This program is totally useless...

as far as controlling any process. Now on the other hand if you would like to see how different instructions are converted read on.

This program was converted using the ANX option. Bits in the I/O Configuration will be remapped to AN-X-TIMAS I/O. Bits outside of this range will point to the BOOL array (IO) or the INT array (WIO).

This is a raw conversion except that the ANX import file has been imported.
DATE: 04-17-09
VERSION: 1.0

0 NOP

This section of rungs show how BITS are converted.

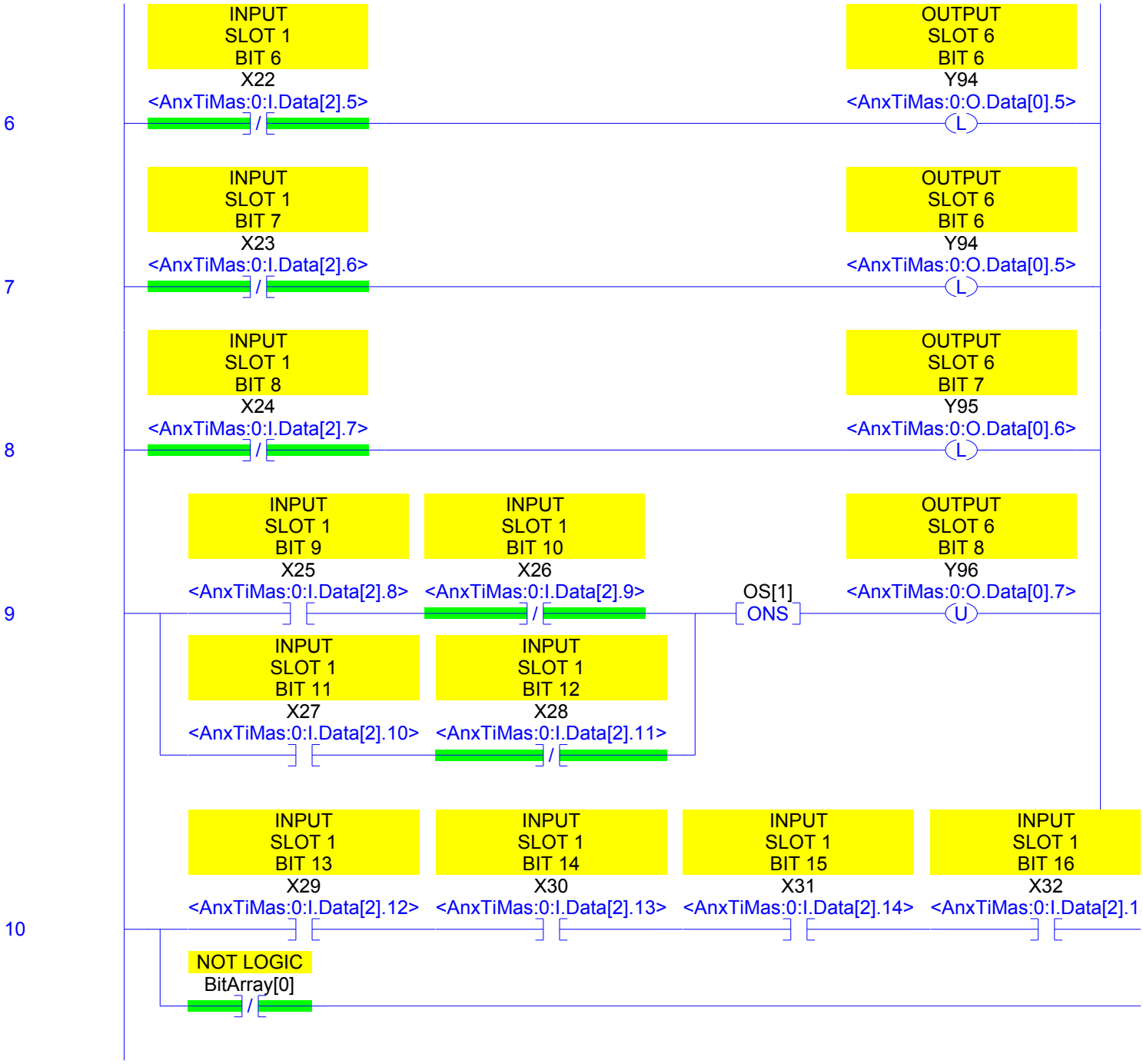
1 INPUT SLOT 1 BIT 1 — X17 — <AnxTiMas:0:I.Data[2].0> — OUTPUT SLOT 6 BIT 1 — Y89 — <AnxTiMas:0:O.Data[0].0>

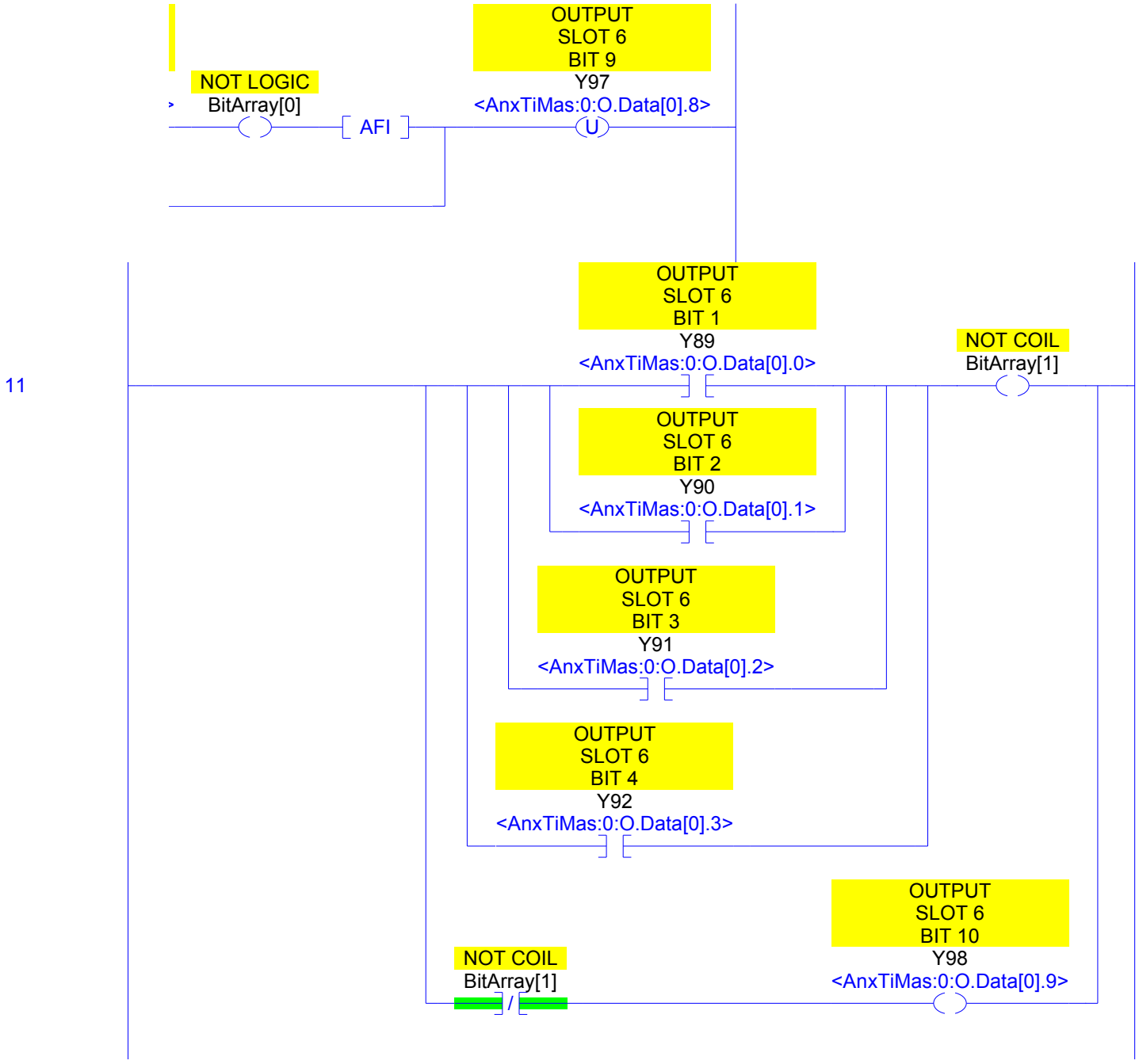
2 INPUT SLOT 1 BIT 2 — X18 — <AnxTiMas:0:I.Data[2].1> — OUTPUT SLOT 6 BIT 2 — Y90 — <AnxTiMas:0:O.Data[0].1>

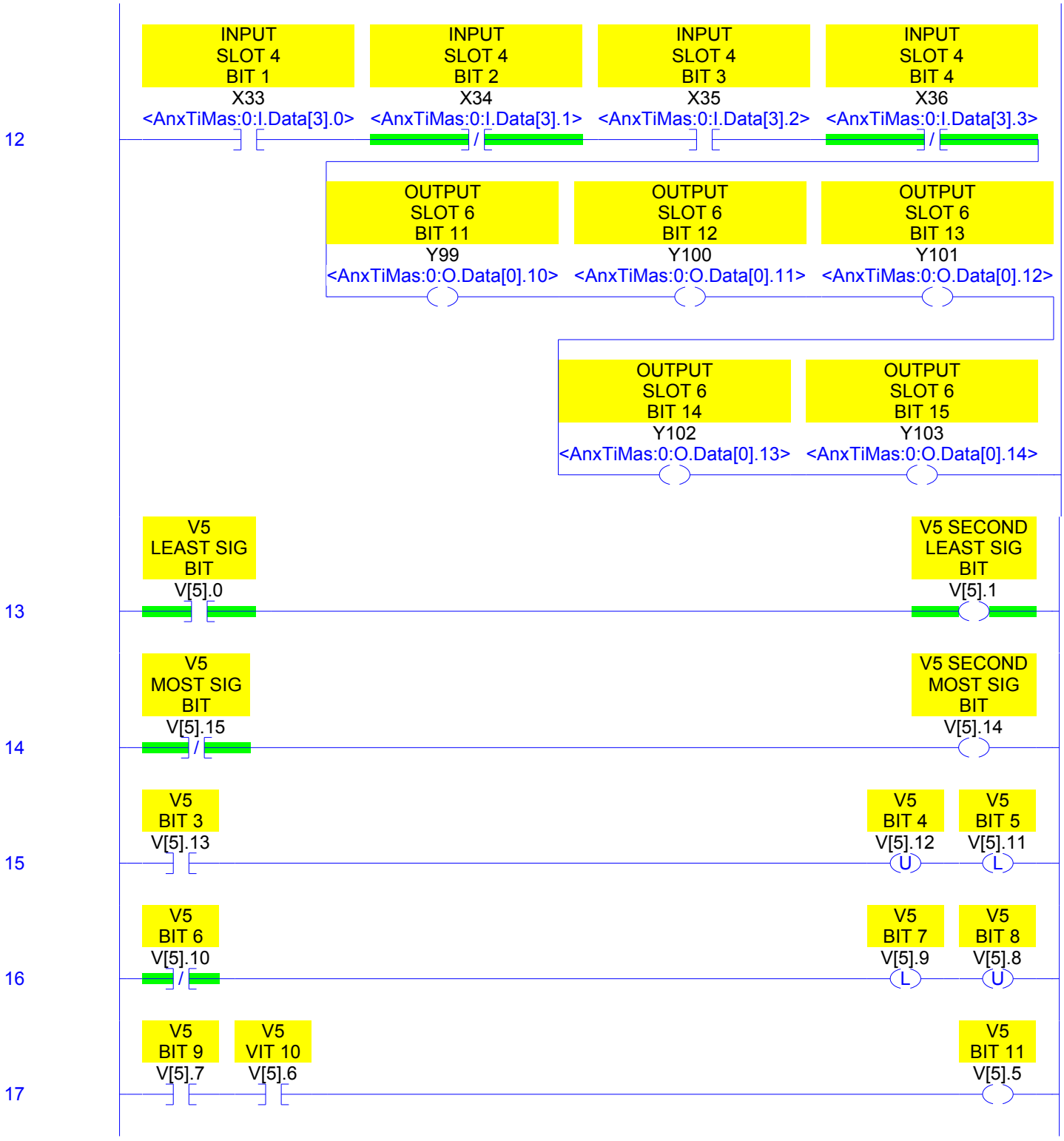
3 INPUT SLOT 1 BIT 3 — X19 — <AnxTiMas:0:I.Data[2].2> — OUTPUT SLOT 6 BIT 3 — Y91 — <AnxTiMas:0:O.Data[0].2>

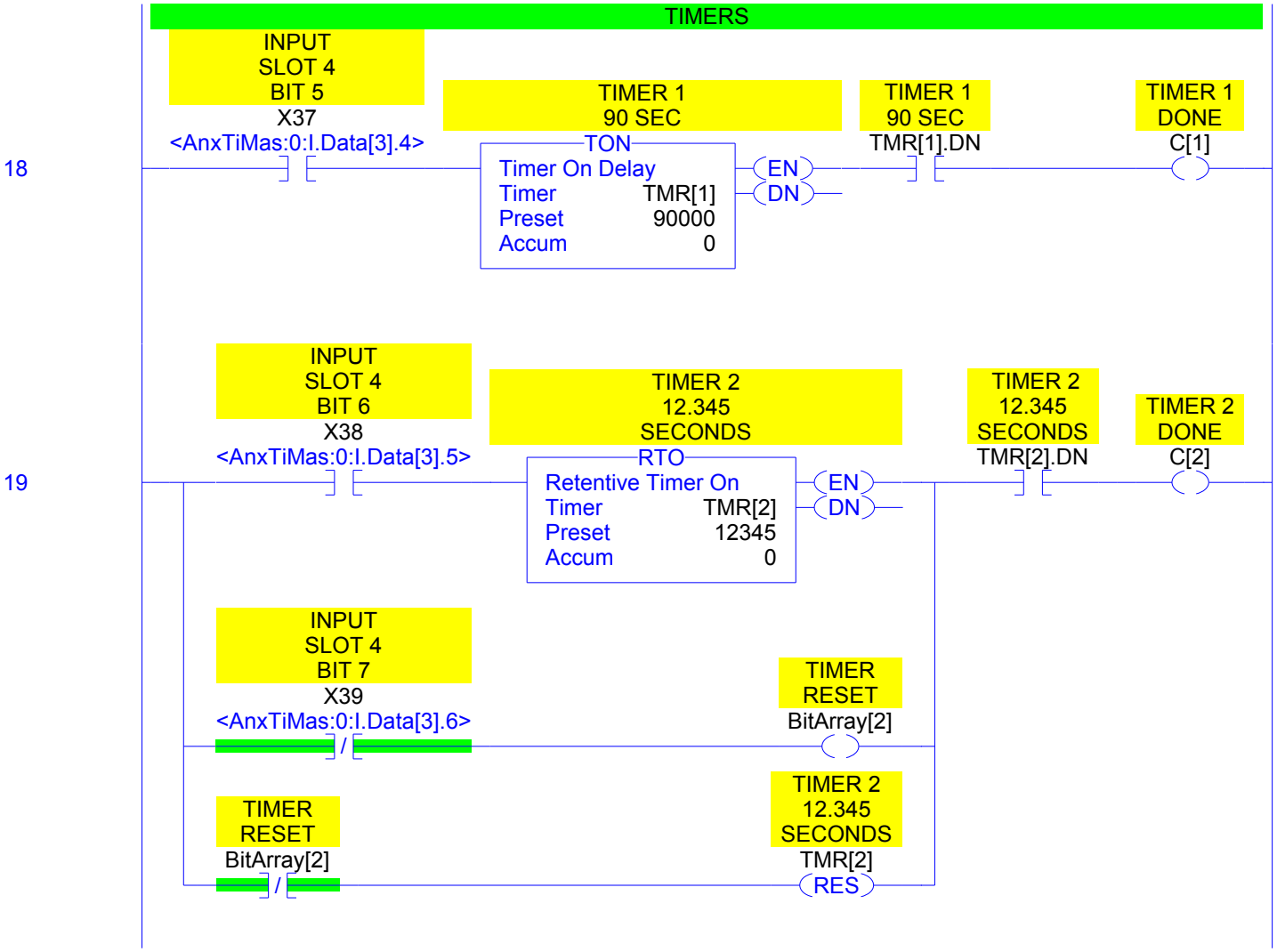
4 INPUT SLOT 1 BIT 4 — X20 — <AnxTiMas:0:I.Data[2].3> — OUTPUT SLOT 6 BIT 4 — Y92 — <AnxTiMas:0:O.Data[0].3>

