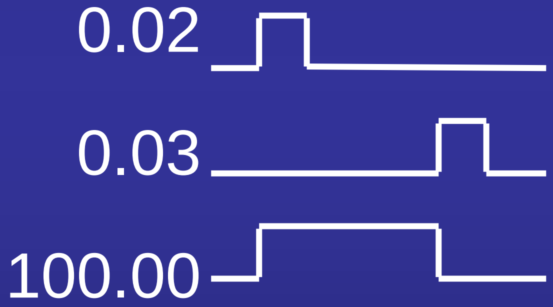
（a）



（b）



（c）

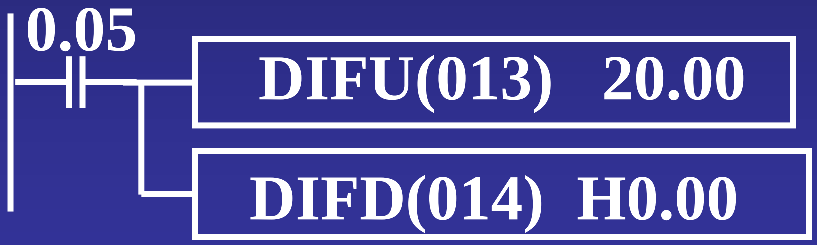


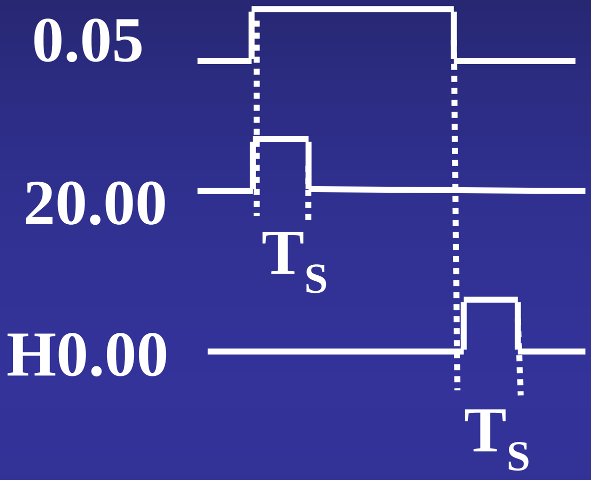
- Start-hold-stop function

Diff.:
- KEEP needs 3 instructions (the least)
- Other instructions can be inserted between SET and RESET

3.2 Basic instructions

3.2.5 DIFFERENTIATE UP and DOWN – DIFU(013) and DIFD(014)





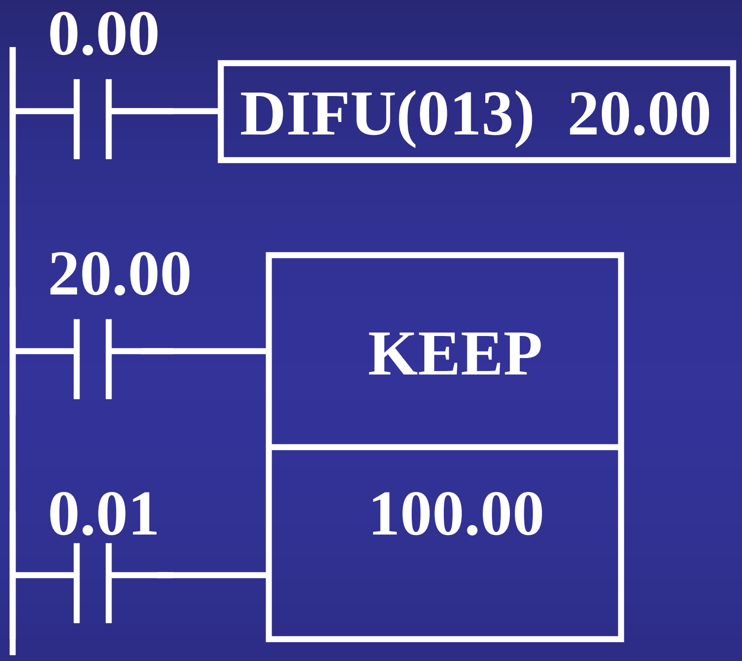
Used to turn the bit ON for one cycle only

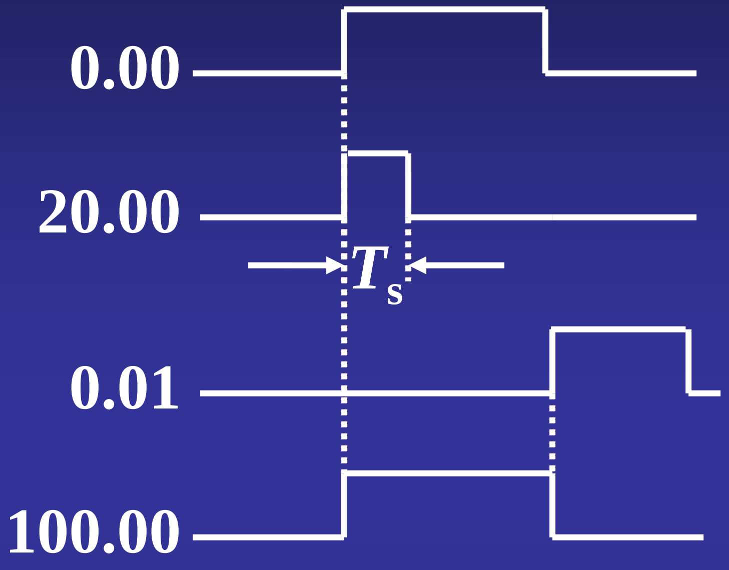
DIFU(013) : Rising edge **DIFD(014) :** Falling edge

- For those differentiated instructions (i.e., those prefixed with an @) are not available and single-cycle execution of a particular instruction is desired
- They can also be used with non-differentiated forms of instructions that have differentiated forms when their use will simplify programming

3.2 Basic instructions

Example:





LD	0.00
DIFU(013)	20.00
LD	20.00
LD	0.01
KEEP (011)	100.00

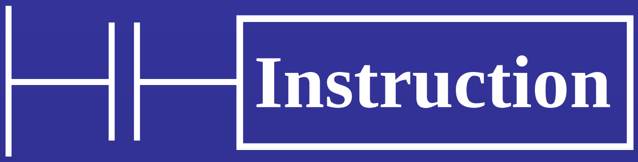
3.2 Basic instructions

3.2.6 Programming regulation

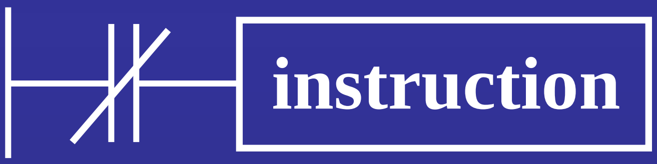
✓Each rung begins with left rail, and the coil is in the rightmost side.

✓Most instructions have execution condition

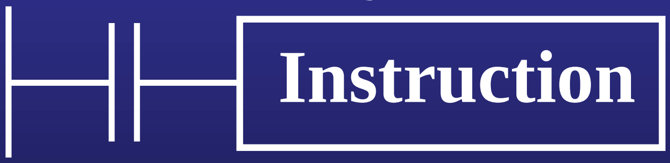
P_On



P_Off

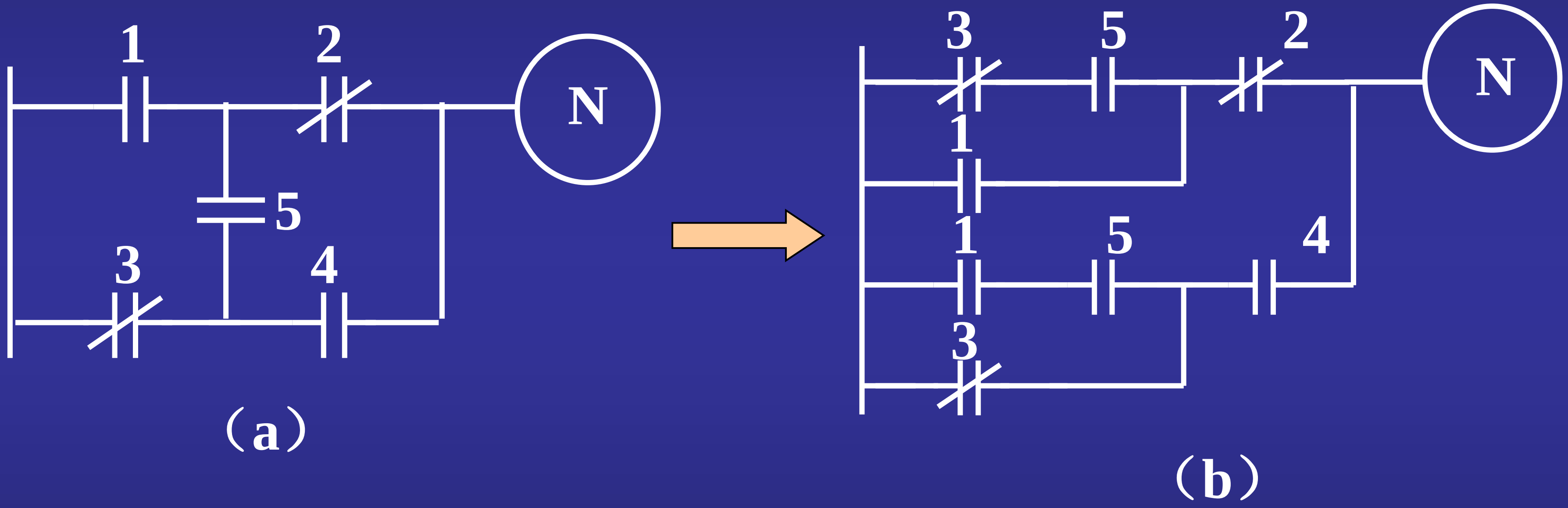


P_First_Cycle



3.2 Basic instructions

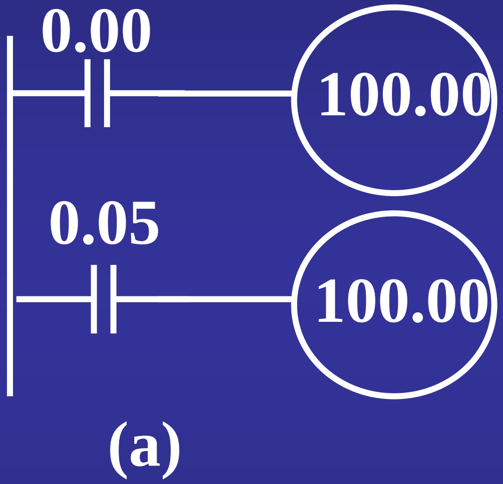
✓From top to bottom, from left to right. No conditions in vertical line.

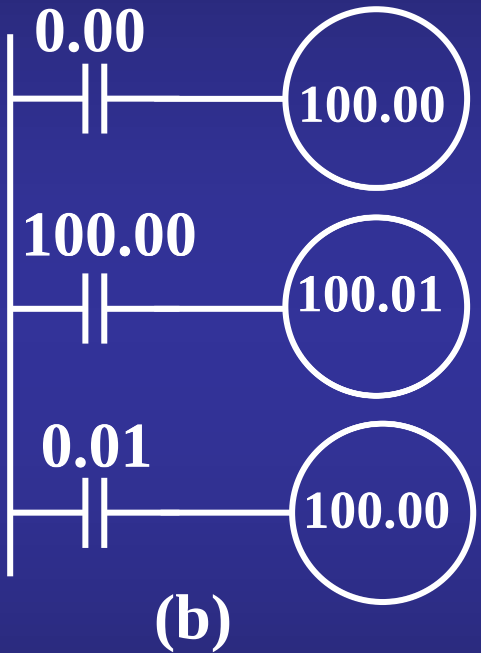


✓END instruction can not be omitted.

3.2 Basic instructions

✓Avoid duplicated output error—a common mistake

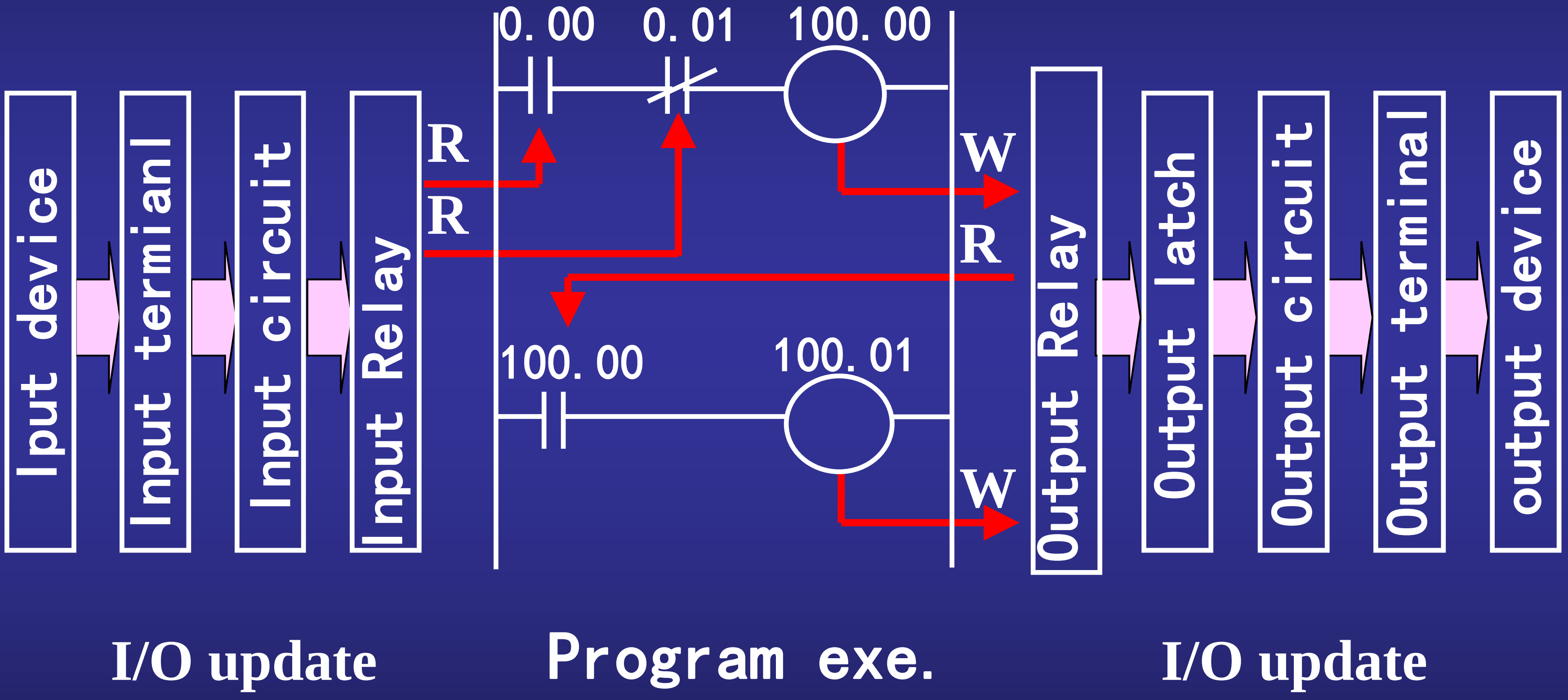




Duplicated Output error should be avoided

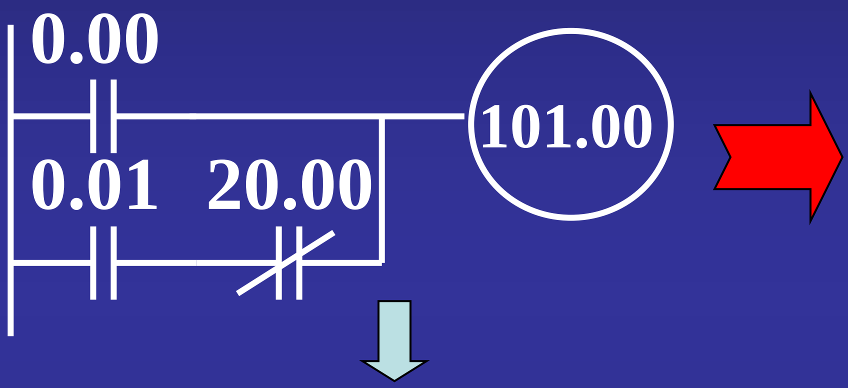
3.2 Basic instructions

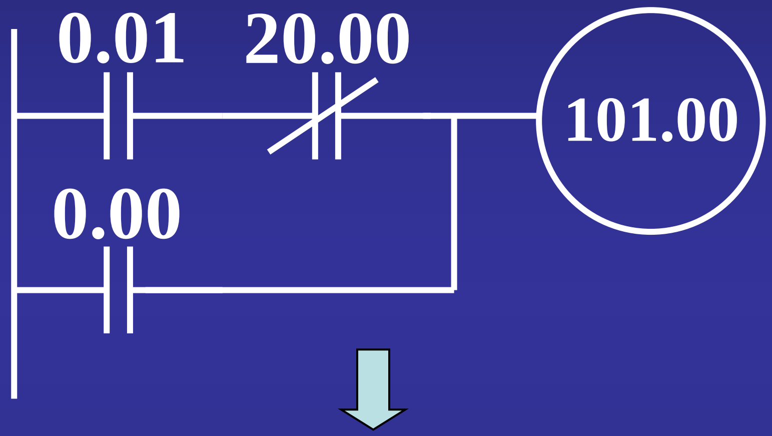
Signal transmission of PLC



3.2 Basic instructions

✓When block of series contacts is connected with a contact in parallel, the contact should be put below.



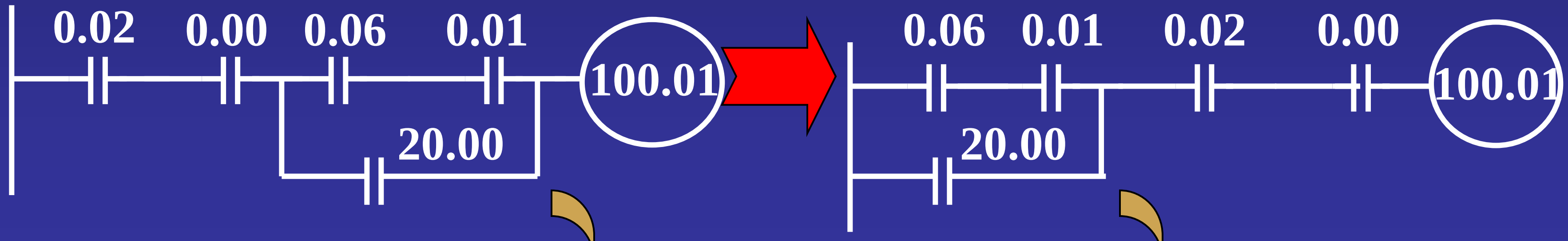


```
LD          0.00
LD          0.01
AND NOT     20.00
OR LD
OUT         101.00
```

```
LD          0.01
AND NOT     20.00
OR          0.00
OUT         101.00
```

3.2 Basic instructions

✓When block of parallel contacts is connected with contact in series, the block should be put leftmost.

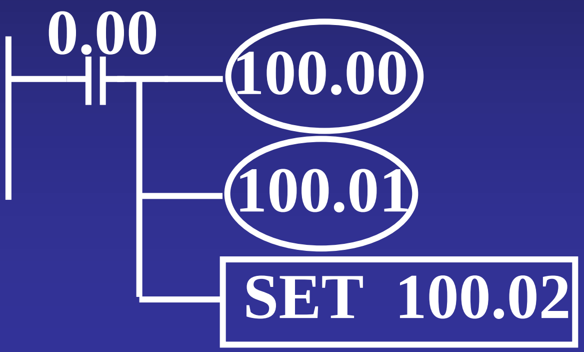


```
LD       0.02
AND      0.00
LD       0.06
AND      0.01
OR       20.00
AND  LD
OUT      100.01
```

```
LD       0.06
AND      0.01
OR       20.00
AND      0.02
AND      0.00
OUT      100.01
```

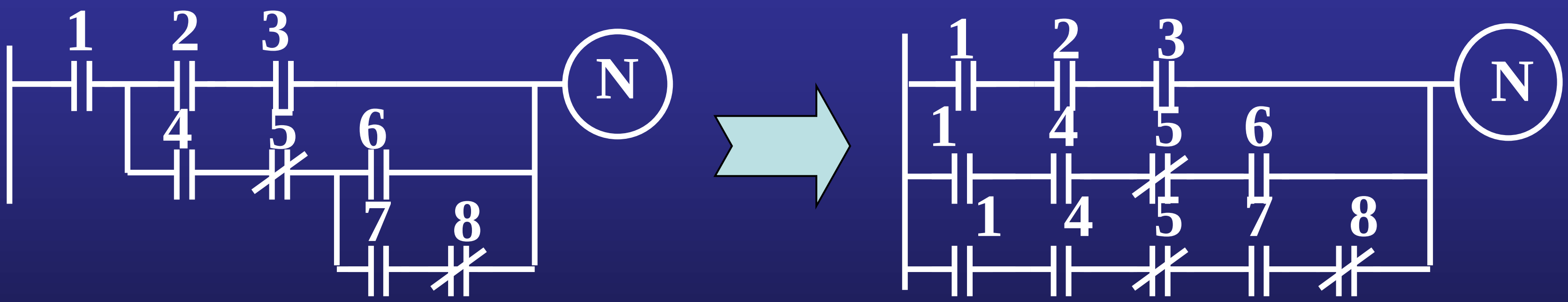
3.2 Basic instructions

✓Two or more coils can connected in parallel



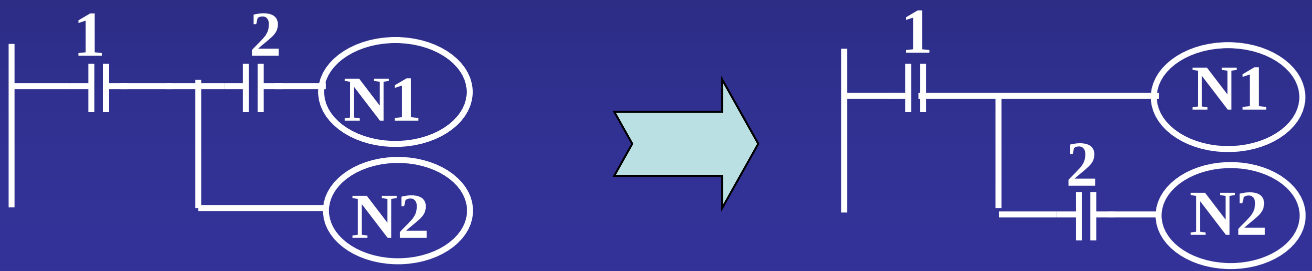
```
LD      0.00
OUT     100.00
OUT     100.01
SET     100.02
```

✓Complicated ladder diagram can be converted to mnemonic program after being arranged.



3.2 Basic instructions

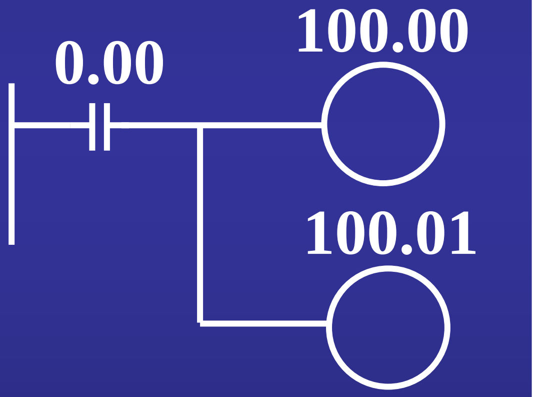
✓When a rung branches into two or more lines, the branch with only coil should be put above.