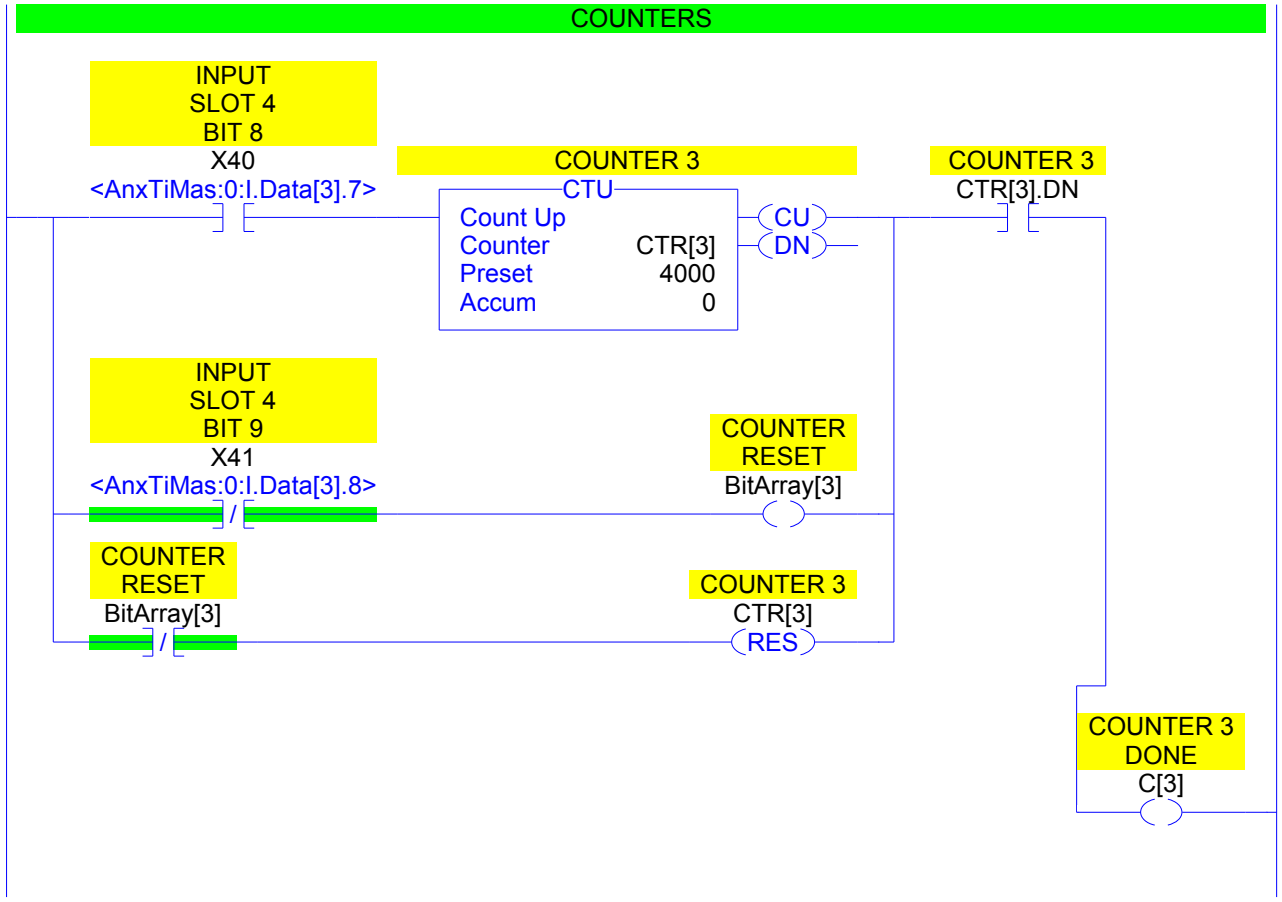
5 INPUT SLOT 1 BIT 5 — X21 — <AnxTiMas:0:I.Data[2].4> — OUTPUT SLOT 6 BIT 5 — Y93 — <AnxTiMas:0:O.Data[0].4>


6
INPUT SLOT 1 BIT 6
X22
<AnxTiMas:0:I.Data[2].5>
/
OUTPUT SLOT 6 BIT 6
Y94
<AnxTiMas:0:O.Data[0].5>
L
7
INPUT SLOT 1 BIT 7
X23
<AnxTiMas:0:I.Data[2].6>
/
OUTPUT SLOT 6 BIT 6
Y94
<AnxTiMas:0:O.Data[0].5>
L
8
INPUT SLOT 1 BIT 8
X24
<AnxTiMas:0:I.Data[2].7>
/
OUTPUT SLOT 6 BIT 7
Y95
<AnxTiMas:0:O.Data[0].6>
L
9
INPUT SLOT 1 BIT 9
X25
<AnxTiMas:0:I.Data[2].8>
INPUT SLOT 1 BIT 10
X26
<AnxTiMas:0:I.Data[2].9>
/
OS[1]
ONS
OUTPUT SLOT 6 BIT 8
Y96
<AnxTiMas:0:O.Data[0].7>
U
INPUT SLOT 1 BIT 11
X27
<AnxTiMas:0:I.Data[2].10>
INPUT SLOT 1 BIT 12
X28
<AnxTiMas:0:I.Data[2].11>
/
10
INPUT SLOT 1 BIT 13
X29
<AnxTiMas:0:I.Data[2].12>
INPUT SLOT 1 BIT 14
X30
<AnxTiMas:0:I.Data[2].13>
INPUT SLOT 1 BIT 15
X31
<AnxTiMas:0:I.Data[2].14>
INPUT SLOT 1 BIT 16
X32
<AnxTiMas:0:I.Data[2].1
NOT LOGIC
BitArray[0]
/

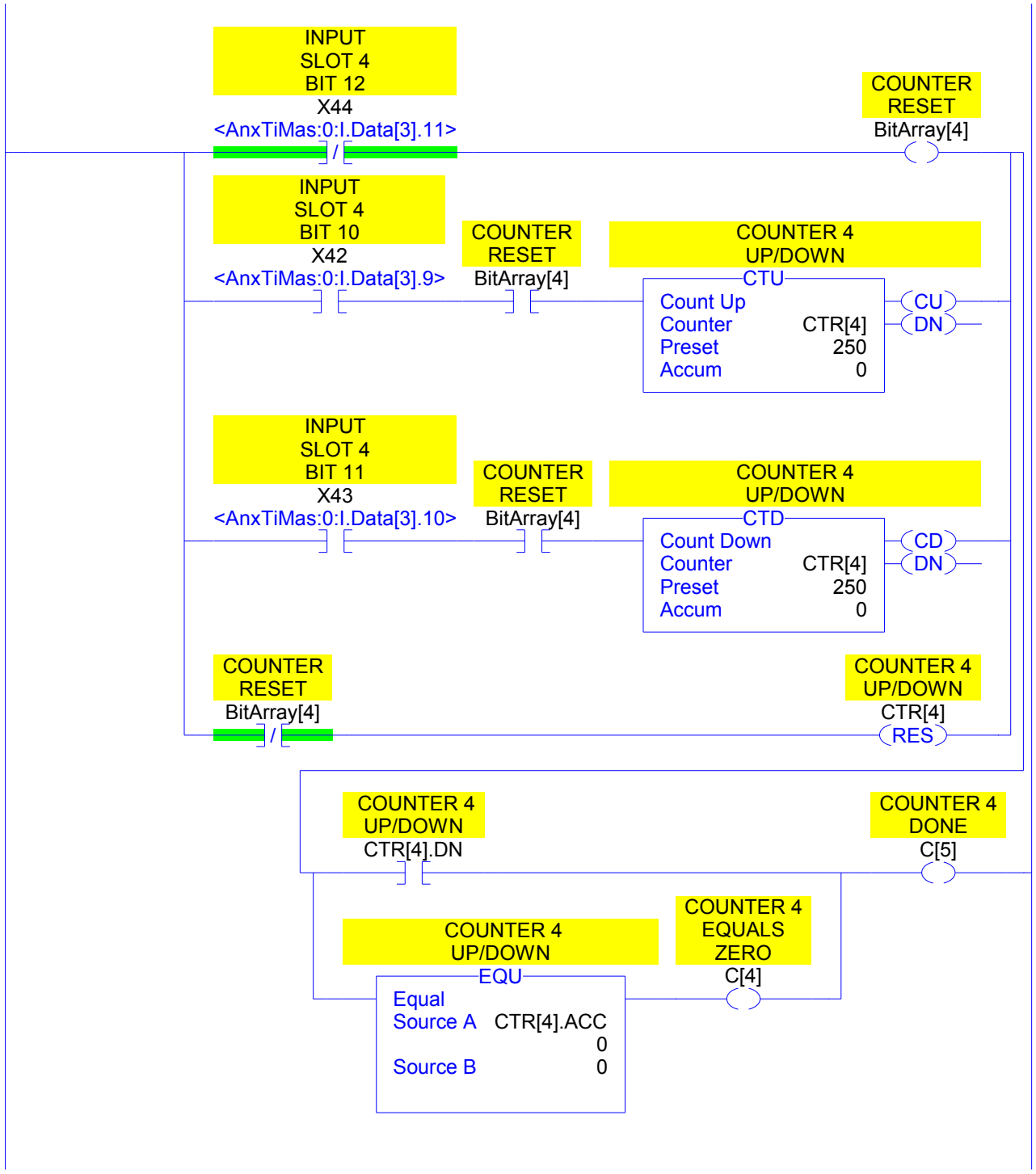

NOT LOGIC
BitArray[0]
AFI
OUTPUT
SLOT 6
BIT 9
Y97
<AnxTiMas:0:O.Data[0].8>
U
11
OUTPUT
SLOT 6
BIT 1
Y89
<AnxTiMas:0:O.Data[0].0>
NOT COIL
BitArray[1]
OUTPUT
SLOT 6
BIT 2
Y90
<AnxTiMas:0:O.Data[0].1>
OUTPUT
SLOT 6
BIT 3
Y91
<AnxTiMas:0:O.Data[0].2>
OUTPUT
SLOT 6
BIT 4
Y92
<AnxTiMas:0:O.Data[0].3>
OUTPUT
SLOT 6
BIT 10
Y98
<AnxTiMas:0:O.Data[0].9>
NOT COIL
BitArray[1]
/

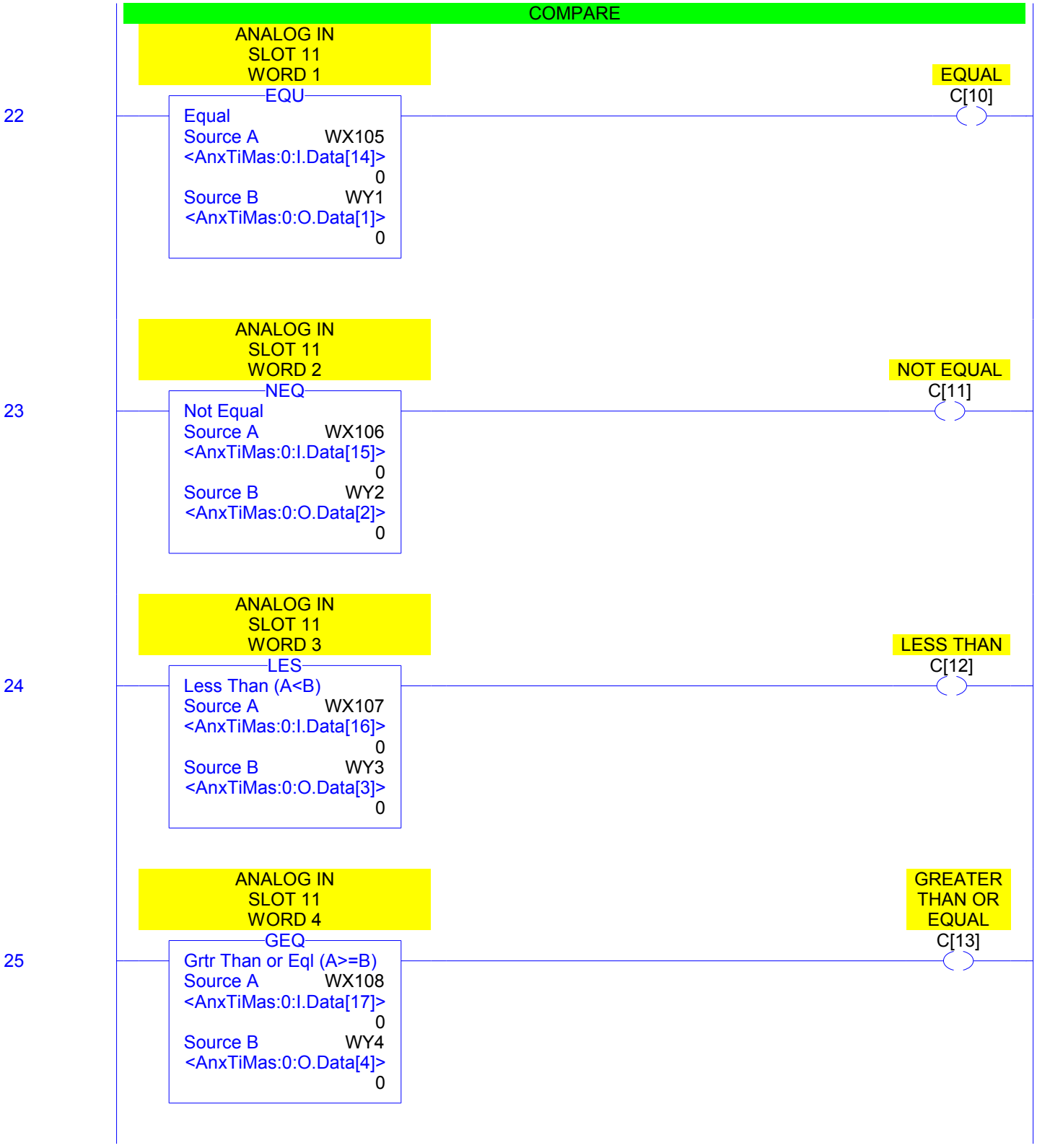

12
INPUT SLOT 4 BIT 1
X33
<AnxTiMas:0:I.Data[3].0>
INPUT SLOT 4 BIT 2
X34
<AnxTiMas:0:I.Data[3].1>
INPUT SLOT 4 BIT 3
X35
<AnxTiMas:0:I.Data[3].2>
INPUT SLOT 4 BIT 4
X36
<AnxTiMas:0:I.Data[3].3>
OUTPUT SLOT 6 BIT 11
Y99
<AnxTiMas:0:O.Data[0].10>
OUTPUT SLOT 6 BIT 12
Y100
<AnxTiMas:0:O.Data[0].11>
OUTPUT SLOT 6 BIT 13
Y101
<AnxTiMas:0:O.Data[0].12>
OUTPUT SLOT 6 BIT 14
Y102
<AnxTiMas:0:O.Data[0].13>
OUTPUT SLOT 6 BIT 15
Y103
<AnxTiMas:0:O.Data[0].14>
13
V5 LEAST SIG BIT
V[5].0
V5 SECOND LEAST SIG BIT
V[5].1
14
V5 MOST SIG BIT
V[5].15
V5 SECOND MOST SIG BIT
V[5].14
15
V5 BIT 3
V[5].13
V5 BIT 4
V[5].12
U
V5 BIT 5
V[5].11
L
16
V5 BIT 6
V[5].10
V5 BIT 7
V[5].9
L
V5 BIT 8
V[5].8
U
17
V5 BIT 9
V[5].7
V5 VIT 10
V[5].6
V5 BIT 11
V[5].5