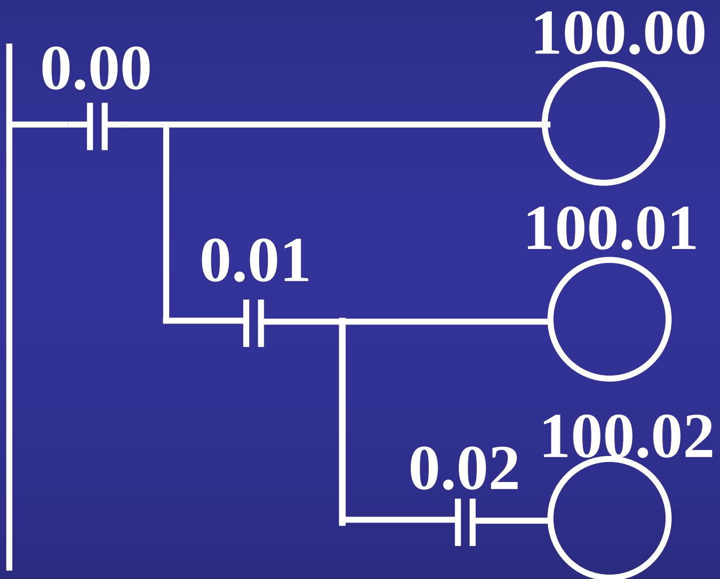
3.2 Basic instructions

✓Out form of diagram

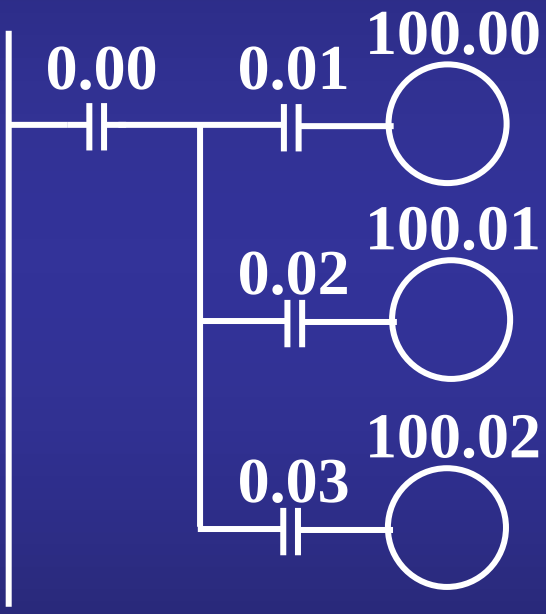
- Parallel output



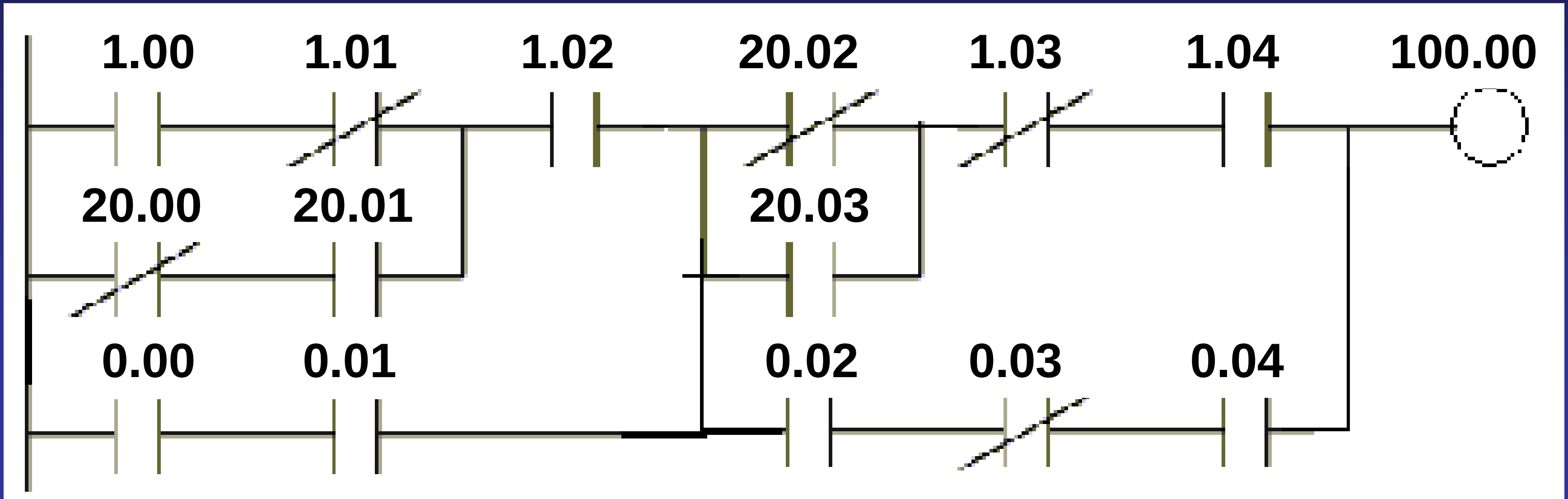
- Series output



- Compound output



3.2 Basic instructions



```
LD          1.00
AND NOT     1.01
LD NOT      20.00
AND         20.01
OR LD
AND         1.02

LD          0.00
AND         0.01
OR LD
LD NOT      20.02
OR          20.03
AND NOT     1.03

AND         1.04
LD          0.02
ANDNOT      0.03
AND         0.04
OR LD
AND LD

OUT     100.00
```

3.3 Application instructions

- Temporary relay
- IL/ILC
- JUMP
- TIMER/COUNTER

3.3 Application instructions

3.3.1Temprorary Relay—TR

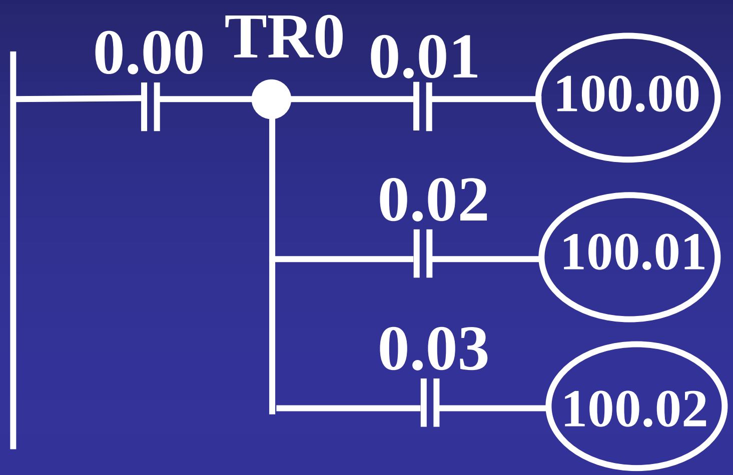
——used to temporarily preserve execution conditions

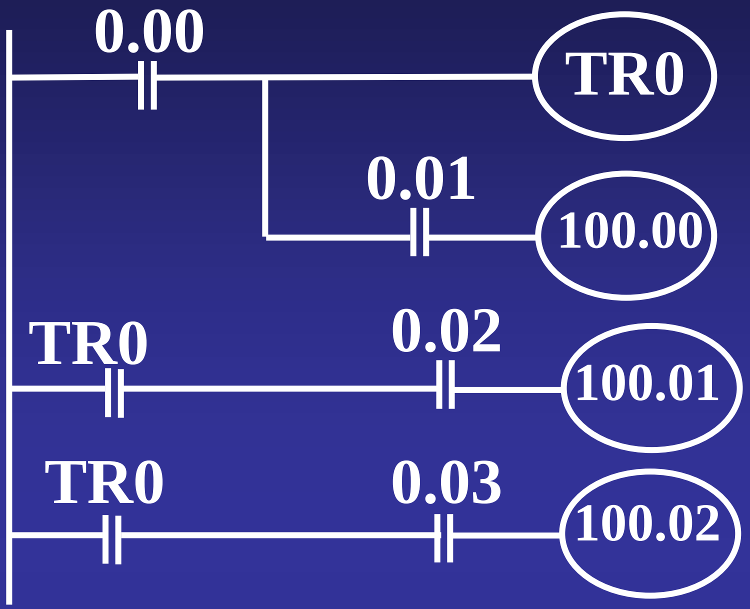
- ✓16 bits: TR00 ~ TR15
- ✓TR is not an instruction
- ✓Can be used to deal with the branching of ladder diagram
- ✓TR number cannot be used repeatedly in the same branching program.

3.3 Application instructions

例:

Compound output

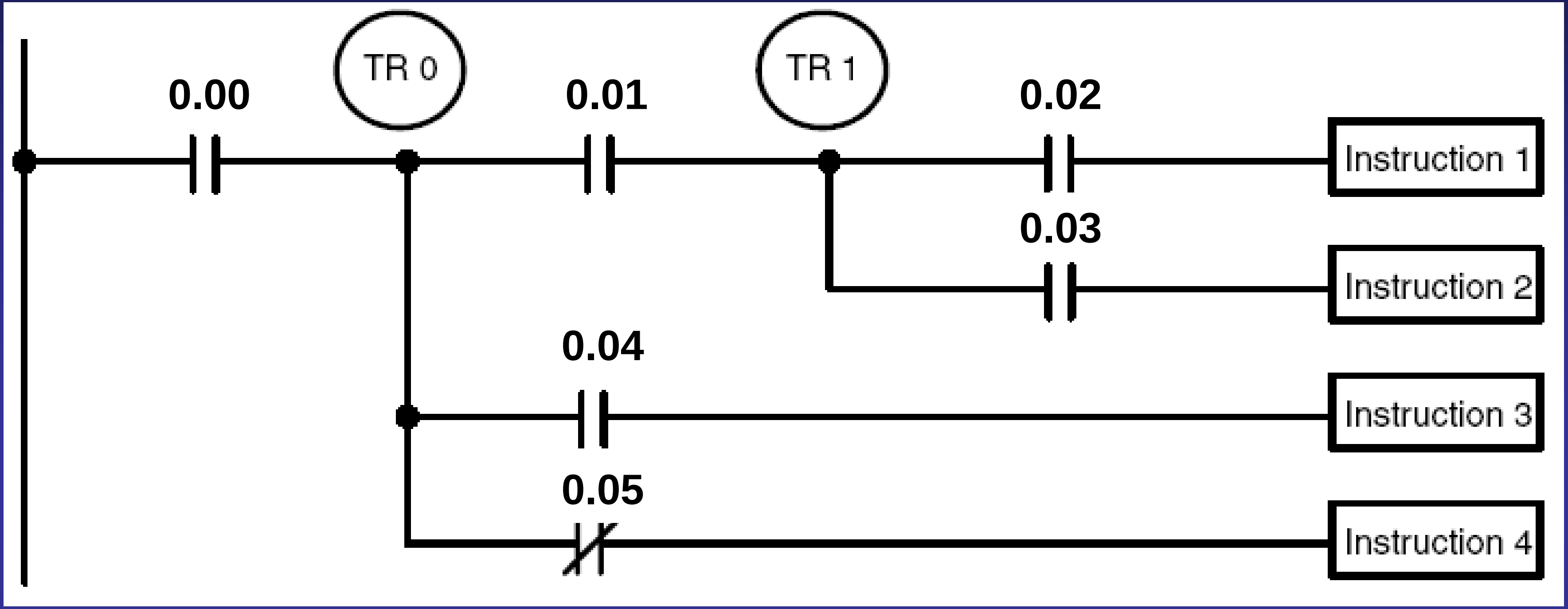




```
LD      0.00
OUT     TR0
AND     0.01
OUT     100.00
LD      TR0
```

```
AND     0.02
OUT     100.01
LD      TR0
AND     0.03
OUT     100.02
```

3.3 Application instructions



only for mnemonic code
ladder diagrams does not need

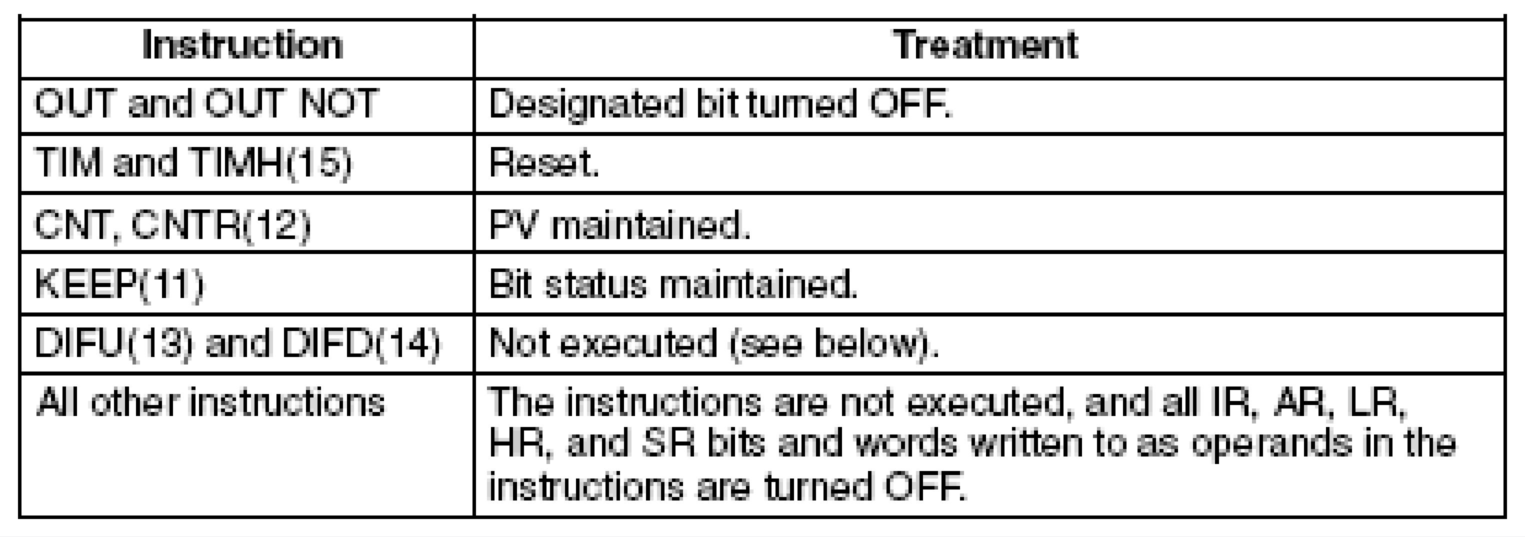
3.3 Application instructions

3.3.2 IL/ILC(Interlock and Interlock clear)

✓usage:

•When execution condition is ON, the program between IL and ILC will be executed.

•When execution condition is OFF, the interlocked section between IL and ILC will be treated as:



Instruction	Treatment
OUT and OUT NOT	Designated bit turned OFF.
TIM and TIMH(15)	Reset.
CNT, CNTR(12)	PV maintained.
KEEP(11)	Bit status maintained.
DIFU(13) and DIFD(14)	Not executed (see below).
All other instructions	The instructions are not executed, and all IR, AR, LR, HR, and SR bits and words written to as operands in the instructions are turned OFF.