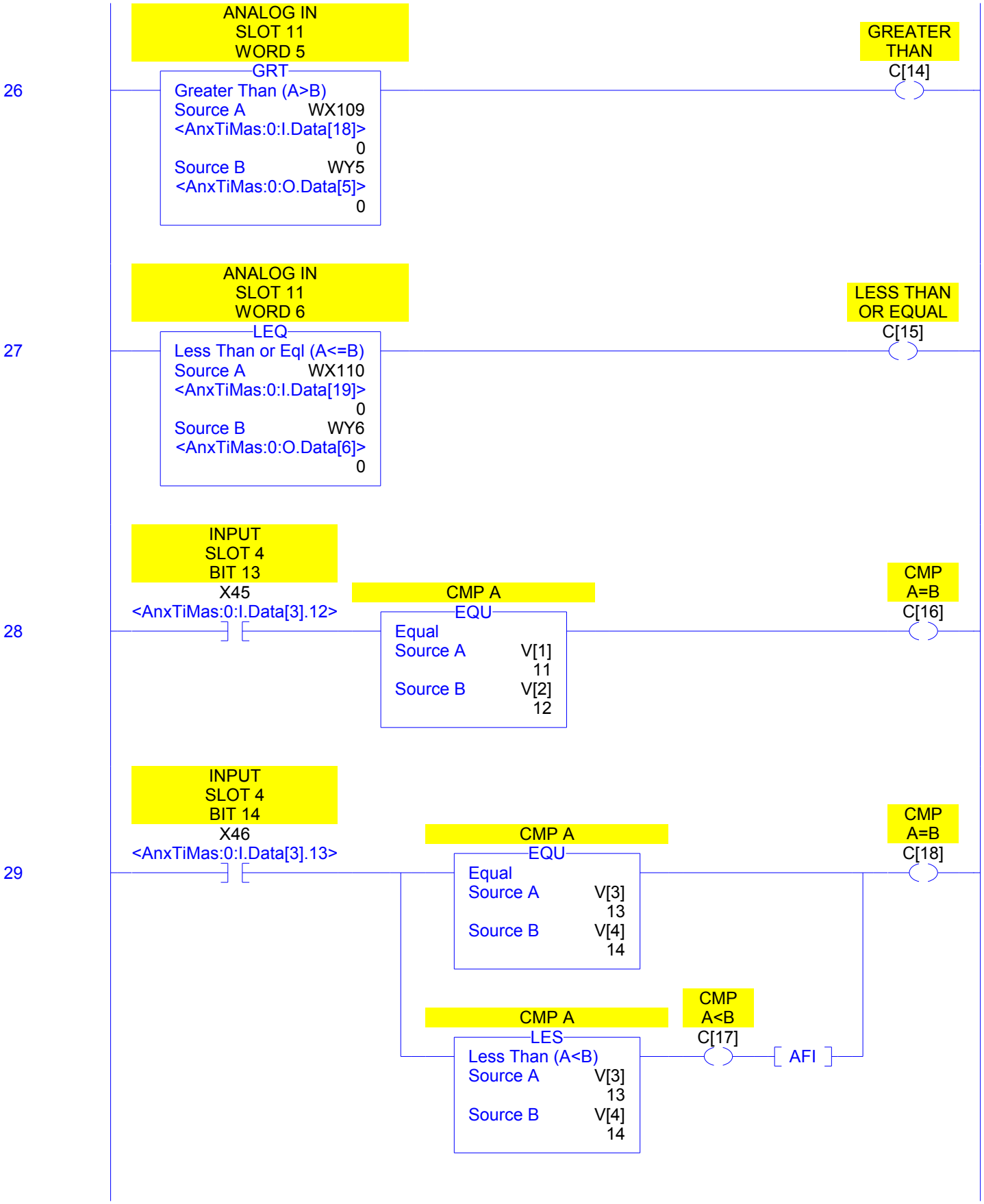

TIMERS
18
INPUT
SLOT 4
BIT 5
X37
<AnxTiMas:0:I.Data[3].4>
TIMER 1
90 SEC
TON
Timer On Delay
Timer TMR[1]
Preset 90000
Accum 0
EN
DN
TIMER 1
90 SEC
TMR[1].DN
TIMER 1
DONE
C[1]
19
INPUT
SLOT 4
BIT 6
X38
<AnxTiMas:0:I.Data[3].5>
TIMER 2
12.345
SECONDS
RTO
Retentive Timer On
Timer TMR[2]
Preset 12345
Accum 0
EN
DN
TIMER 2
12.345
SECONDS
TMR[2].DN
TIMER 2
DONE
C[2]
INPUT
SLOT 4
BIT 7
X39
<AnxTiMas:0:I.Data[3].6>
TIMER
RESET
BitArray[2]
TIMER
RESET
BitArray[2]
TIMER 2
12.345
SECONDS
TMR[2]
RES

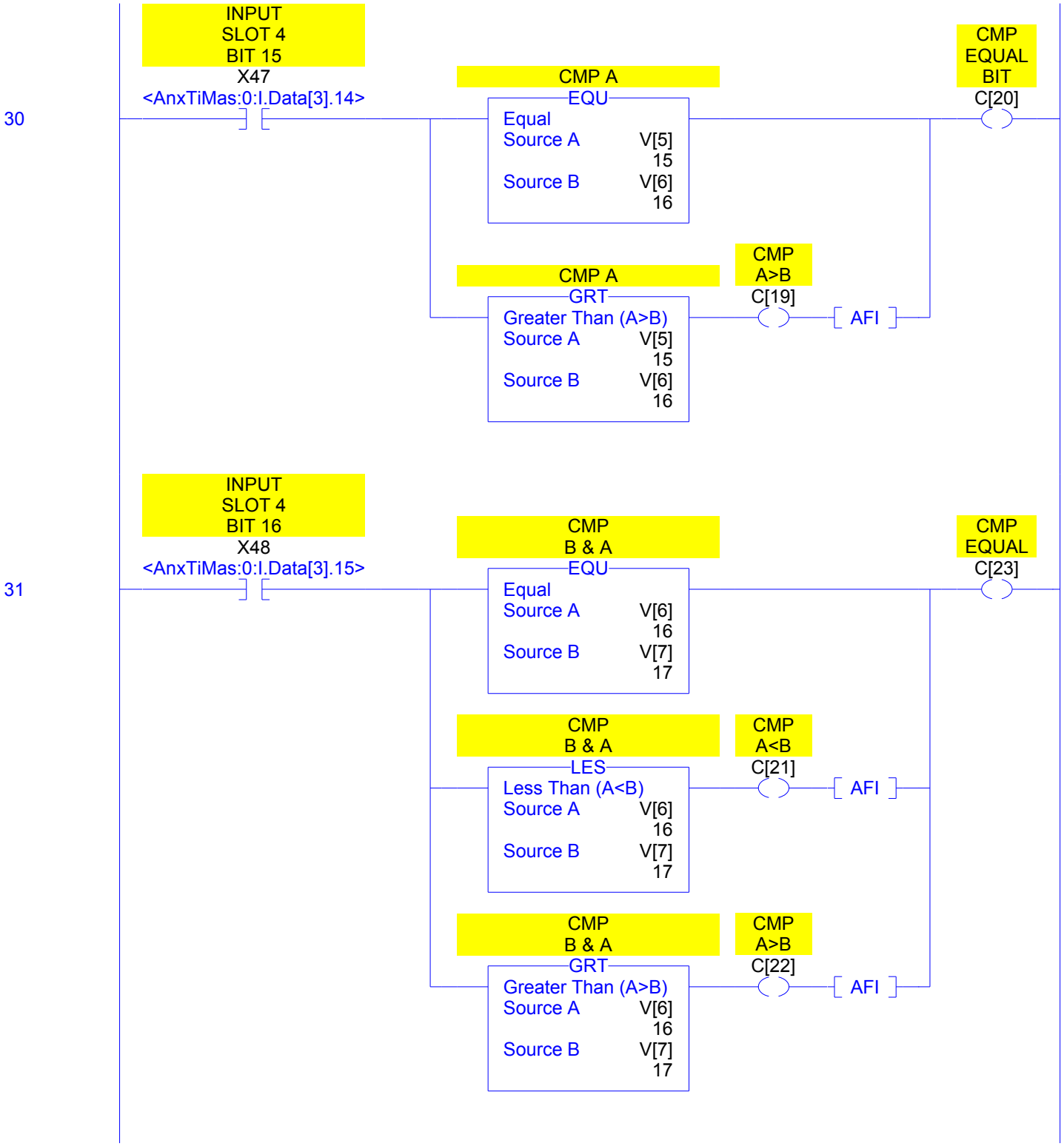

COUNTERS
20
INPUT
SLOT 4
BIT 8
X40
<AnxTiMas:0:I.Data[3].7>
COUNTER 3
CTU
Count Up
Counter CTR[3]
Preset 4000
Accum 0
CU
DN
COUNTER 3
CTR[3].DN
INPUT
SLOT 4
BIT 9
X41
<AnxTiMas:0:I.Data[3].8>
/
COUNTER
RESET
BitArray[3]
COUNTER
RESET
BitArray[3]
/
COUNTER 3
CTR[3]
RES
COUNTER 3
DONE
C[3]

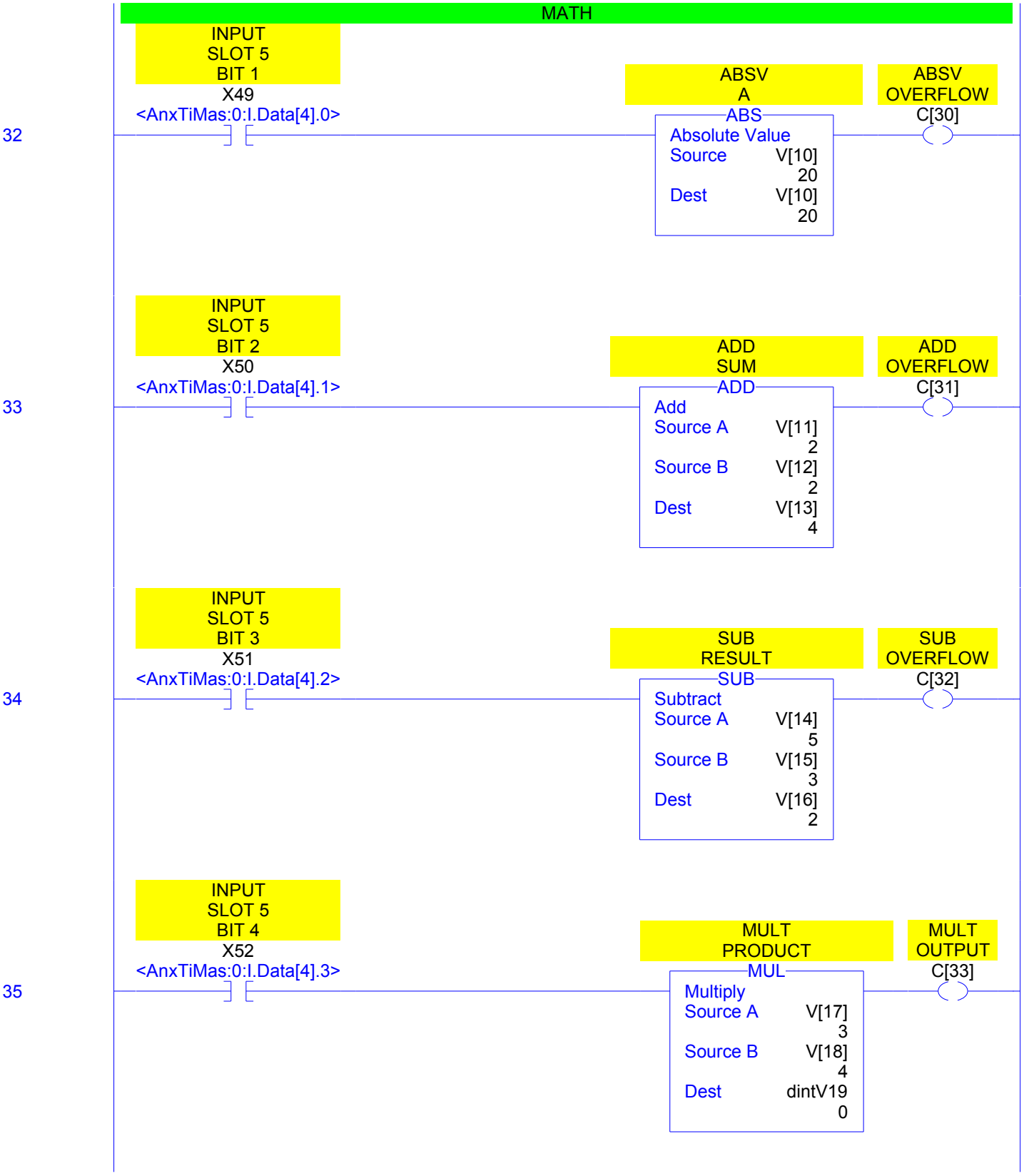
21

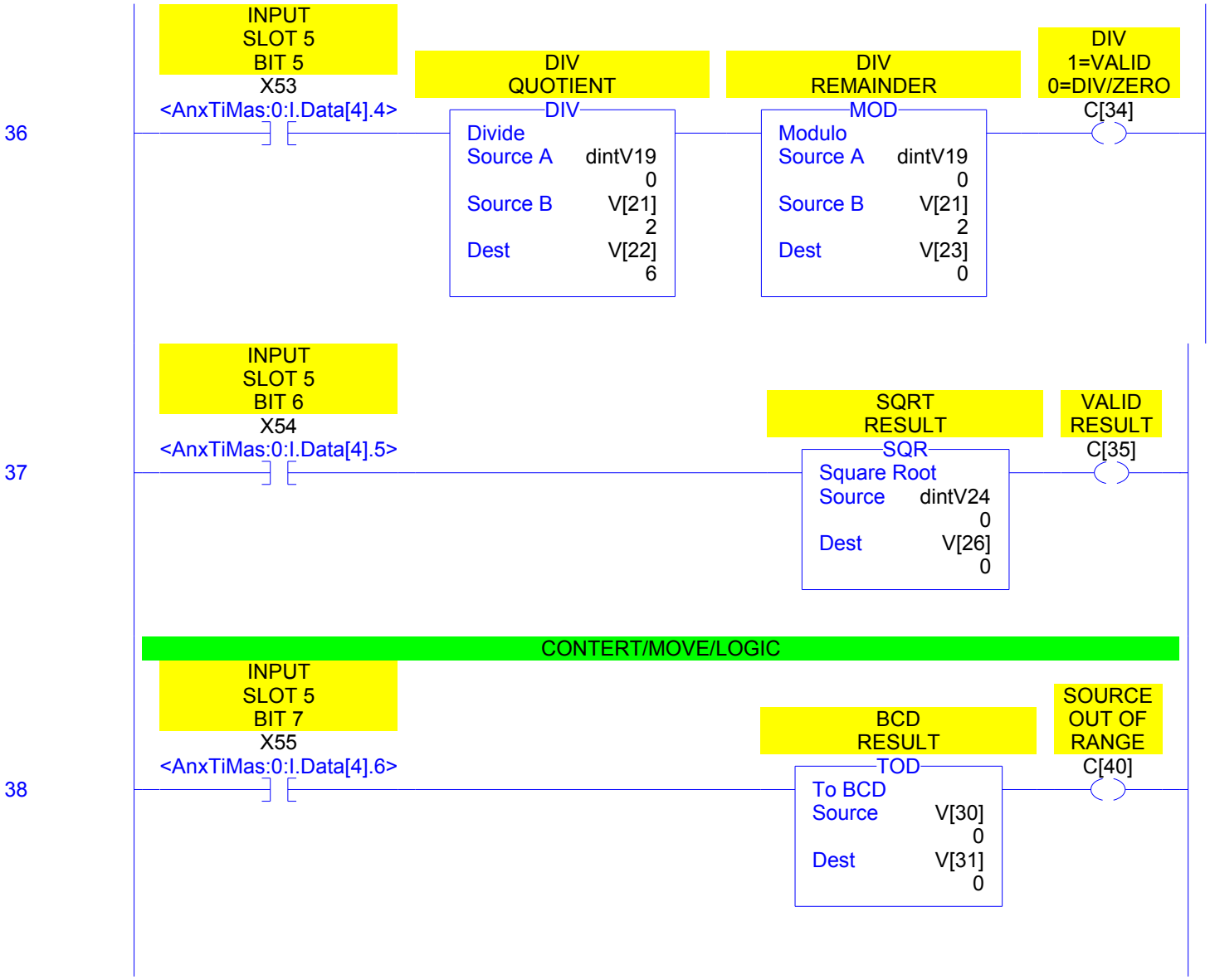

INPUT
SLOT 4
BIT 12
X44
<AnxTiMas:0:I.Data[3].11>
COUNTER
RESET
BitArray[4]
INPUT
SLOT 4
BIT 10
X42
<AnxTiMas:0:I.Data[3].9>
COUNTER
RESET
BitArray[4]
COUNTER 4
UP/DOWN
CTU
Count Up
Counter CTR[4]
Preset 250
Accum 0
CU
DN
INPUT
SLOT 4
BIT 11
X43
<AnxTiMas:0:I.Data[3].10>
COUNTER
RESET
BitArray[4]
COUNTER 4
UP/DOWN
CTD
Count Down
Counter CTR[4]
Preset 250
Accum 0
CD
DN
COUNTER
RESET
BitArray[4]
COUNTER 4
UP/DOWN
CTR[4]
RES
COUNTER 4
UP/DOWN
CTR[4].DN
COUNTER 4
DONE
C[5]
COUNTER 4
UP/DOWN
EQU
Equal
Source A CTR[4].ACC
0
Source B 0
COUNTER 4
EQUALS
ZERO
C[4]

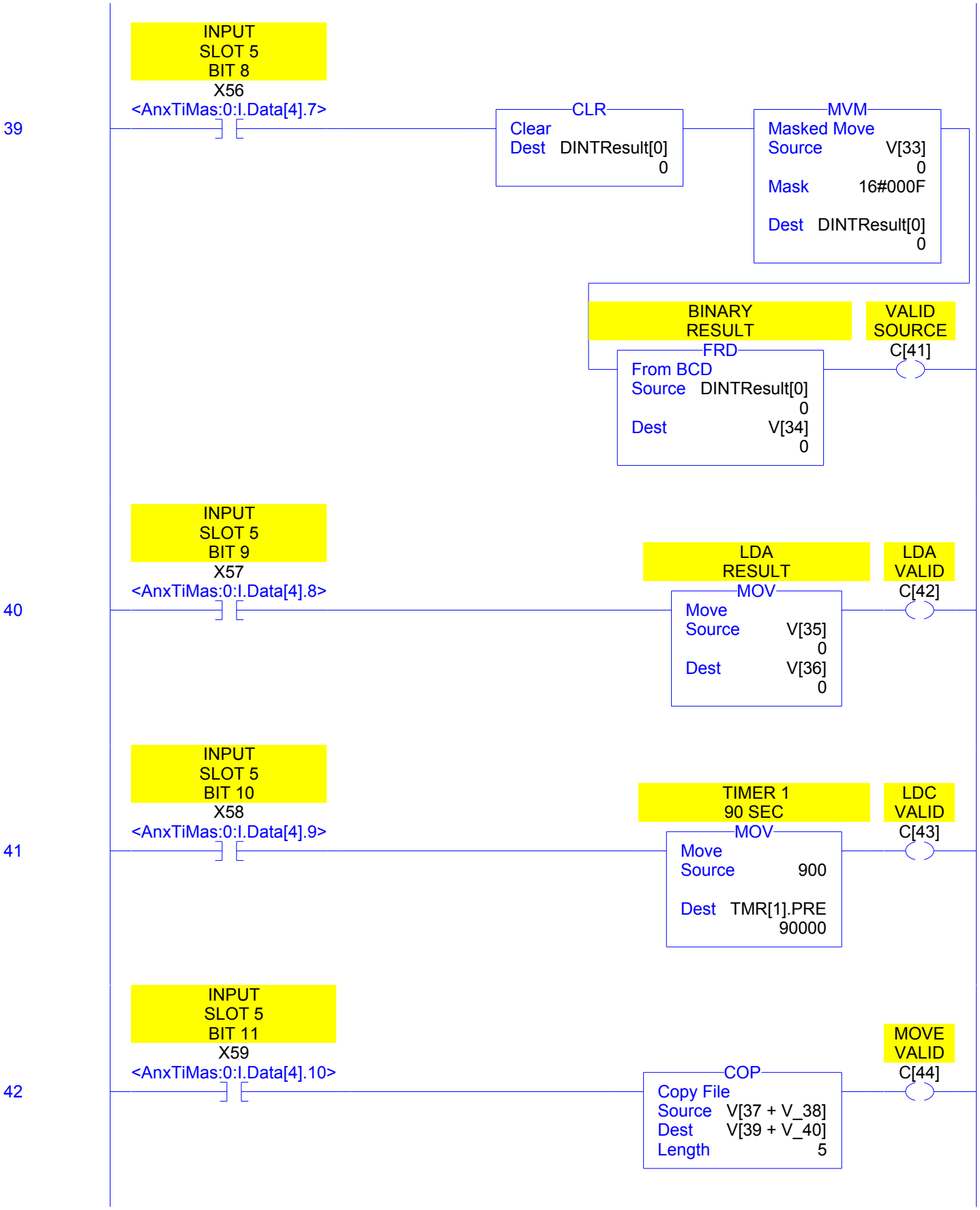

COMPARE
ANALOG IN
SLOT 11
WORD 1
EQU
Equal
Source A WX105
<AnxTiMas:0:I.Data[14]>
0
Source B WY1
<AnxTiMas:0:O.Data[1]>
0
EQUAL
C[10]
22
ANALOG IN
SLOT 11
WORD 2
NEQ
Not Equal
Source A WX106
<AnxTiMas:0:I.Data[15]>
0
Source B WY2
<AnxTiMas:0:O.Data[2]>
0
NOT EQUAL
C[11]
23
ANALOG IN
SLOT 11
WORD 3
LES
Less Than (A<B)
Source A WX107
<AnxTiMas:0:I.Data[16]>
0
Source B WY3
<AnxTiMas:0:O.Data[3]>
0
LESS THAN
C[12]
24
ANALOG IN
SLOT 11
WORD 4
GEQ
Grtr Than or Eql (A>=B)
Source A WX108
<AnxTiMas:0:I.Data[17]>
0
Source B WY4
<AnxTiMas:0:O.Data[4]>
0
GREATER
THAN OR
EQUAL
C[13]
25

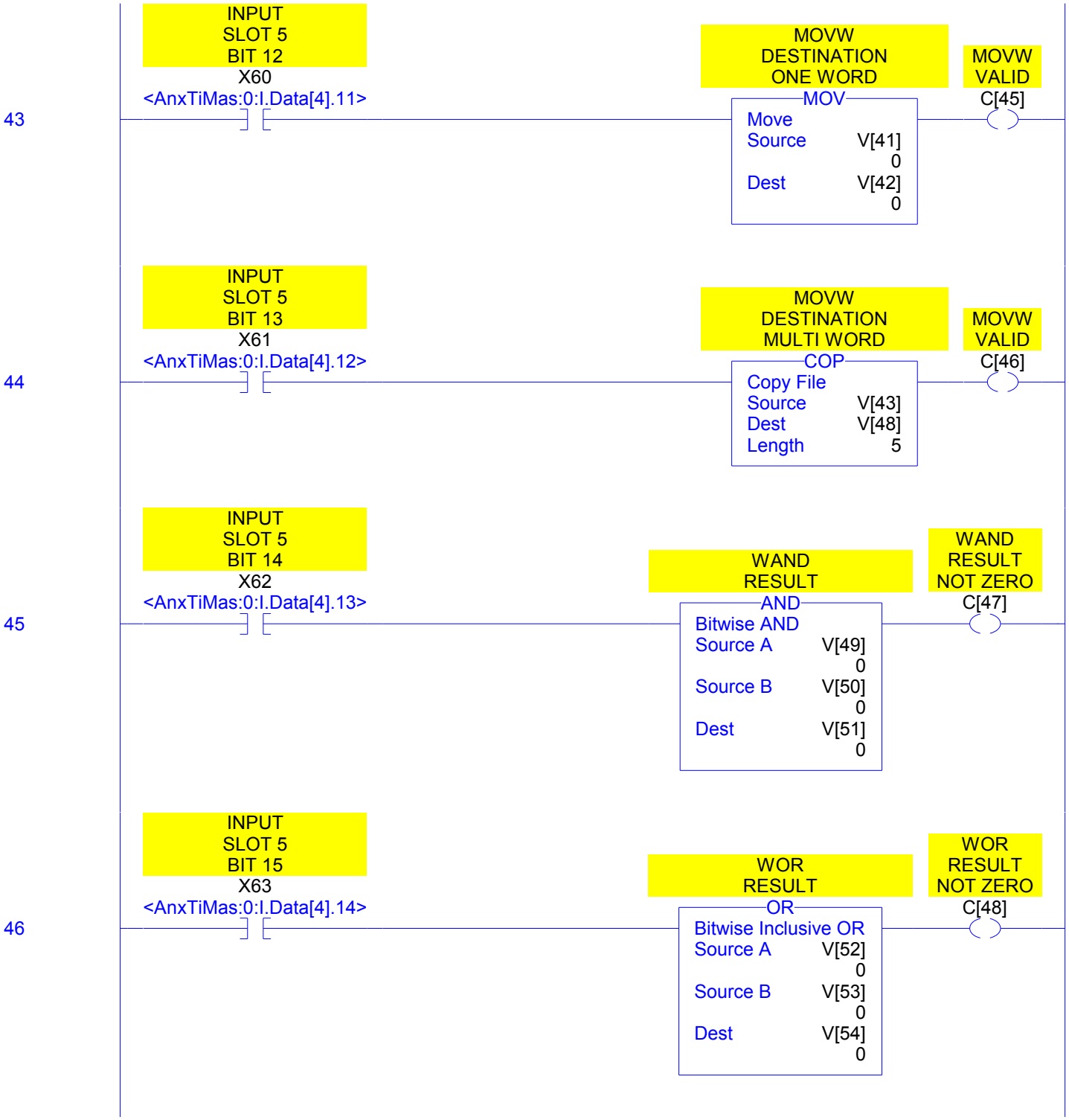

ANALOG IN
SLOT 11
WORD 5
GRT
Greater Than (A>B)
Source A WX109
<AnxTiMas:0:I.Data[18]>
0
Source B WY5
<AnxTiMas:0:O.Data[5]>
0
GREATER THAN
C[14]
26
ANALOG IN
SLOT 11
WORD 6
LEQ
Less Than or Eql (A<=B)
Source A WX110
<AnxTiMas:0:I.Data[19]>
0
Source B WY6
<AnxTiMas:0:O.Data[6]>
0
LESS THAN OR EQUAL
C[15]
27
INPUT
SLOT 4
BIT 13
X45
<AnxTiMas:0:I.Data[3].12>
CMP A
EQU
Equal
Source A V[1]
11
Source B V[2]
12
CMP A=B
C[16]
28
INPUT
SLOT 4
BIT 14
X46
<AnxTiMas:0:I.Data[3].13>
CMP A
EQU
Equal
Source A V[3]
13
Source B V[4]
14
CMP A=B
C[18]
29
CMP A
LES
Less Than (A<B)
Source A V[3]
13
Source B V[4]
14
CMP A<B
C[17]
AFI

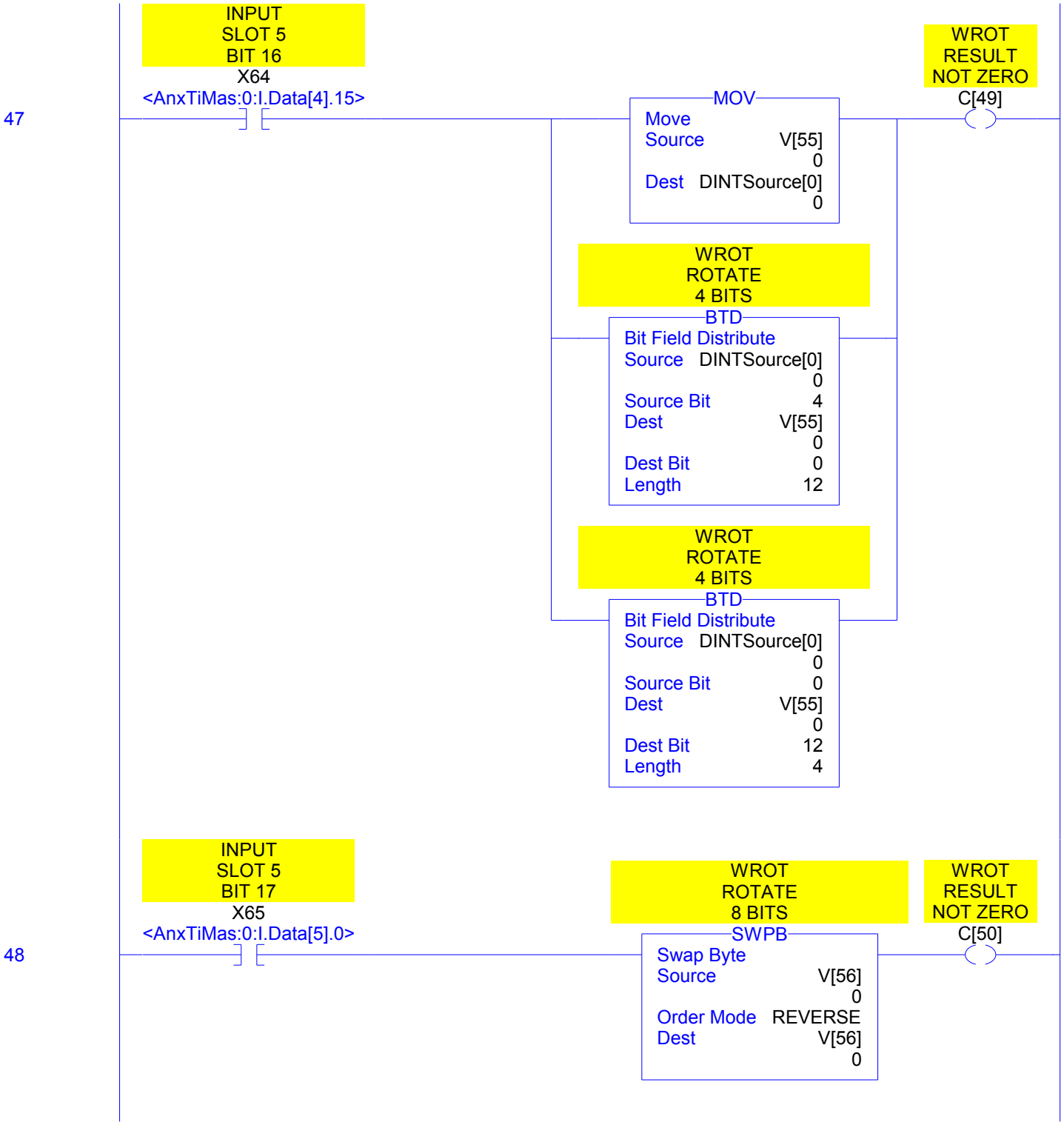

30
INPUT
SLOT 4
BIT 15
X47
<AnxTiMas:0:I.Data[3].14>
CMP A
EQU
Equal
Source A V[5] 15
Source B V[6] 16
CMP EQUAL BIT
C[20]
CMP A
GRT
Greater Than (A>B)
Source A V[5] 15
Source B V[6] 16
CMP A>B
C[19]
AFI
31
INPUT
SLOT 4
BIT 16
X48
<AnxTiMas:0:I.Data[3].15>
CMP B & A
EQU
Equal
Source A V[6] 16
Source B V[7] 17
CMP EQUAL
C[23]
CMP B & A
LES
Less Than (A<B)
Source A V[6] 16
Source B V[7] 17
CMP A<B
C[21]
AFI
CMP B & A
GRT
Greater Than (A>B)
Source A V[6] 16
Source B V[7] 17
CMP A>B
C[22]
AFI