3.3 Application instructions

- IL(02) and ILC(03) do not necessarily have to be used in pairs.
- There must be an ILC(03) following any one or more IL(02).
- Nesting is not supported for IL and ILC

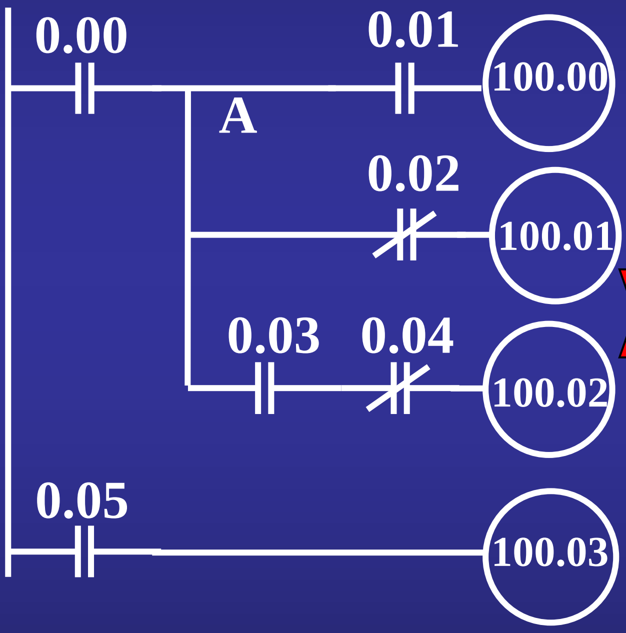
IL/ILC vs. TR

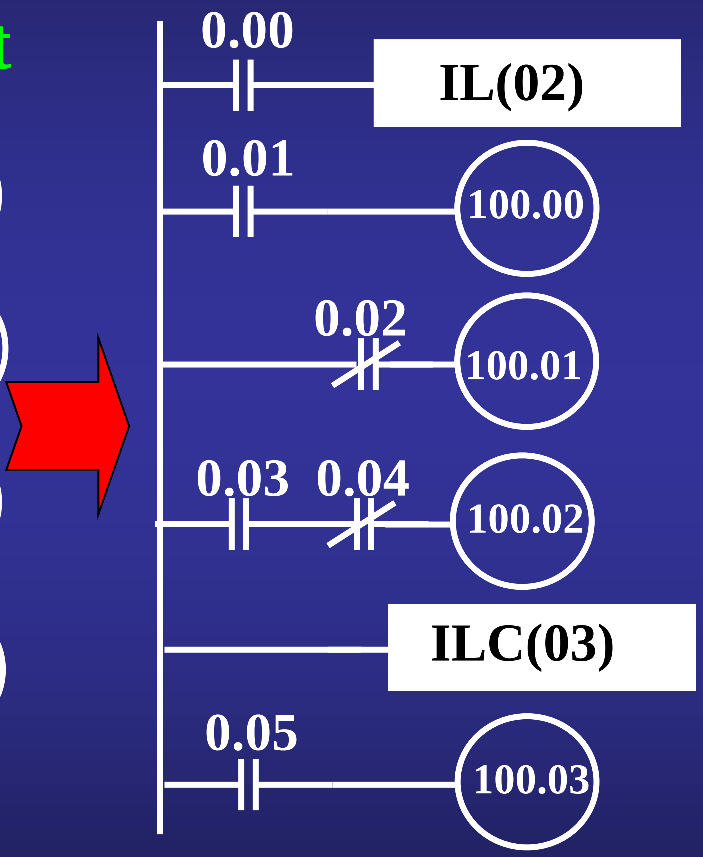
- **Similarity:** Interlocks are used to enable branching in the same way as can be achieved with TR bits
- **Difference:** Treatment of instructions between IL(02) and ILC(03) differs from that with TR bits when the execution condition for IL(02) is OFF

3.3 Application instructions

Example:

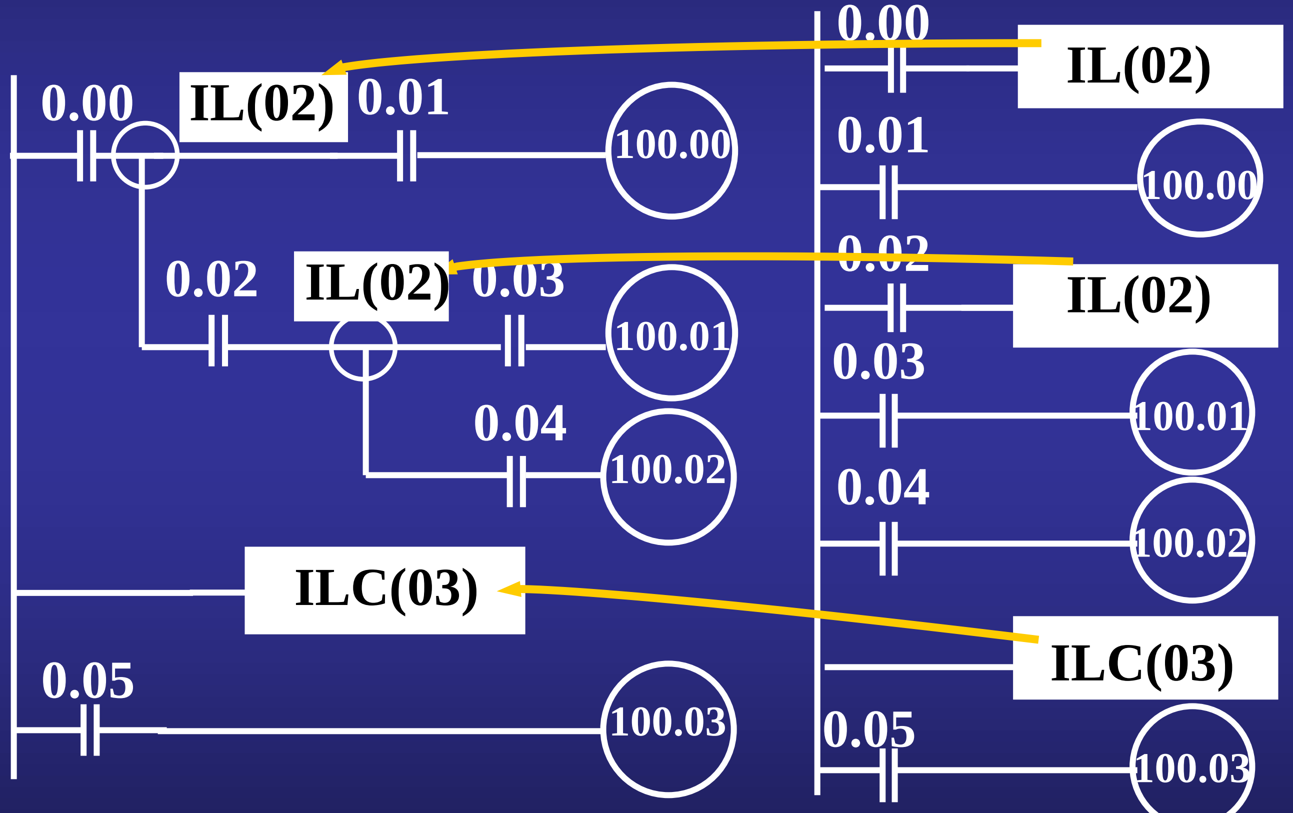
Compound output





LD	0.00
IL (02)	
LD	0.01
OUT	100.00
LD NOT	0.02
OUT	100.01
LD	0.03
AND NOT	0.04
OUT	100.02
ILC (03)	
LD	0.05
OUT	100.03

3.3 Application instructions



```
LD        0.00
IL (02)
LD        0.01
OUT     100.00
LD        0.02
IL (02)
LD        0.03
OUT     100.01
LD        0.04
OUT     100.02
ILC (03)
LD        0.05
OUT     100.03
```

3.3.3 JMP/JME

——control the execution of program branching

➢Usage

- ✓ If the execution condition for a JUMP instruction is ON, the program is executed normally as if the jump did not exist.
- ✓ If the execution condition for the JUMP instruction is OFF, program execution moves immediately to a JME instruction without changing the status of anything between the JUMP and JUMP END instruction.
- ✓ All JUMP and JUMP END instructions are assigned jump numbers ranging between 00 and 49.

3.3 Application instructions

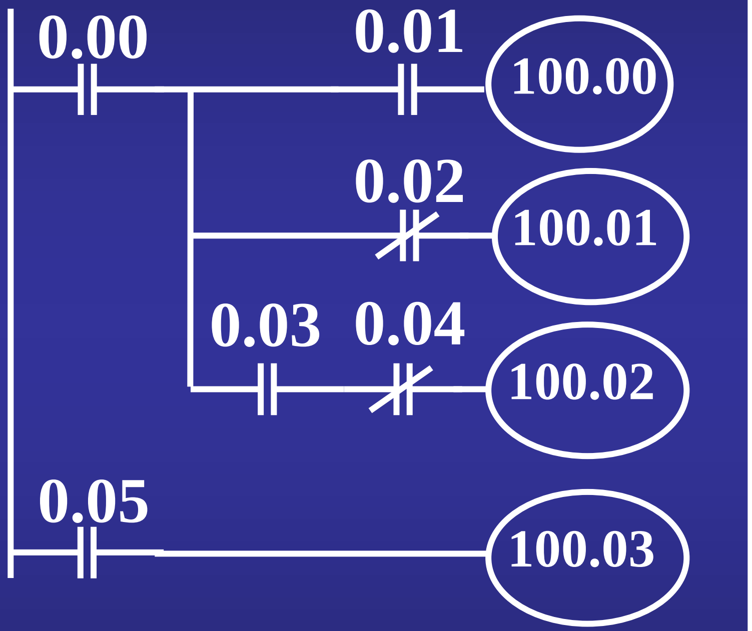
- ✓ A jump can be defined using jump numbers 01 through 49 only once, i.e., each of these numbers can be used once in a JUMP instruction and once in a JUMP END instruction.
- ✓Jump number 00 can be used as many times as desired.
- ✓ Nesting is supported with different numbers

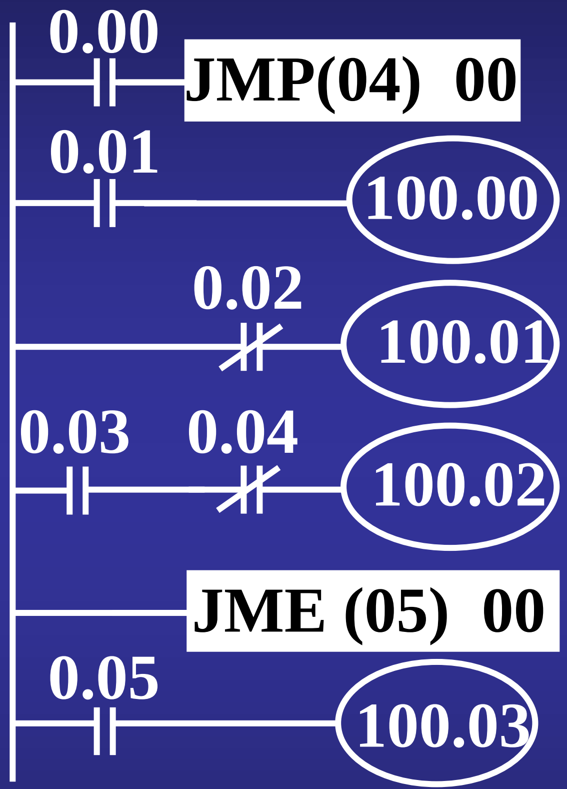
Example:
JMP 00
JMP 01
JME 01
JME 00

3.3 Application instructions

Example:

Compound output





```
LD          0.00
JMP (04)      00
LD          0.01
OUT       100.00
LD NOT      0.02
OUT       100.01
LD          0.03
AND NOT     0.04
OUT       100.02
JME (05)      00
LD          0.05
OUT       100.03
```

3.3 Application instructions

- 0.00 ON、0.01 OFF:

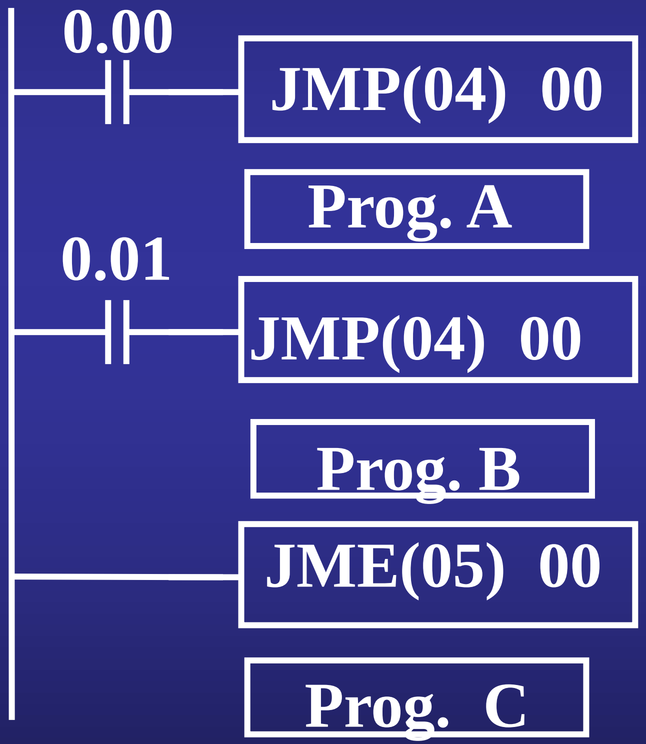
  Execute A→C

- 0.00 ON、0.01 ON:

  Execute A→B →C

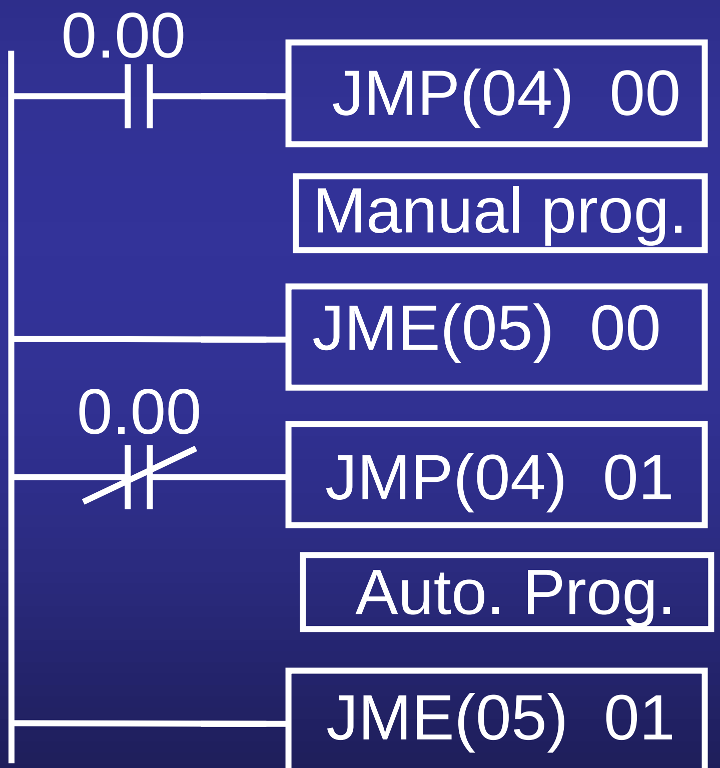
- 0.00 OFF、0.01 OFF:

  Execute C



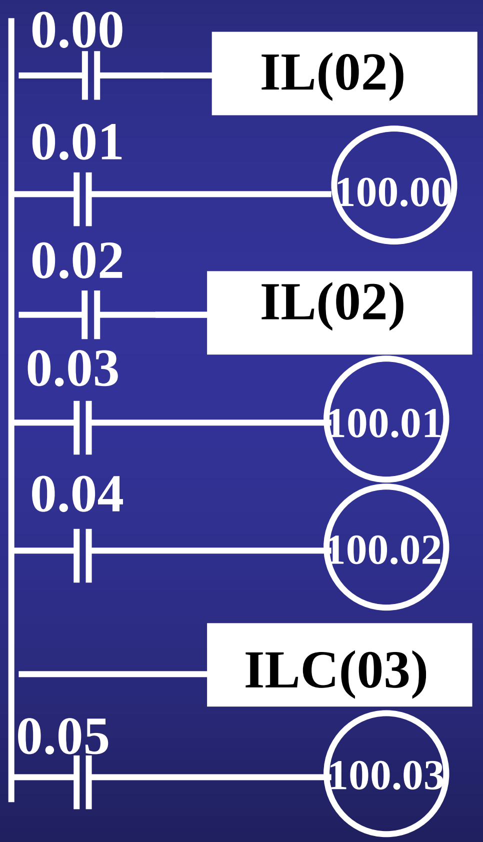
3.3 Application instructions

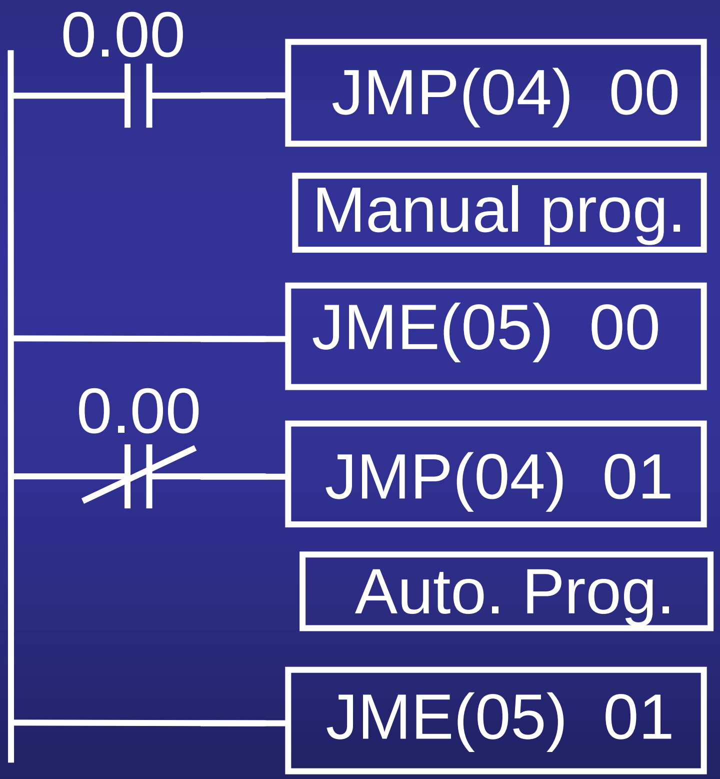
Example: used to switch between manual program and automatic program



3.3 Application instructions

IL/ILC vs JMP/JME





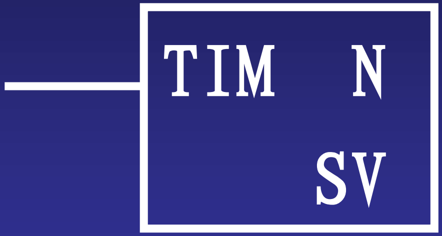
3.3.4 TIM/CNT

- ✓ TIM/CNT use the number 0000 ~ 4095, which can not be used twice.
- ✓ The operand of TIM/CNT can be constant, word address. Constant and content in word should be in BCD.
- ✓ Counter has the power failure memory function

3.3 Application instructions

➢Timer—TIM

✓Ladder symbol



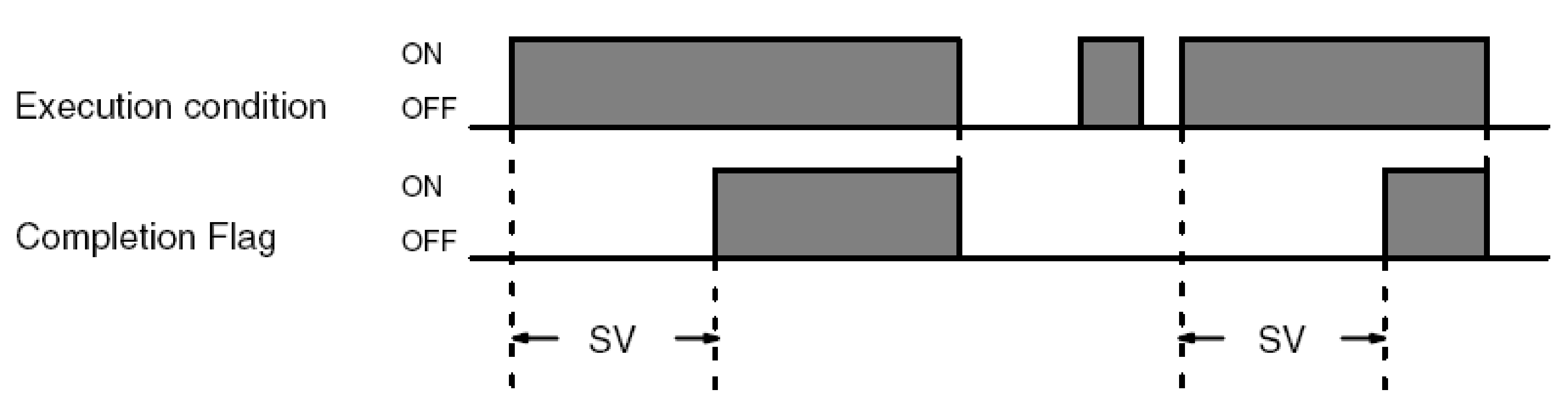
N: number 0000-4095

SV： 0-9999 Unit: 0.1 s Timing range: 0~999.9s

✓ Function

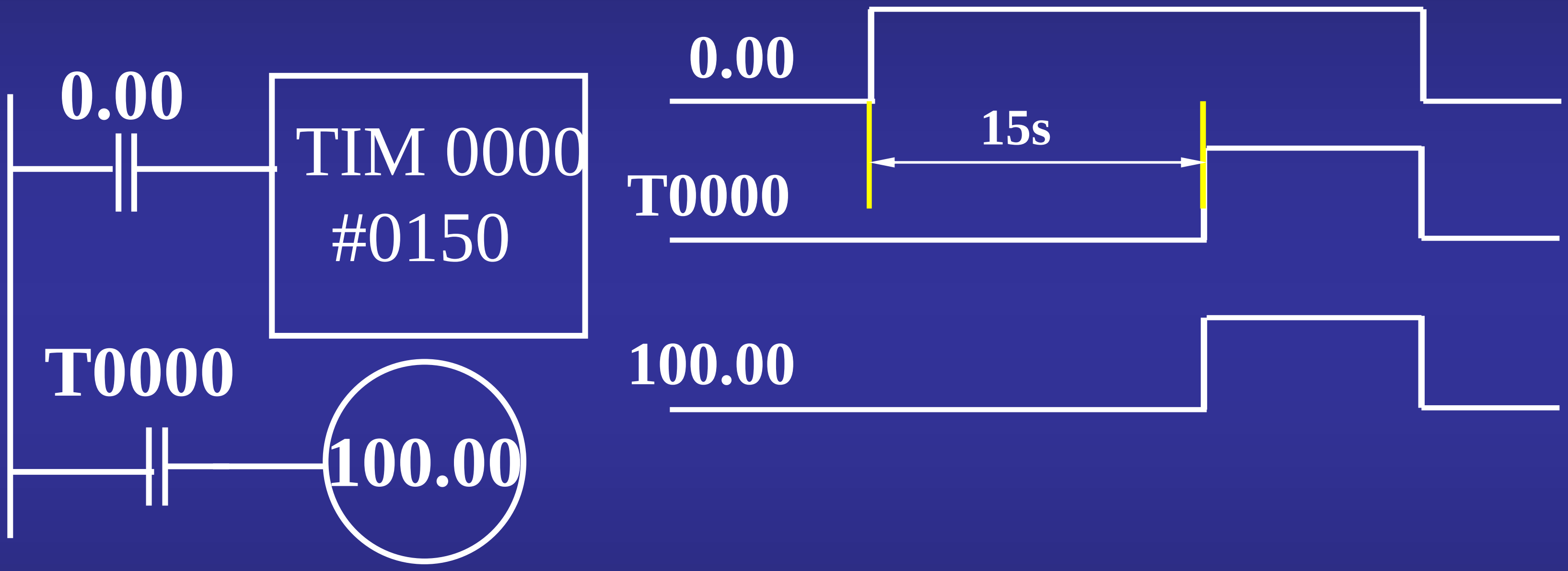
- On-delay timer
- If execution condition is OFF， Timer is reset， PV=SV, timer is OFF
- If execution condition is ON， timer is activated， PV=PV-1， when PV=0， timer is ON