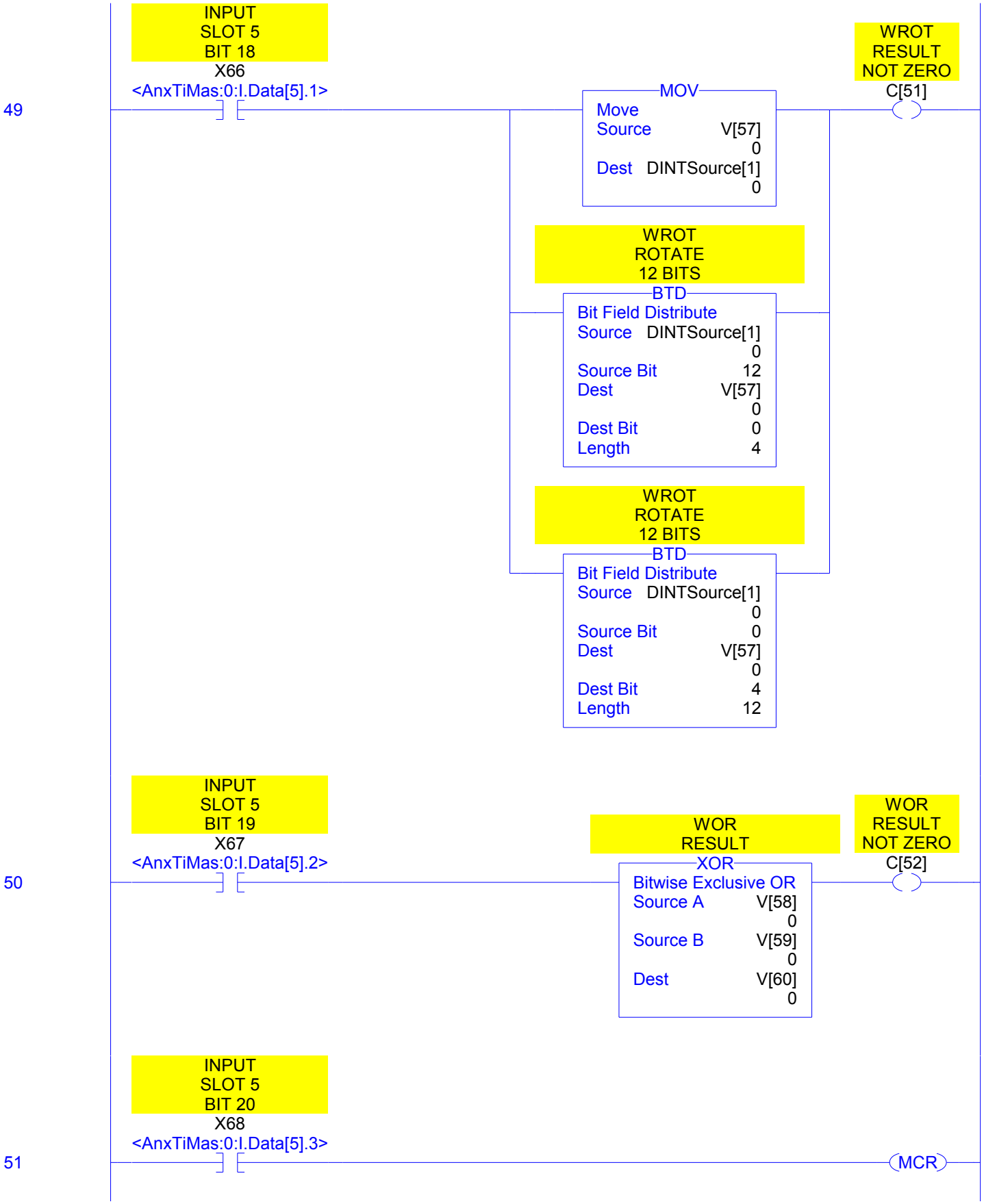

MATH
INPUT
SLOT 5
BIT 1
X49
<AnxTiMas:0:I.Data[4].0>
32
ABSV
A
ABS
Absolute Value
Source V[10] 20
Dest V[10] 20
ABSV
OVERFLOW
C[30]
INPUT
SLOT 5
BIT 2
X50
<AnxTiMas:0:I.Data[4].1>
33
ADD
SUM
ADD
Add
Source A V[11] 2
Source B V[12] 2
Dest V[13] 4
ADD
OVERFLOW
C[31]
INPUT
SLOT 5
BIT 3
X51
<AnxTiMas:0:I.Data[4].2>
34
SUB
RESULT
SUB
Subtract
Source A V[14] 5
Source B V[15] 3
Dest V[16] 2
SUB
OVERFLOW
C[32]
INPUT
SLOT 5
BIT 4
X52
<AnxTiMas:0:I.Data[4].3>
35
MULT
PRODUCT
MUL
Multiply
Source A V[17] 3
Source B V[18] 4
Dest dintV19 0
MULT
OUTPUT
C[33]

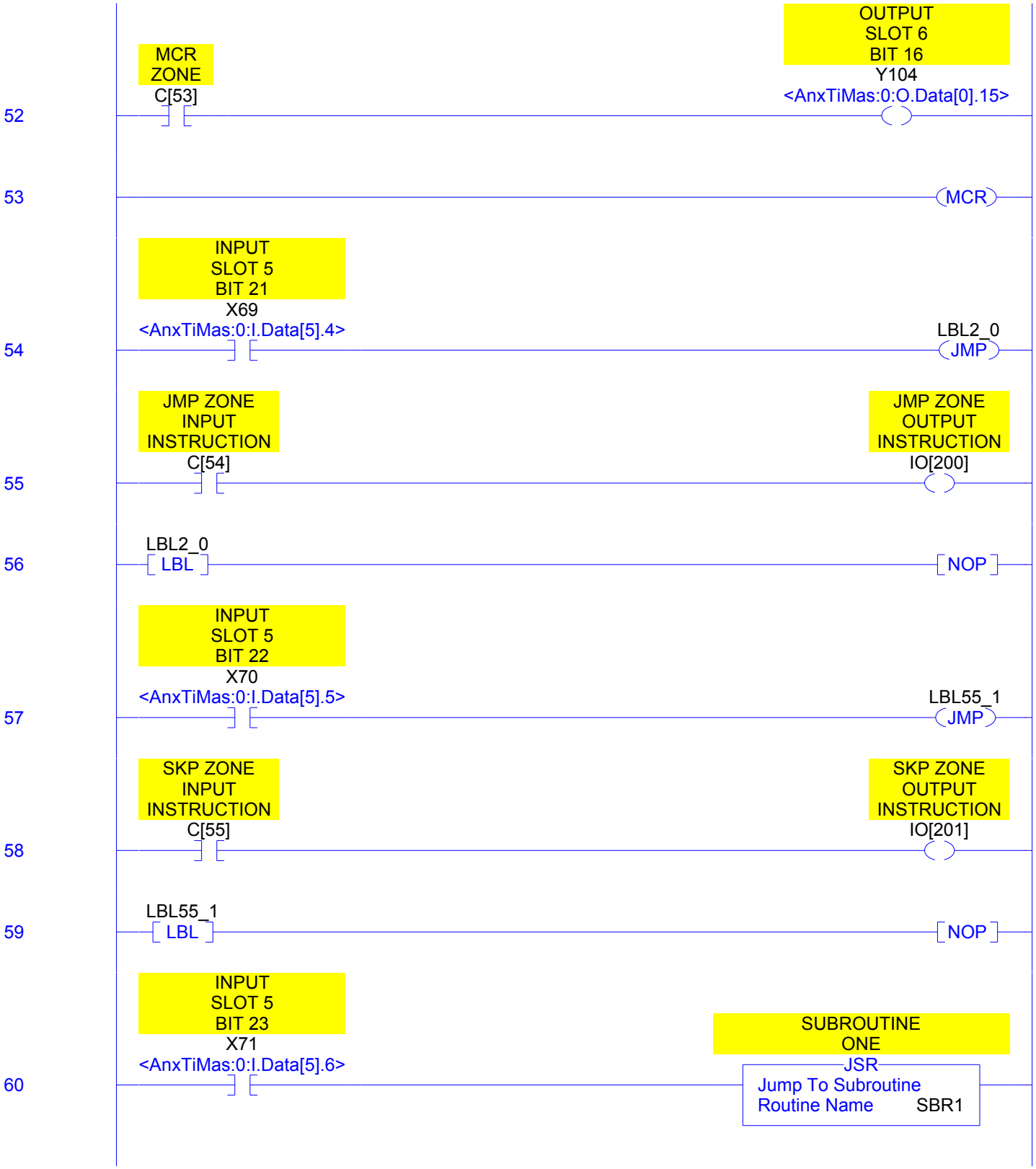

INPUT
SLOT 5
BIT 5
X53
<AnxTiMas:0:I.Data[4].4>
DIV
QUOTIENT
DIV
REMAINDER
DIV
1=VALID
0=DIV/ZERO
C[34]
36
DIV
Divide
Source A dintV19 0
Source B V[21] 2
Dest V[22] 6
MOD
Modulo
Source A dintV19 0
Source B V[21] 2
Dest V[23] 0
INPUT
SLOT 5
BIT 6
X54
<AnxTiMas:0:I.Data[4].5>
SQRT
RESULT
VALID
RESULT
C[35]
37
SQR
Square Root
Source dintV24 0
Dest V[26] 0
CONTERT/MOVE/LOGIC
INPUT
SLOT 5
BIT 7
X55
<AnxTiMas:0:I.Data[4].6>
BCD
RESULT
SOURCE
OUT OF
RANGE
C[40]
38
TOD
To BCD
Source V[30] 0
Dest V[31] 0

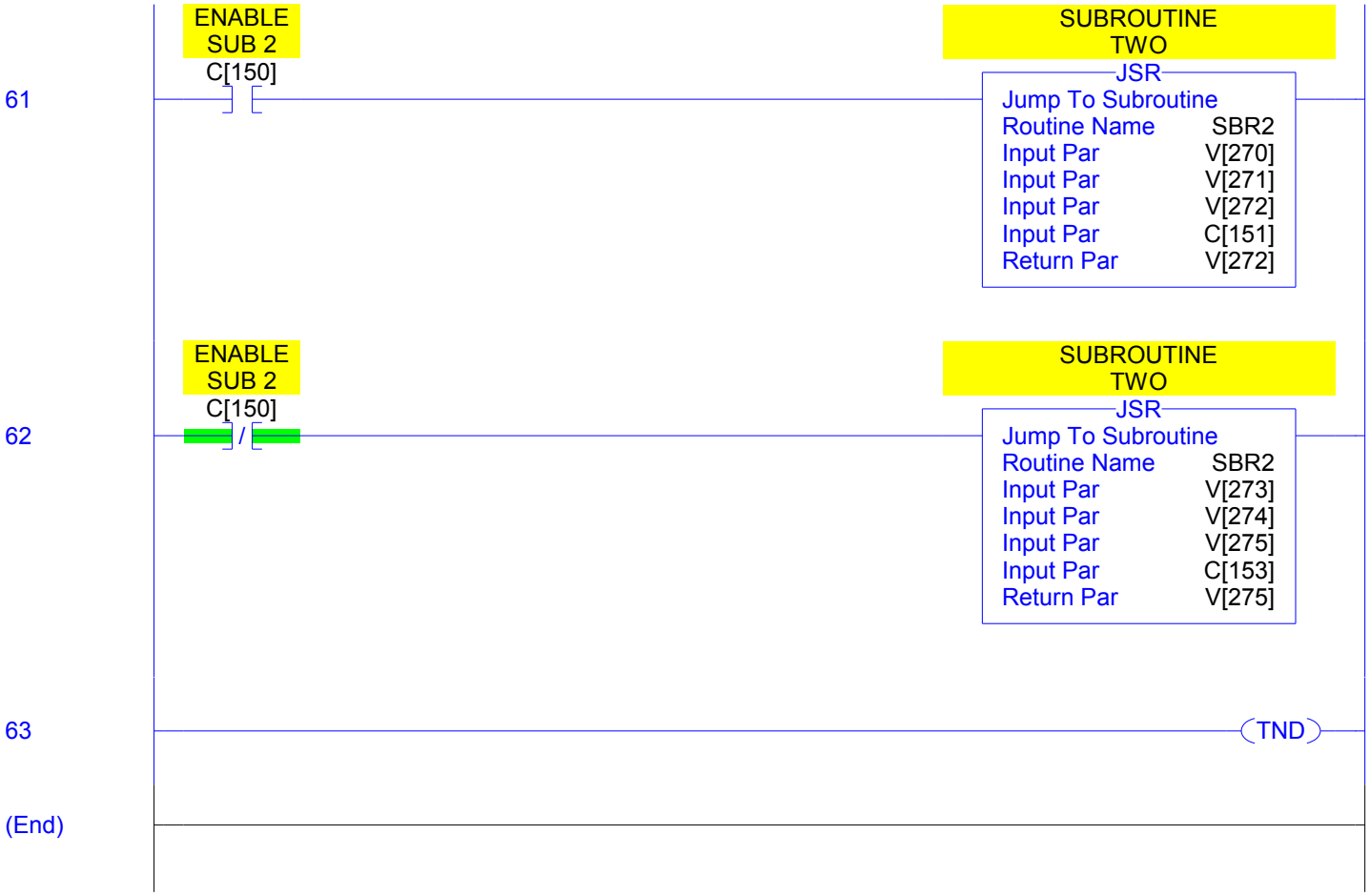

39
INPUT
SLOT 5
BIT 8
X56
<AnxTiMas:0:I.Data[4].7>
CLR
Clear
Dest DINTResult[0]
0
MVM
Masked Move
Source V[33]
0
Mask 16#000F
Dest DINTResult[0]
0
BINARY
RESULT
FRD
From BCD
Source DINTResult[0]
0
Dest V[34]
0
VALID
SOURCE
C[41]
40
INPUT
SLOT 5
BIT 9
X57
<AnxTiMas:0:I.Data[4].8>
LDA
RESULT
MOV
Move
Source V[35]
0
Dest V[36]
0
LDA
VALID
C[42]
41
INPUT
SLOT 5
BIT 10
X58
<AnxTiMas:0:I.Data[4].9>
TIMER 1
90 SEC
MOV
Move
Source 900
Dest TMR[1].PRE
90000
LDC
VALID
C[43]
42
INPUT
SLOT 5
BIT 11
X59
<AnxTiMas:0:I.Data[4].10>
COP
Copy File
Source V[37 + V_38]
Dest V[39 + V_40]
Length 5
MOVE
VALID
C[44]

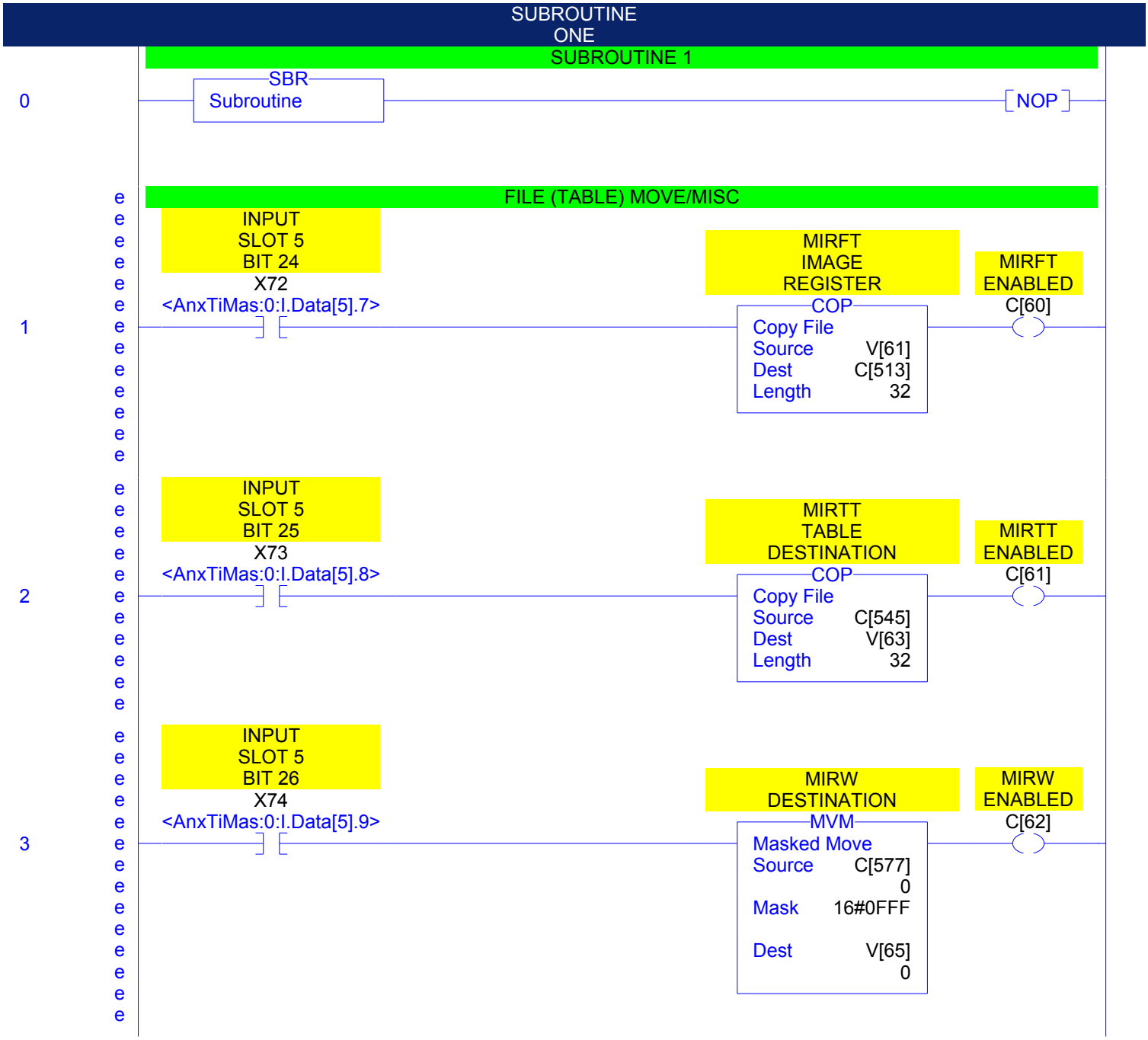

43
INPUT
SLOT 5
BIT 12
X60
<AnxTiMas:0:I.Data[4].11>
MOVW
DESTINATION
ONE WORD
MOV
Move
Source V[41]
0
Dest V[42]
0
MOVW
VALID
C[45]
44
INPUT
SLOT 5
BIT 13
X61
<AnxTiMas:0:I.Data[4].12>
MOVW
DESTINATION
MULTI WORD
COP
Copy File
Source V[43]
Dest V[48]
Length 5
MOVW
VALID
C[46]
45
INPUT
SLOT 5
BIT 14
X62
<AnxTiMas:0:I.Data[4].13>
WAND
RESULT
AND
Bitwise AND
Source A V[49]
0
Source B V[50]
0
Dest V[51]
0
WAND
RESULT
NOT ZERO
C[47]
46
INPUT
SLOT 5
BIT 15
X63
<AnxTiMas:0:I.Data[4].14>
WOR
RESULT
OR
Bitwise Inclusive OR
Source A V[52]
0
Source B V[53]
0
Dest V[54]
0
WOR
RESULT
NOT ZERO
C[48]