3.3 Application instructions



3.3 Application instructions

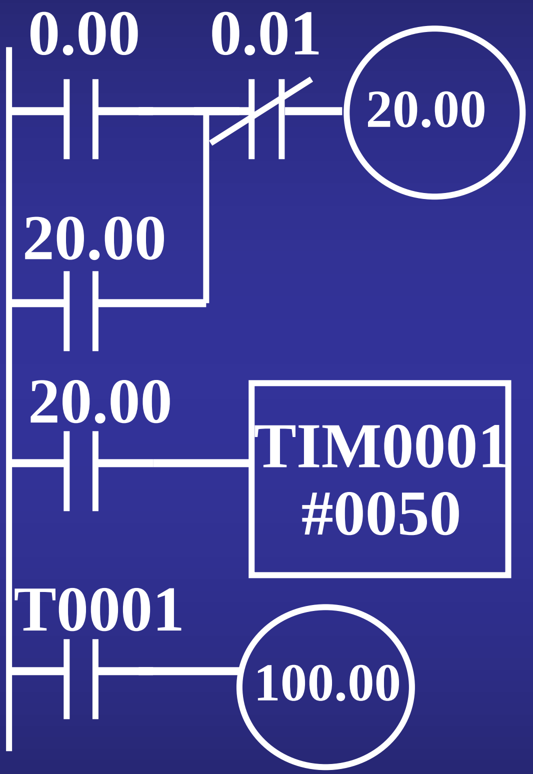
✓**Example**

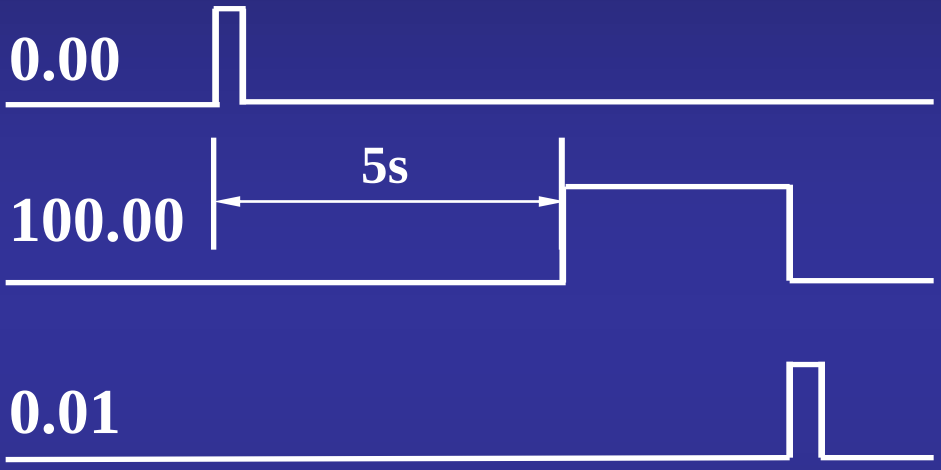


Time: 150 × 0.1 = 15s

3.3 Application instructions

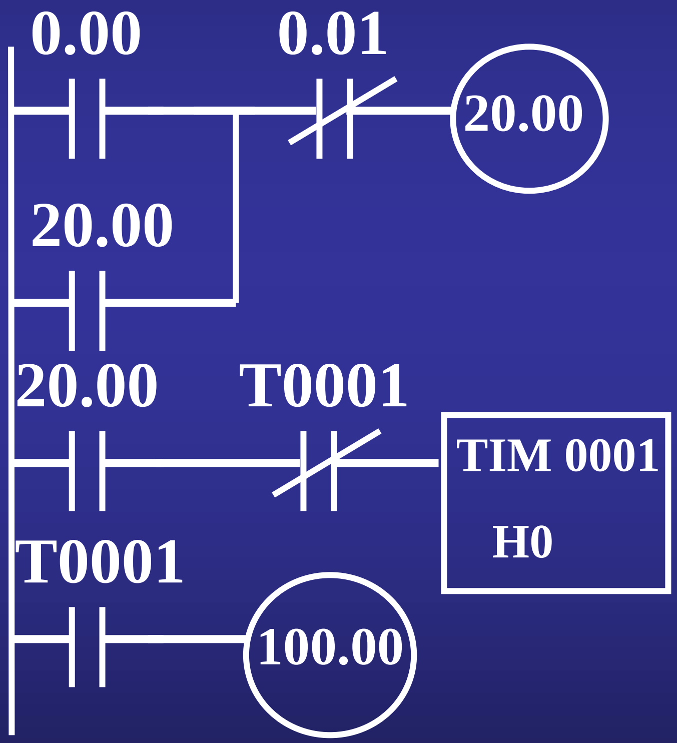
Example:

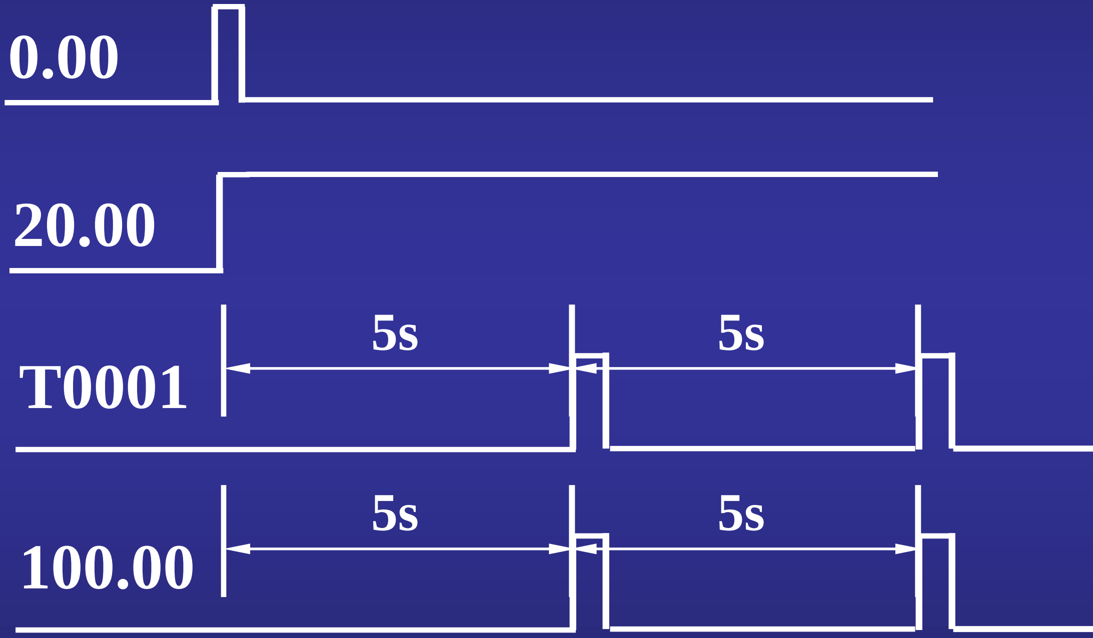




3.3 Application instructions

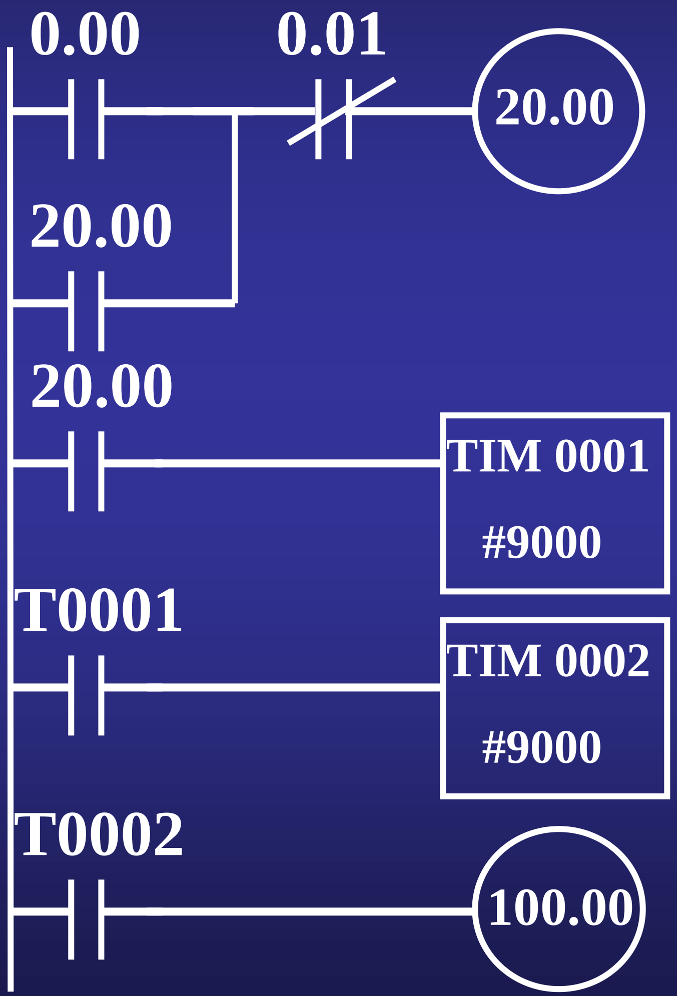
Example: Self-reset timing circuit

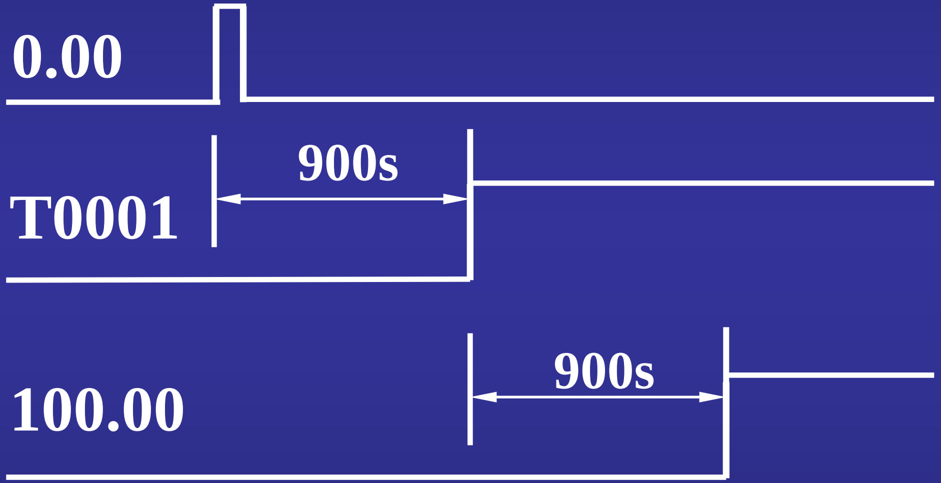




3.3 Application instructions

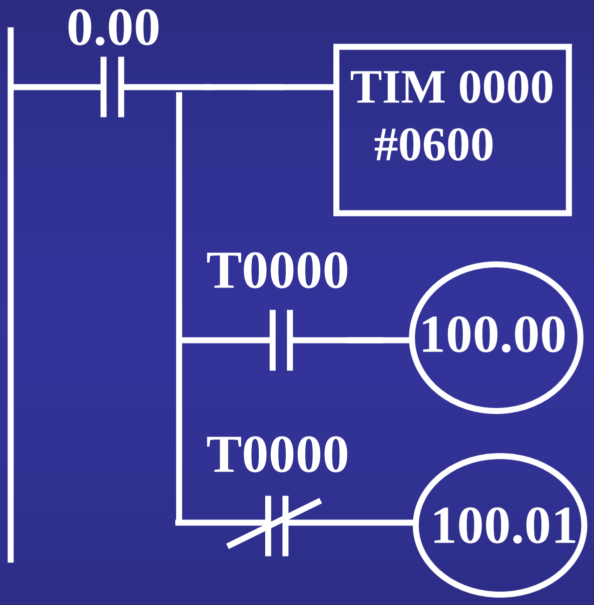
Example: cascade timer

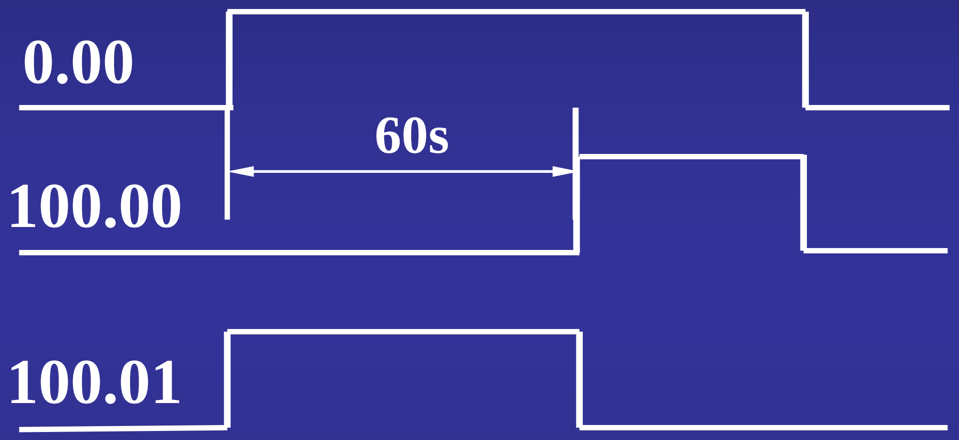




3.3 Application instructions

Example: On-delay

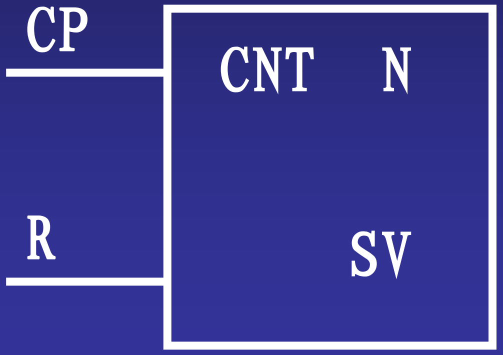




3.3 Application instructions

➢Counter—CNT

✓Ladder symbol



✓Description

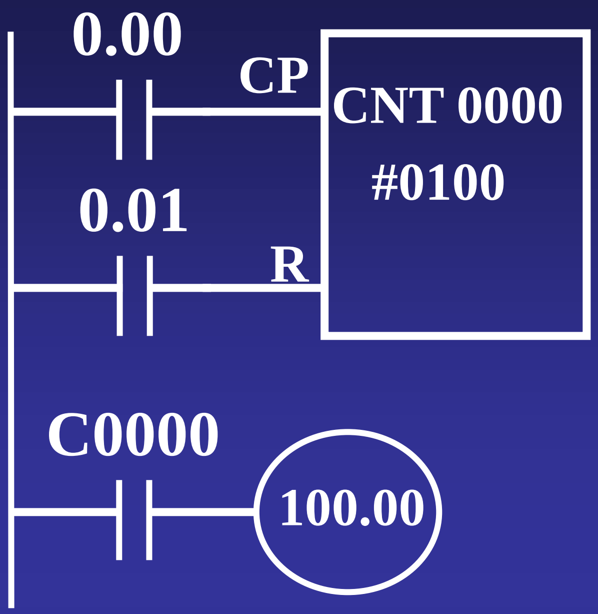
CP pulse input

R: reset input

- SV can be constant or word address
- CNT is a down counter, and has power failure memory function