47
INPUT
SLOT 5
BIT 16
X64
<AnxTiMas:0:I.Data[4].15>
MOV
Move
Source V[55]
0
Dest DINTSource[0]
0
WROT
RESULT
NOT ZERO
C[49]
WROT
ROTATE
4 BITS
BTD
Bit Field Distribute
Source DINTSource[0]
0
Source Bit 4
Dest V[55]
0
Dest Bit 0
Length 12
WROT
ROTATE
4 BITS
BTD
Bit Field Distribute
Source DINTSource[0]
0
Source Bit 0
Dest V[55]
0
Dest Bit 12
Length 4
48
INPUT
SLOT 5
BIT 17
X65
<AnxTiMas:0:I.Data[5].0>
WROT
ROTATE
8 BITS
SWPB
Swap Byte
Source V[56]
0
Order Mode REVERSE
Dest V[56]
0
WROT
RESULT
NOT ZERO
C[50]

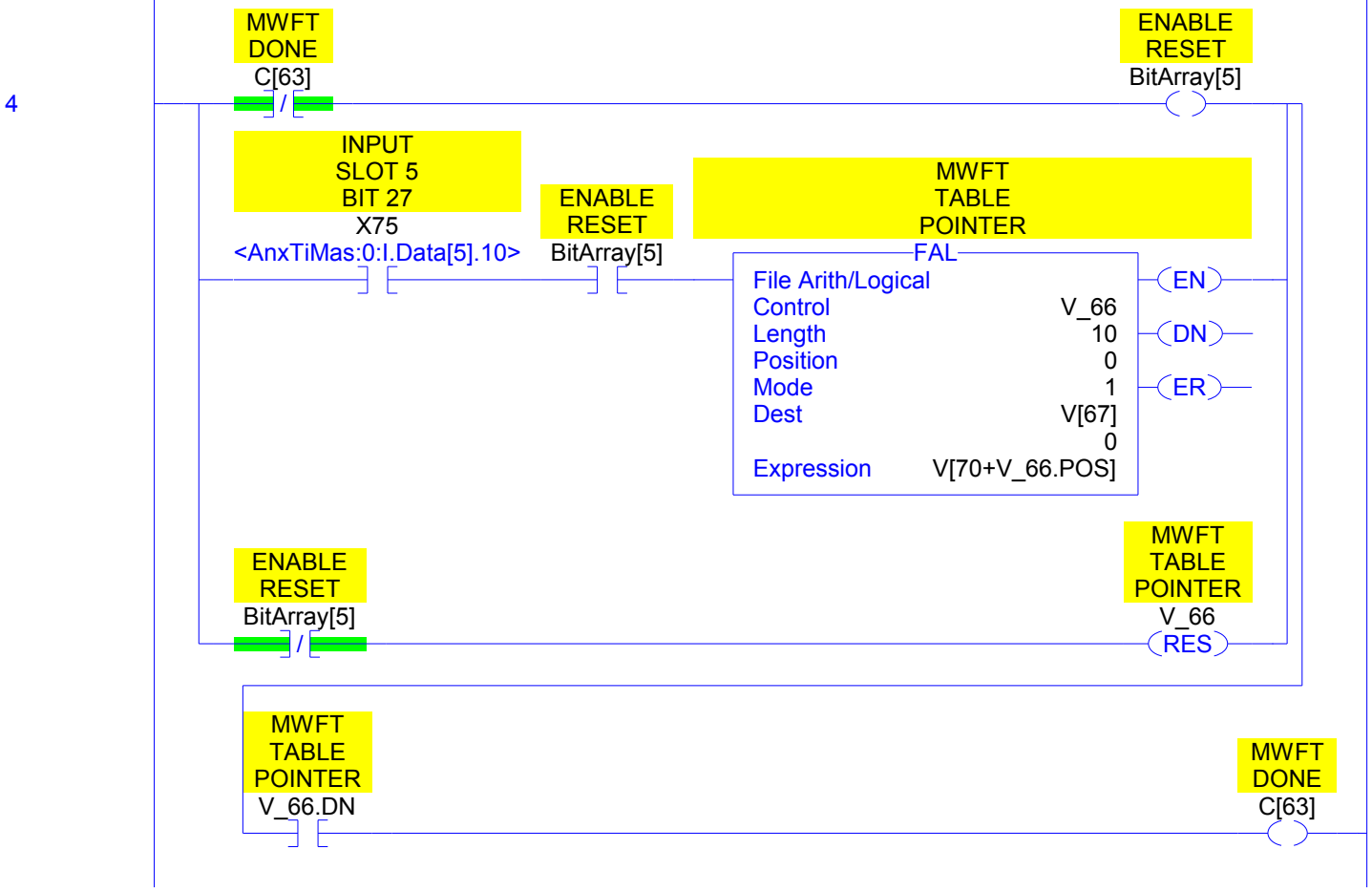

INPUT
SLOT 5
BIT 18
X66
<AnxTiMas:0:I.Data[5].1>
49
MOV
Move
Source V[57]
0
Dest DINTSource[1]
0
WROT
RESULT
NOT ZERO
C[51]
WROT
ROTATE
12 BITS
BTD
Bit Field Distribute
Source DINTSource[1]
0
Source Bit 12
Dest V[57]
0
Dest Bit 0
Length 4
WROT
ROTATE
12 BITS
BTD
Bit Field Distribute
Source DINTSource[1]
0
Source Bit 0
Dest V[57]
0
Dest Bit 4
Length 12
INPUT
SLOT 5
BIT 19
X67
<AnxTiMas:0:I.Data[5].2>
50
WOR
RESULT
XOR
Bitwise Exclusive OR
Source A V[58]
0
Source B V[59]
0
Dest V[60]
0
WOR
RESULT
NOT ZERO
C[52]
INPUT
SLOT 5
BIT 20
X68
<AnxTiMas:0:I.Data[5].3>
51
MCR


MCR ZONE
C[53]
OUTPUT SLOT 6 BIT 16
Y104
<AnxTiMas:0:O.Data[0].15>
52
53
MCR
INPUT SLOT 5 BIT 21
X69
<AnxTiMas:0:I.Data[5].4>
LBL2_0
JMP
54
JMP ZONE INPUT INSTRUCTION
C[54]
JMP ZONE OUTPUT INSTRUCTION
IO[200]
55
LBL2_0
LBL
NOP
56
INPUT SLOT 5 BIT 22
X70
<AnxTiMas:0:I.Data[5].5>
LBL55_1
JMP
57
SKP ZONE INPUT INSTRUCTION
C[55]
SKP ZONE OUTPUT INSTRUCTION
IO[201]
58
LBL55_1
LBL
NOP
59
INPUT SLOT 5 BIT 23
X71
<AnxTiMas:0:I.Data[5].6>
SUBROUTINE ONE
JSR
Jump To Subroutine
Routine Name SBR1
60

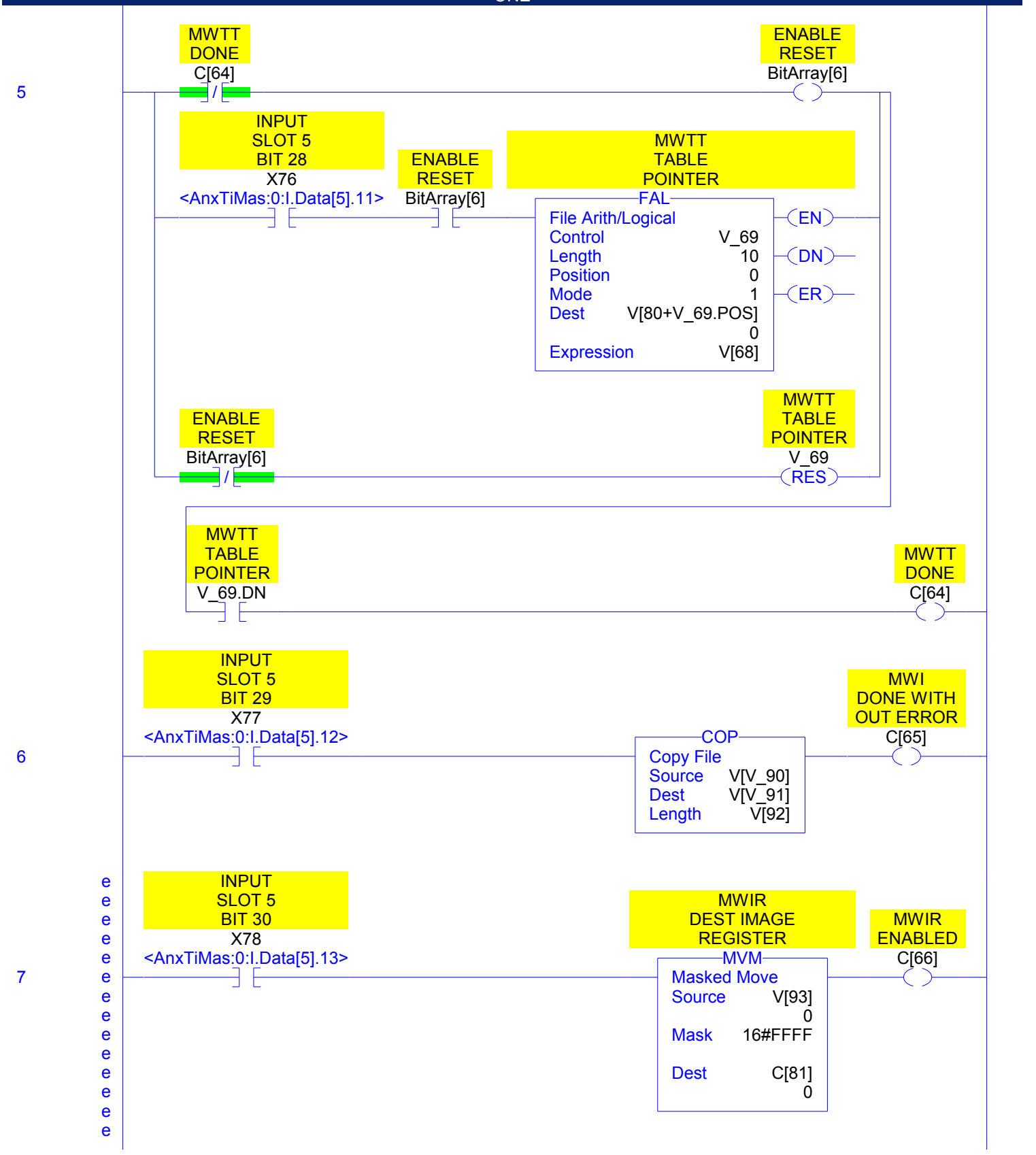

61
ENABLE SUB 2
C[150]
SUBROUTINE TWO
JSR
Jump To Subroutine
Routine Name SBR2
Input Par V[270]
Input Par V[271]
Input Par V[272]
Input Par C[151]
Return Par V[272]
62
ENABLE SUB 2
C[150]
/
SUBROUTINE TWO
JSR
Jump To Subroutine
Routine Name SBR2
Input Par V[273]
Input Par V[274]
Input Par V[275]
Input Par C[153]
Return Par V[275]
63
TND
(End)


SUBROUTINE
ONE
SUBROUTINE 1
0
SBR
Subroutine
NOP
FILE (TABLE) MOVE/MISC
1
INPUT
SLOT 5
BIT 24
X72
<AnxTiMas:0:I.Data[5].7>
MIRFT
IMAGE
REGISTER
COP
Copy File
Source V[61]
Dest C[513]
Length 32
MIRFT
ENABLED
C[60]
2
INPUT
SLOT 5
BIT 25
X73
<AnxTiMas:0:I.Data[5].8>
MIRTT
TABLE
DESTINATION
COP
Copy File
Source C[545]
Dest V[63]
Length 32
MIRTT
ENABLED
C[61]
3
INPUT
SLOT 5
BIT 26
X74
<AnxTiMas:0:I.Data[5].9>
MIRW
DESTINATION
MVM
Masked Move
Source C[577]
0
Mask 16#0FFF
Dest V[65]
0
MIRW
ENABLED
C[62]