3.3 Application instructions

✓**Analysis**



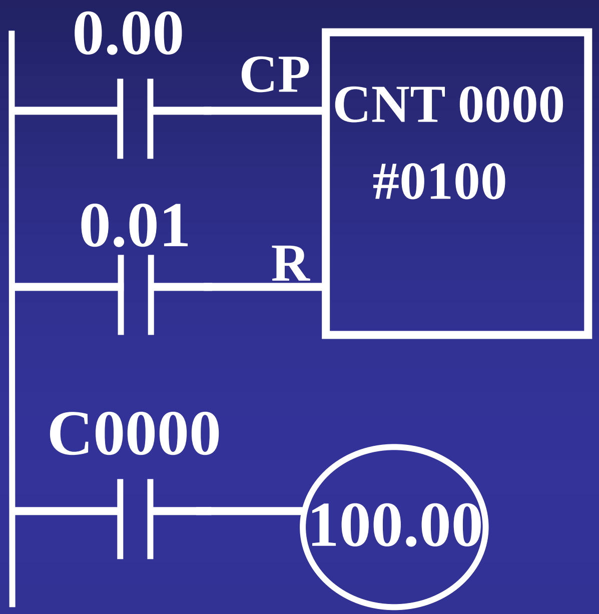
•If R is ON， counter is reset and PV=SV;

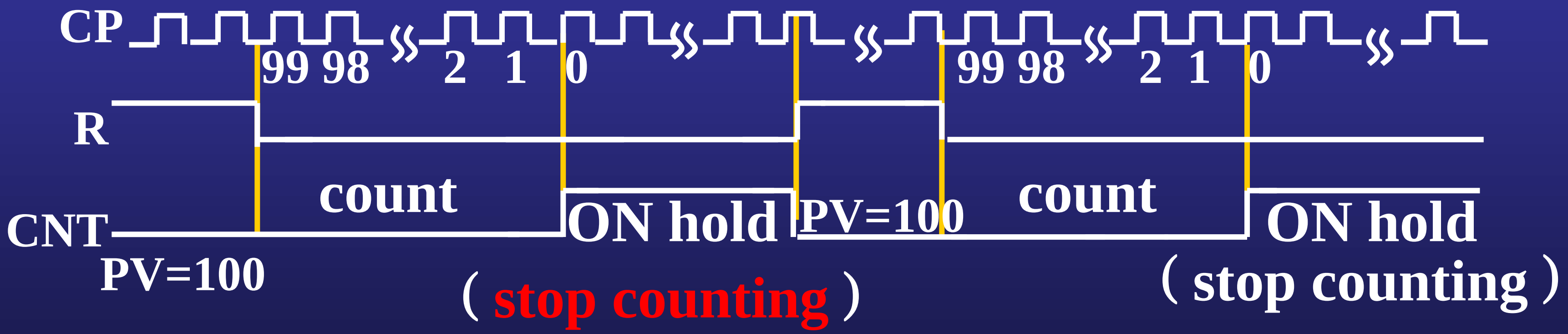
•When R from ON to OFF， counter **counts down from SV** .

•When 0.00 from OFF to ON， PV=PV-1;

•When PV=0, C0000 is ON , 100.00 is ON and counting is stopped.

3.3 Application instructions

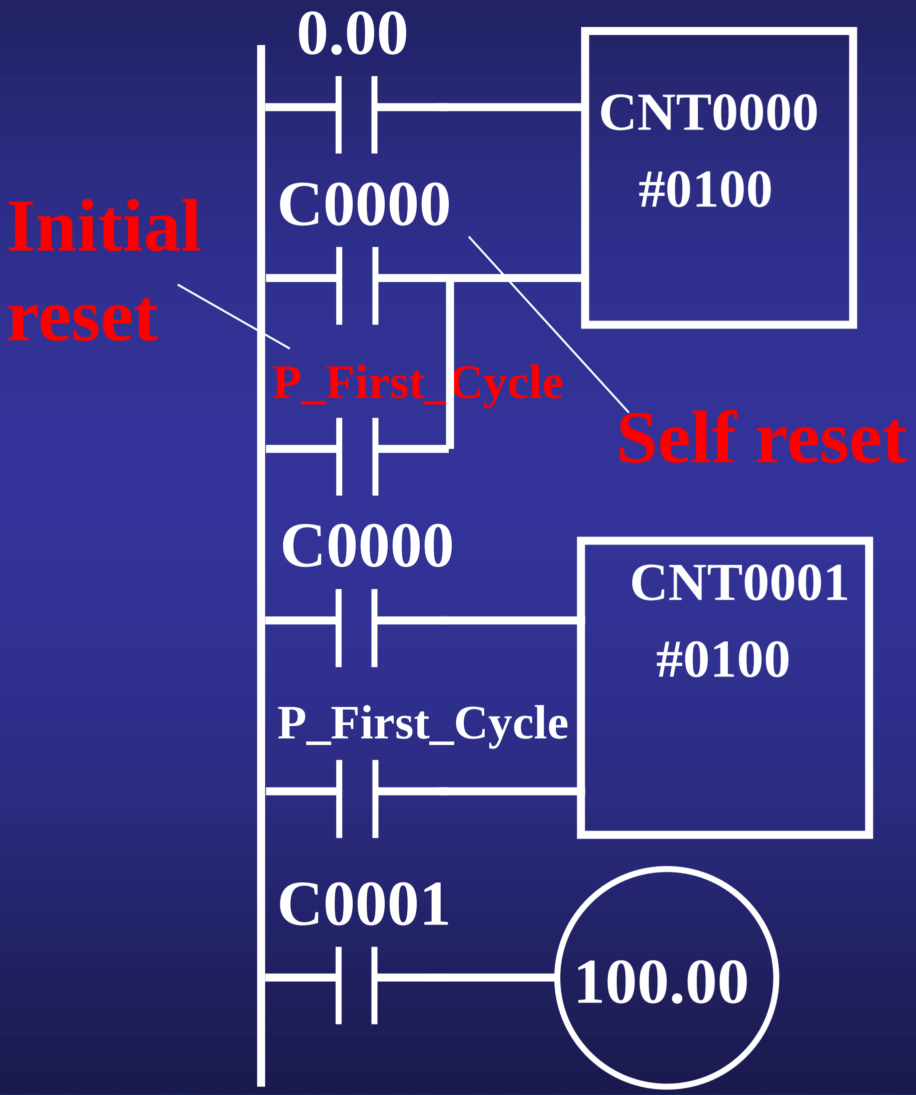


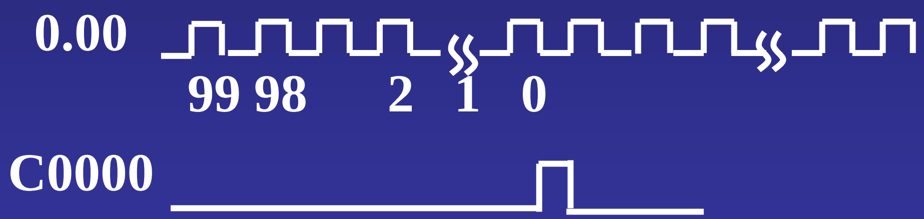


3.3 Application instructions

✓**Extended counter**

Can count 10000 pulses

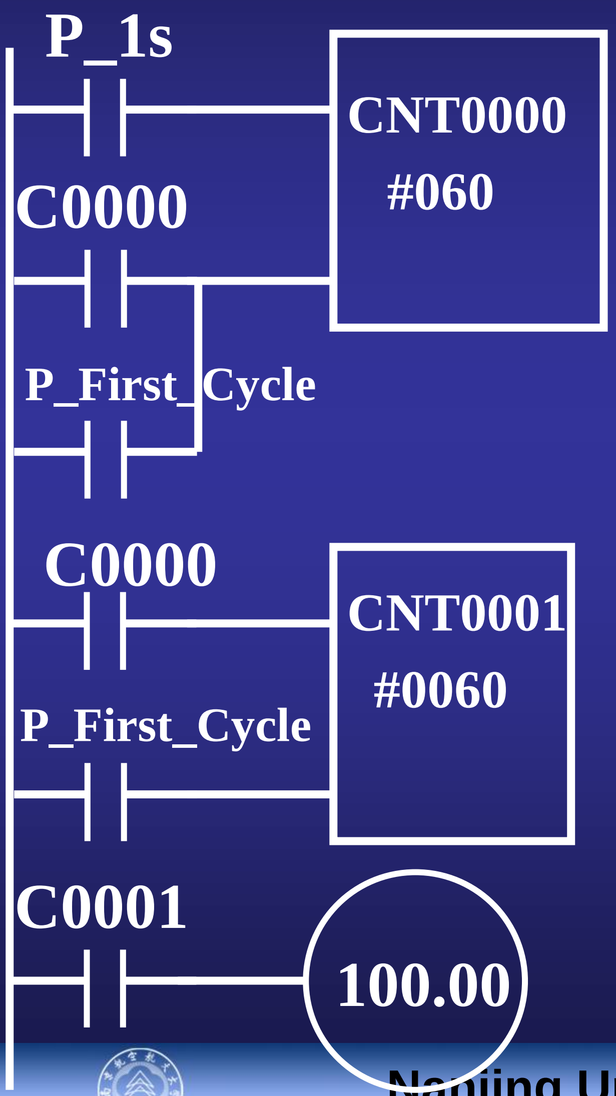


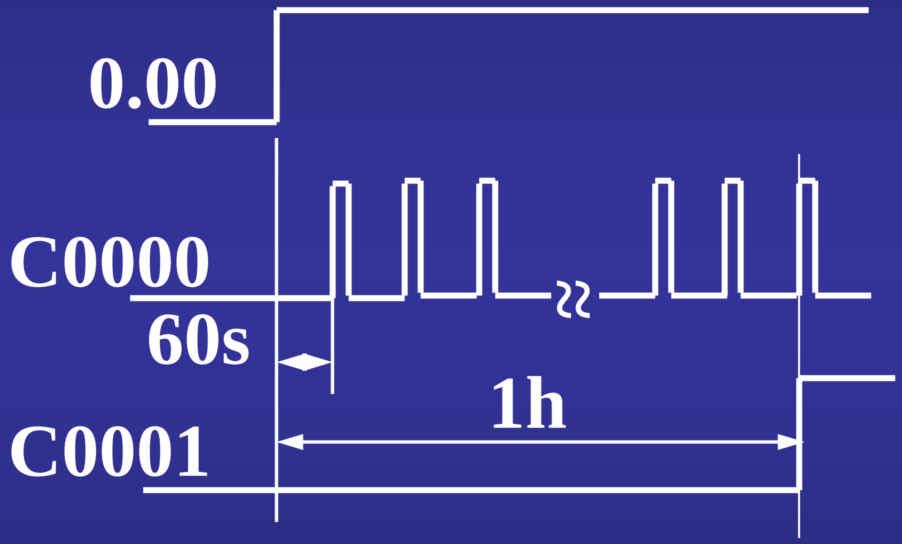


3.3 Application instructions

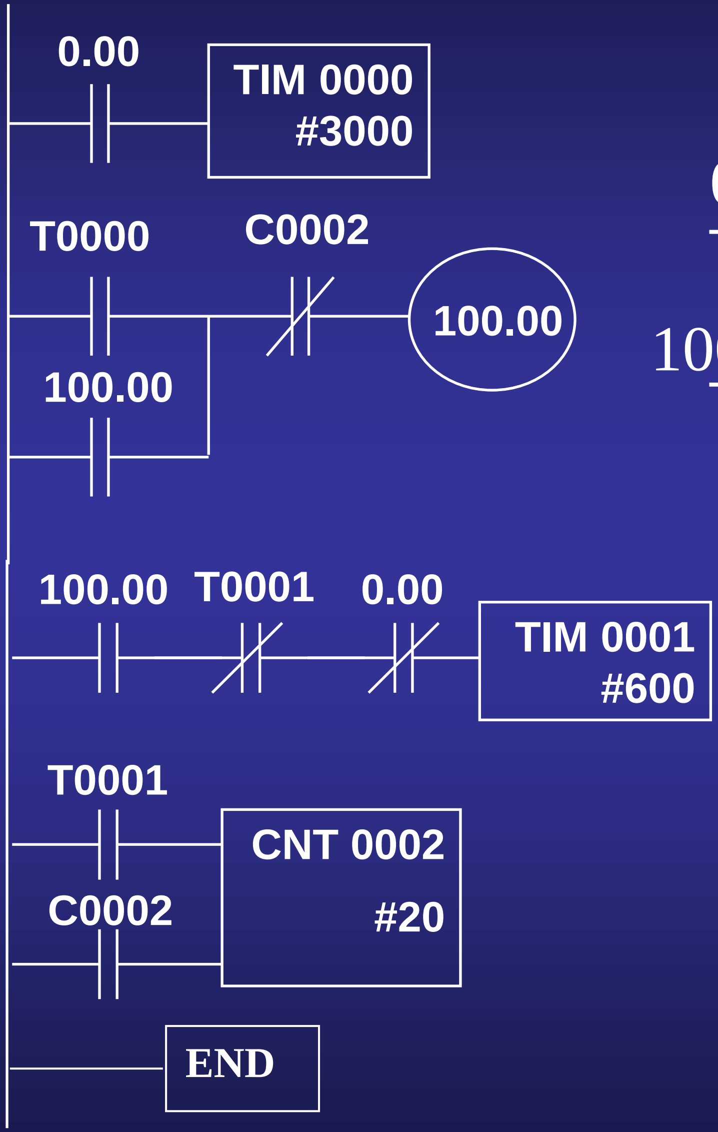
✓Extended timer

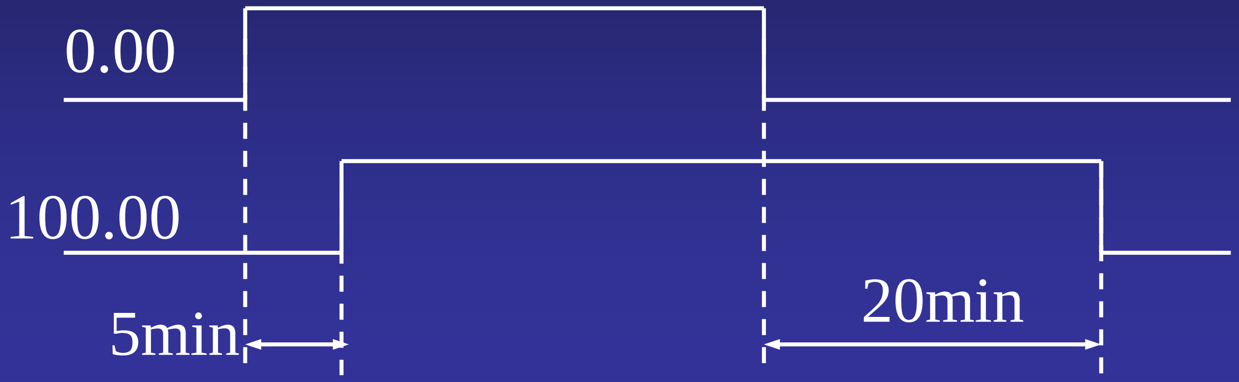
P_1s: 1s clock pulse





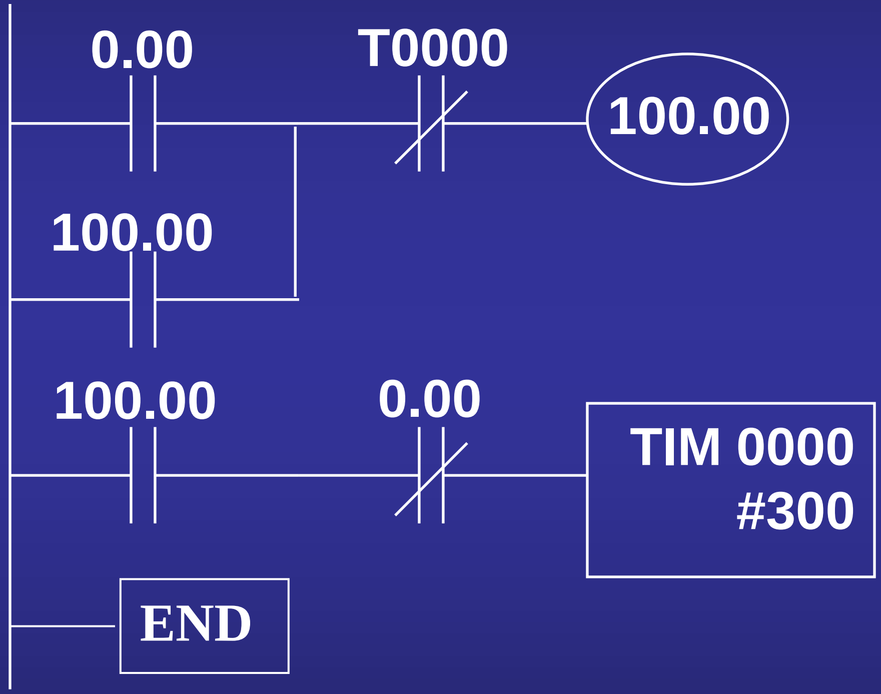
3.3 Application instructions





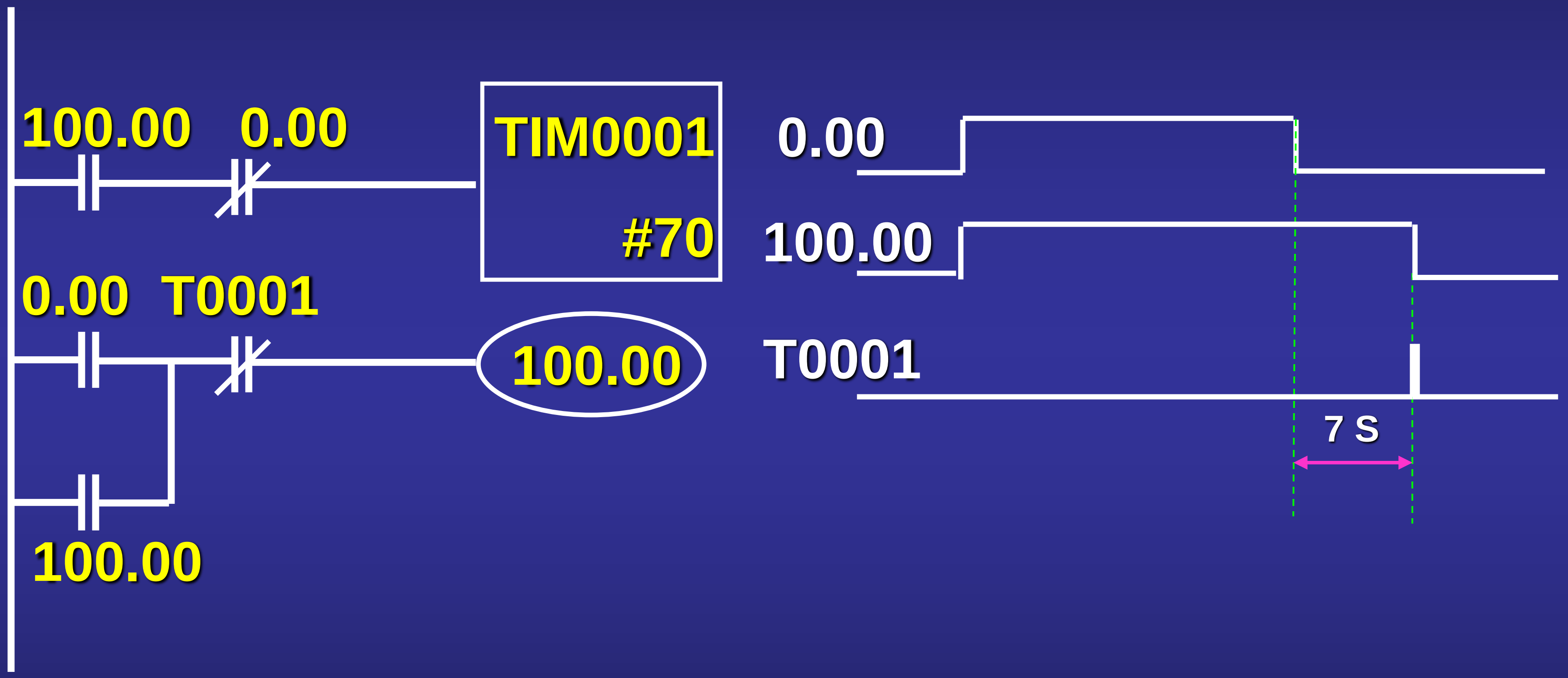
3.3 Application instructions

Lamp will light for 30s since the last press.



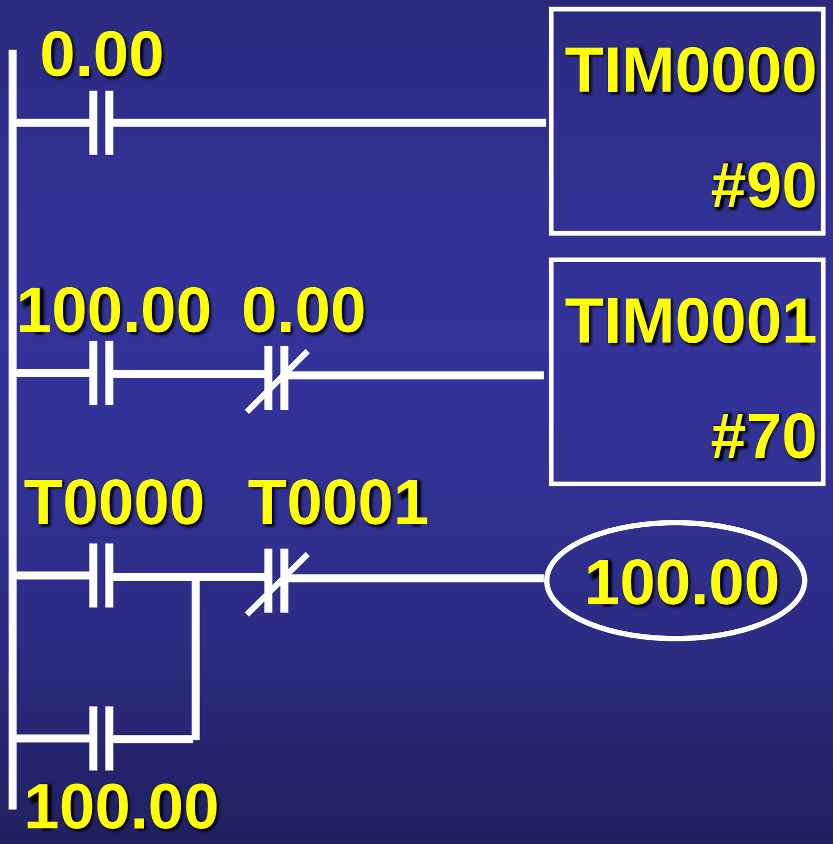
3.3 Application instructions

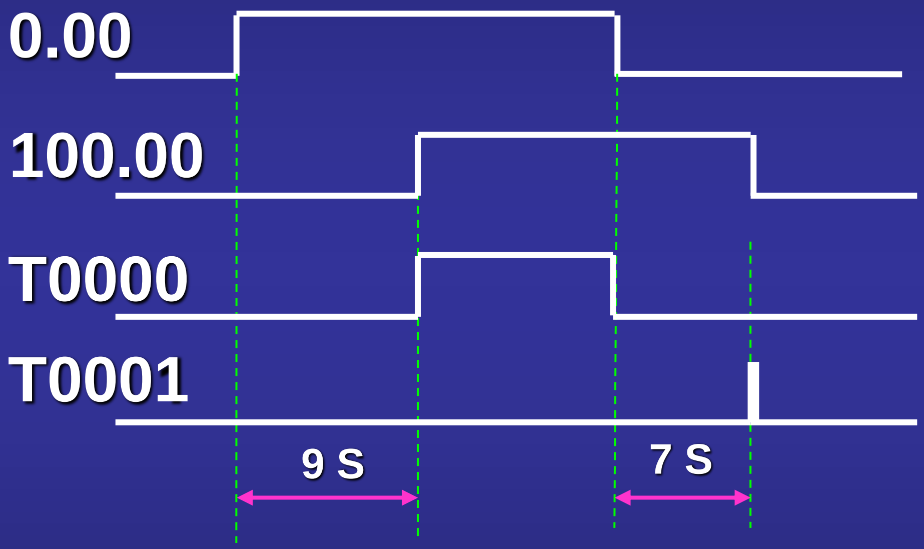
✓Off-delay



3.3 Application instructions

✓On delay/off delay


0.00
TIM0000
#90
100.00
0.00
TIM0001
#70
T0000
T0001
100.00
100.00


0.00
100.00
T0000
T0001
9 S
7 S