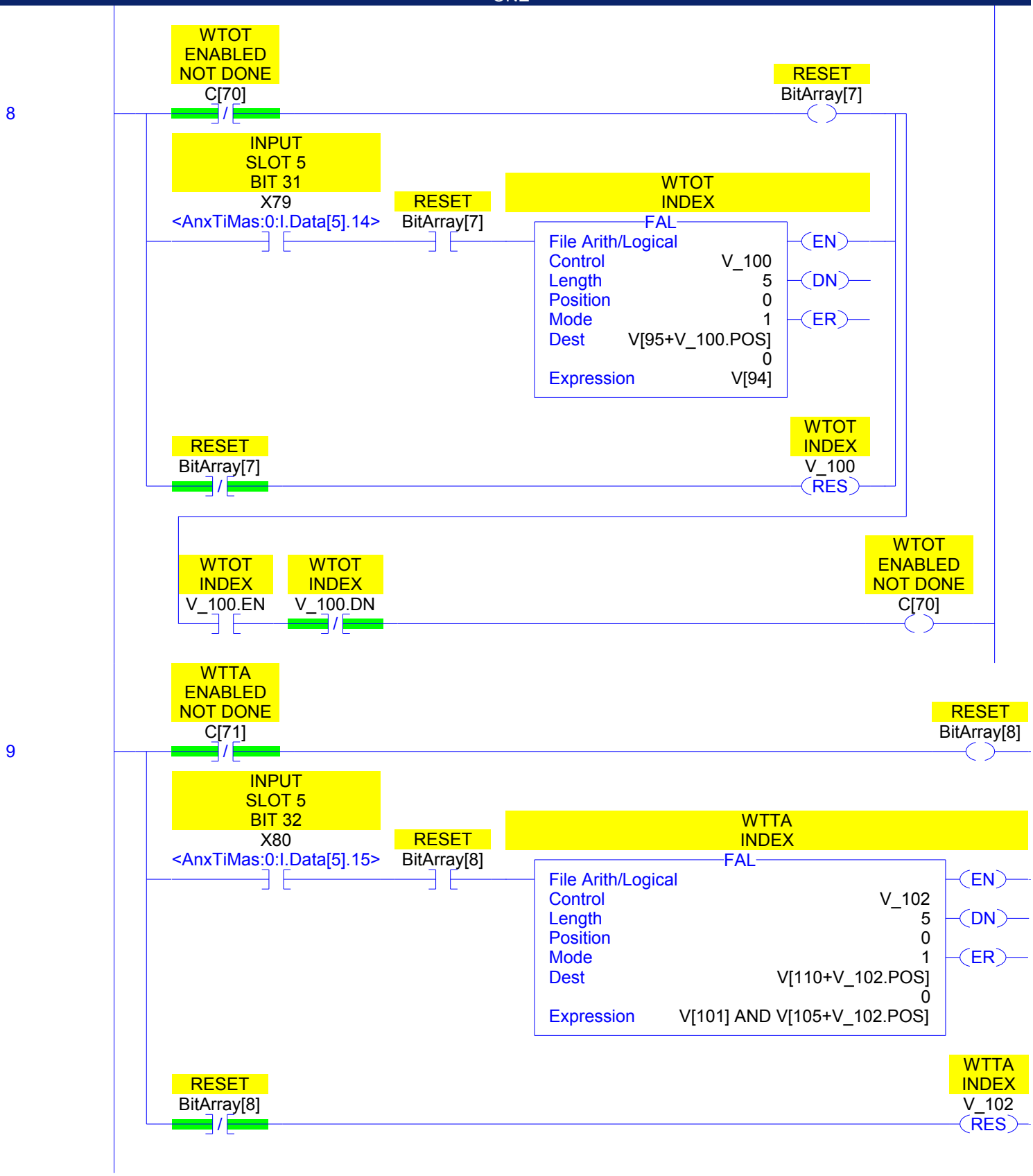
SUBROUTINE
ONE



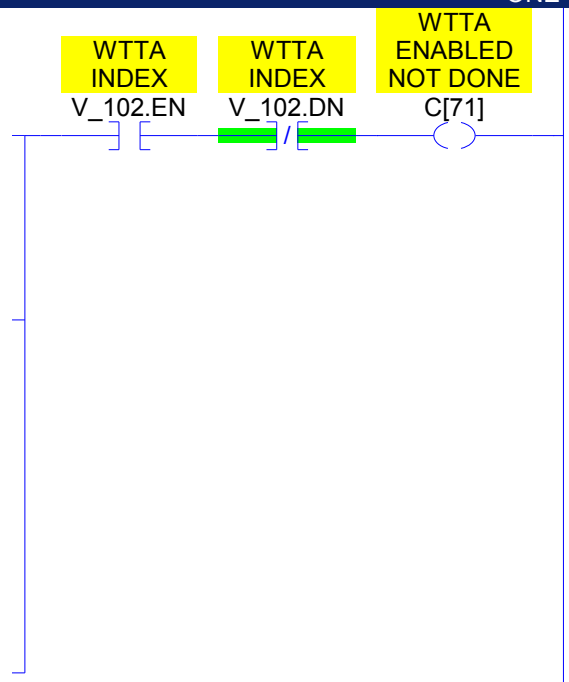
SUBROUTINE
ONE

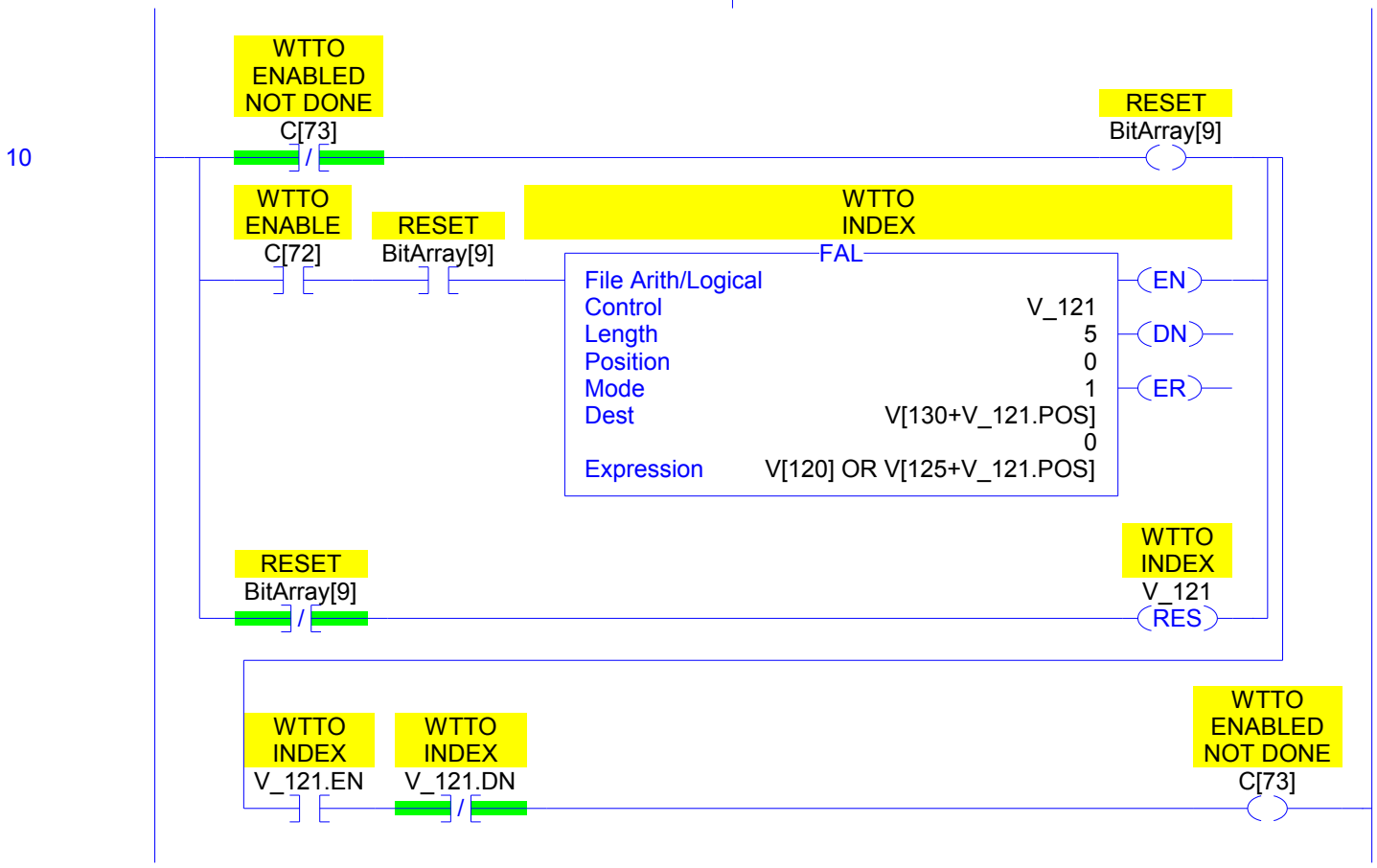


SUBROUTINE
ONE

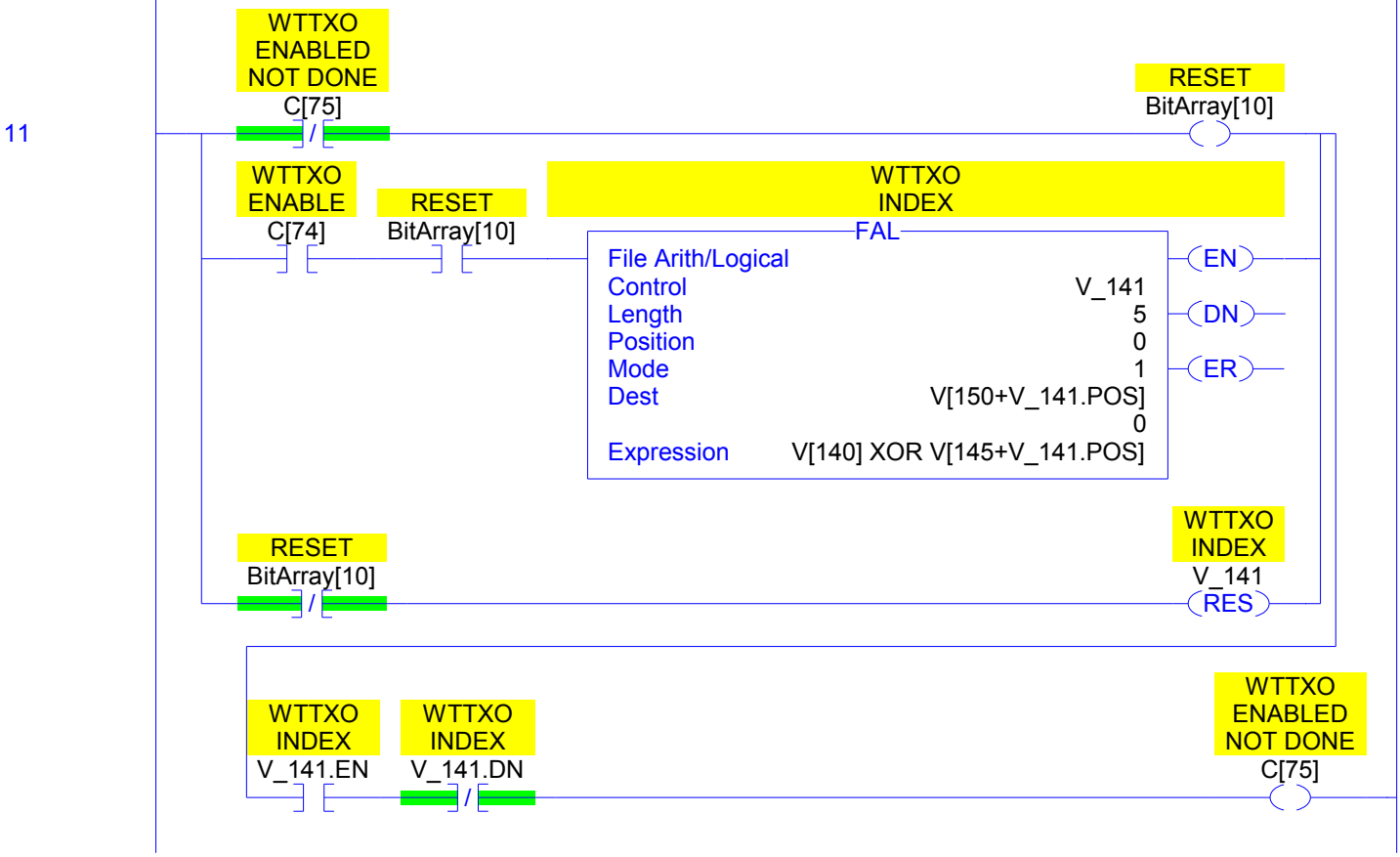


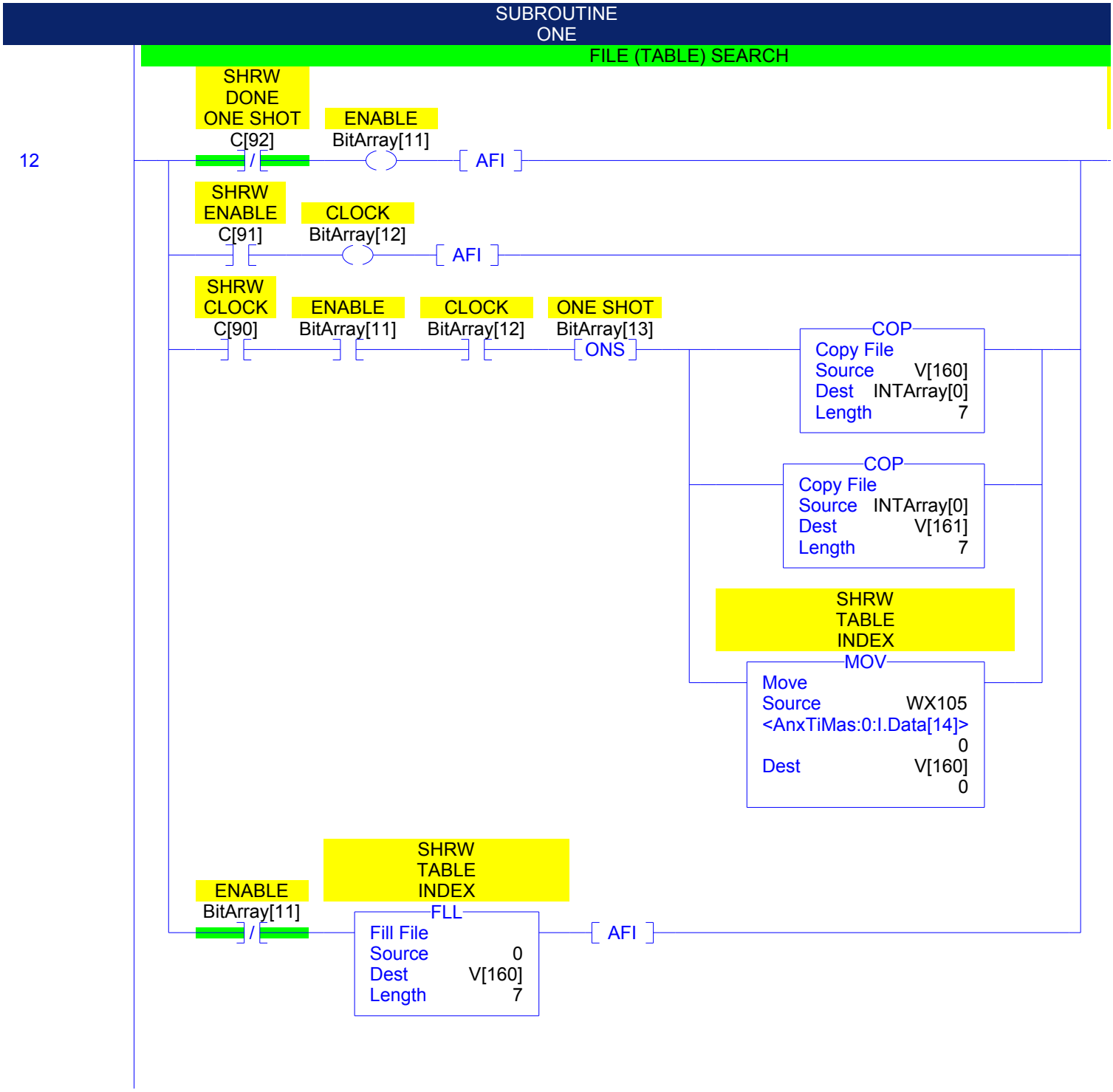
SUBROUTINE
ONE





SUBROUTINE
ONE

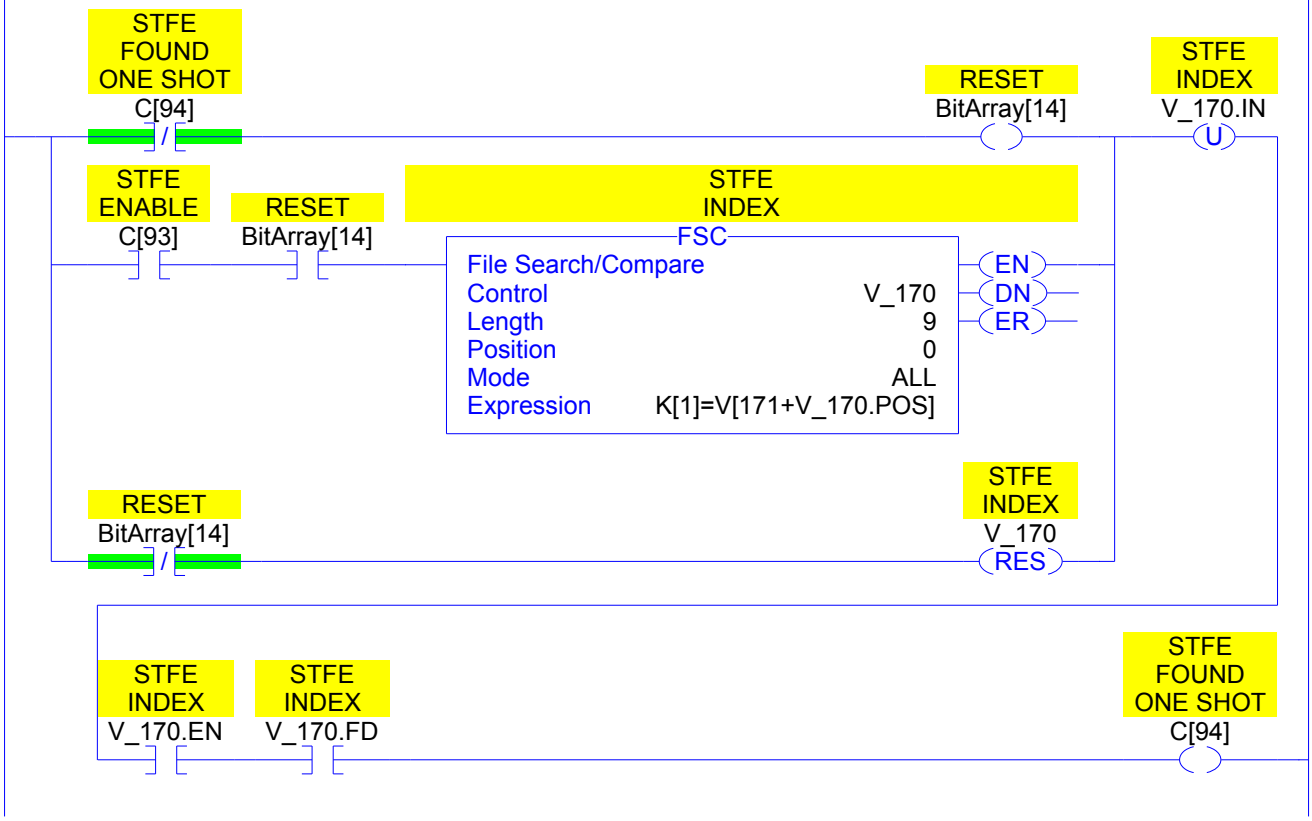



SUBROUTINE
ONE
FILE (TABLE) SEARCH
12
SHRW
DONE
ONE SHOT
C[92]
ENABLE
BitArray[11]
AFI
SHRW
ENABLE
C[91]
CLOCK
BitArray[12]
AFI
SHRW
CLOCK
C[90]
ENABLE
BitArray[11]
CLOCK
BitArray[12]
ONE SHOT
BitArray[13]
ONS
COP
Copy File
Source V[160]
Dest INTArray[0]
Length 7
COP
Copy File
Source INTArray[0]
Dest V[161]
Length 7
SHRW
TABLE
INDEX
MOV
Move
Source WX105
<AnxTiMas:0:I.Data[14]>
0
Dest V[160]
0
SHRW
TABLE
INDEX
ENABLE
BitArray[11]
FLL
Fill File
Source 0
Dest V[160]
Length 7
AFI

SUBROUTINE
ONE

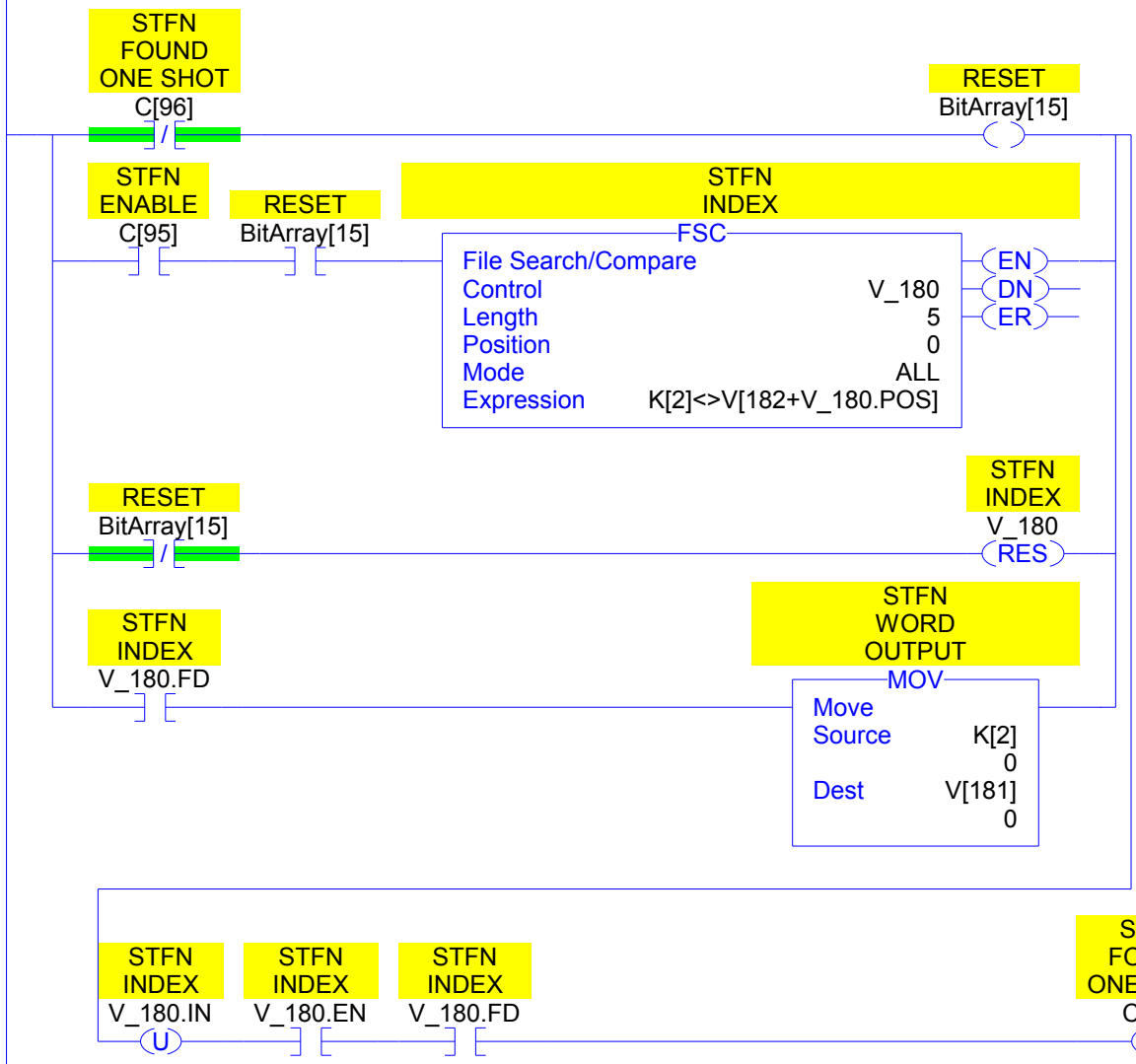
SUBROUTINE
ONE

13



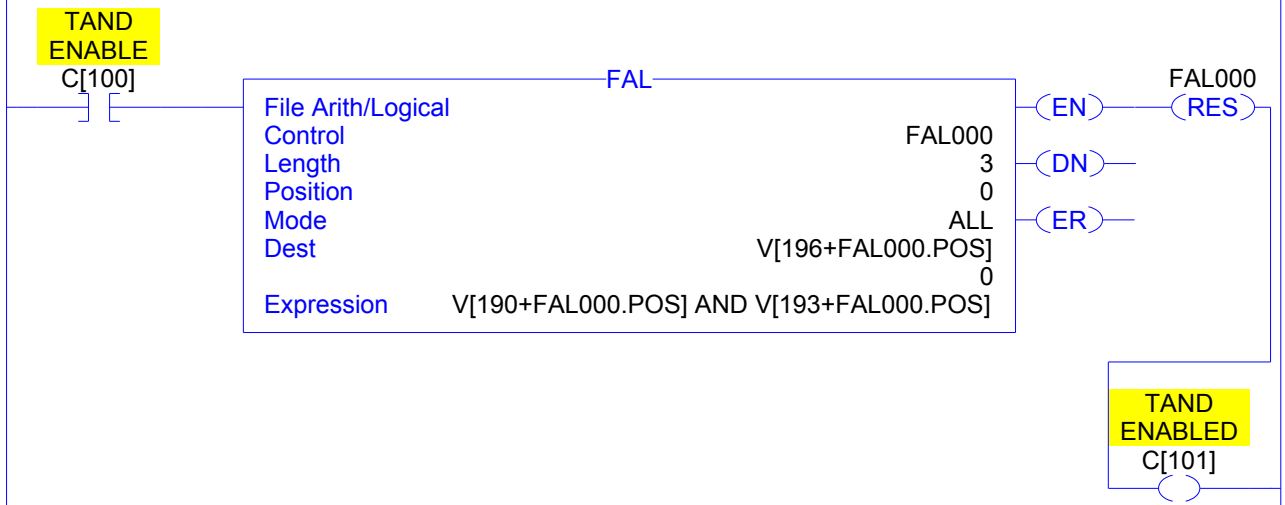
SUBROUTINE
ONE

14



FILE (TABLE) LOGIC/MOVE

15



SUBROUTINE
ONE

16

TCPL ENABLE
C[102]

FAL

File Arith/Logical	
Control	FAL001
Length	3
Position	0
Mode	ALL
Dest	V[203+FAL001.POS]
	0
Expression	NOT V[200+FAL001.POS]

EN
DN
ER

FAL001
RES

TCPL ENABLED
C[103]

17

TOR ENABLE
C[104]

FAL

File Arith/Logical	
Control	FAL002
Length	3
Position	0
Mode	ALL
Dest	V[216+FAL002.POS]
	0
Expression	V[210+FAL002.POS] OR V[213+FAL002.POS]

EN
DN
ER

FAL002
RES

TOR ENABLED
C[105]

18

TXOR ENABLE
C[106]

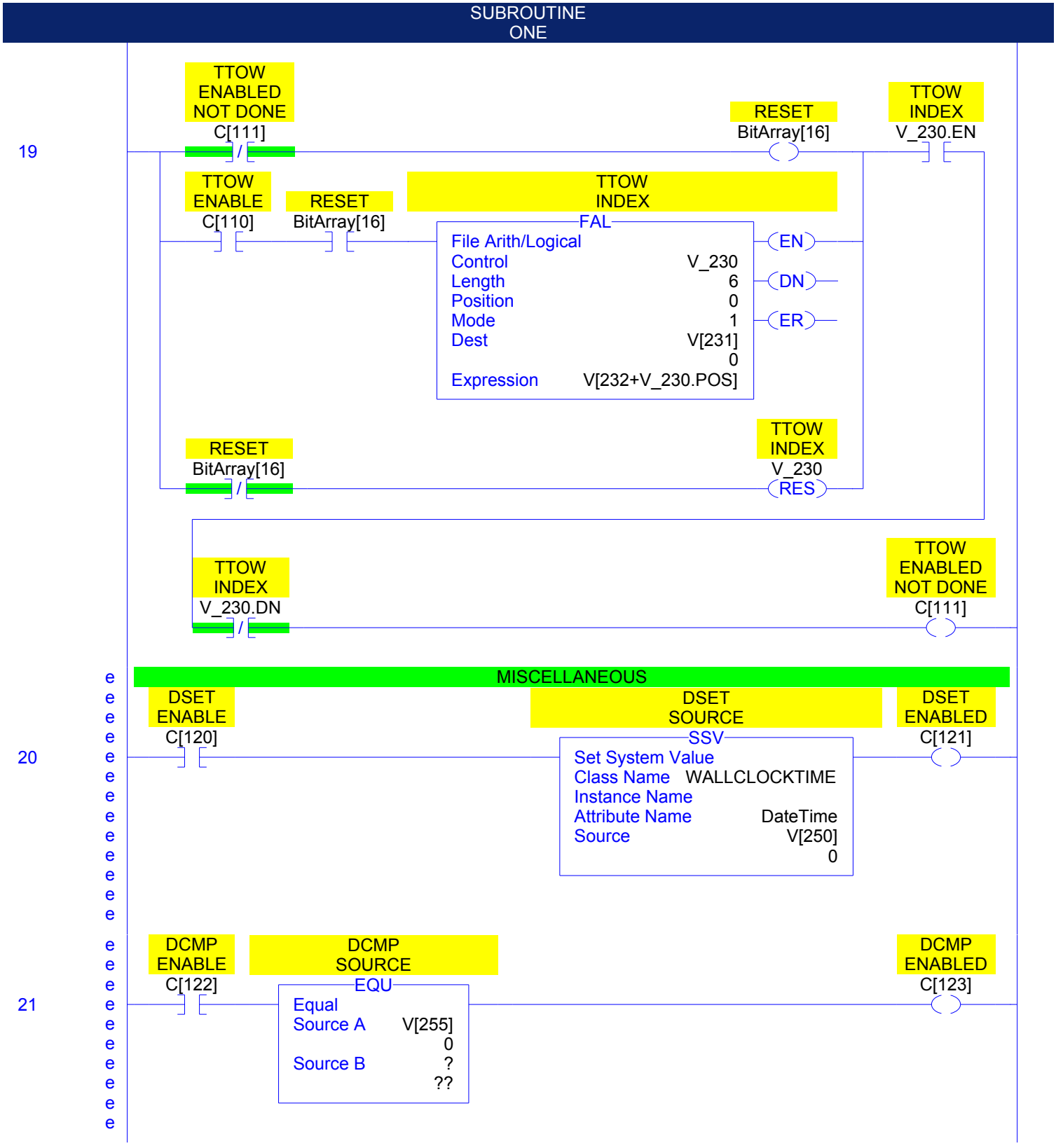
FAL

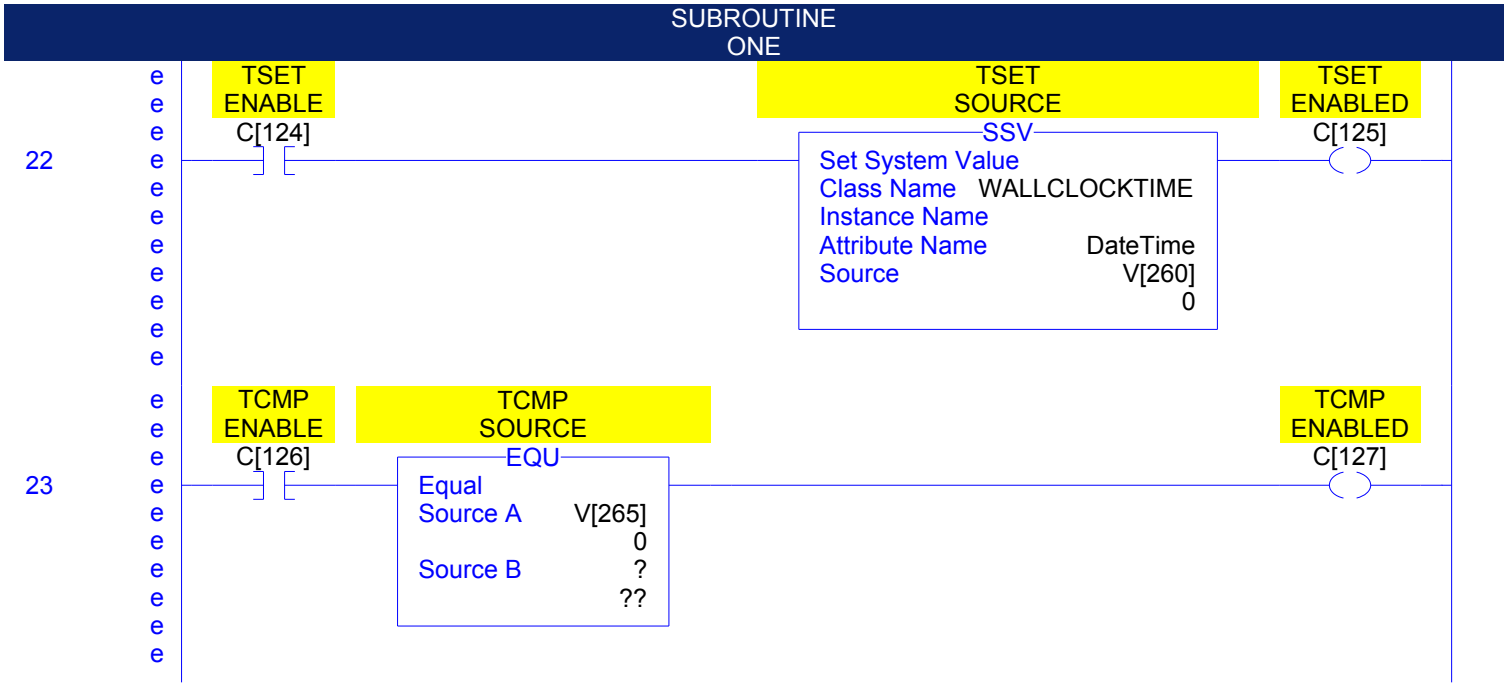
File Arith/Logical	
Control	FAL003
Length	3
Position	0
Mode	ALL
Dest	V[226+FAL003.POS]
	0
Expression	V[220+FAL003.POS] XOR V[223+FAL003.POS]

EN
DN
ER

FAL003
RES

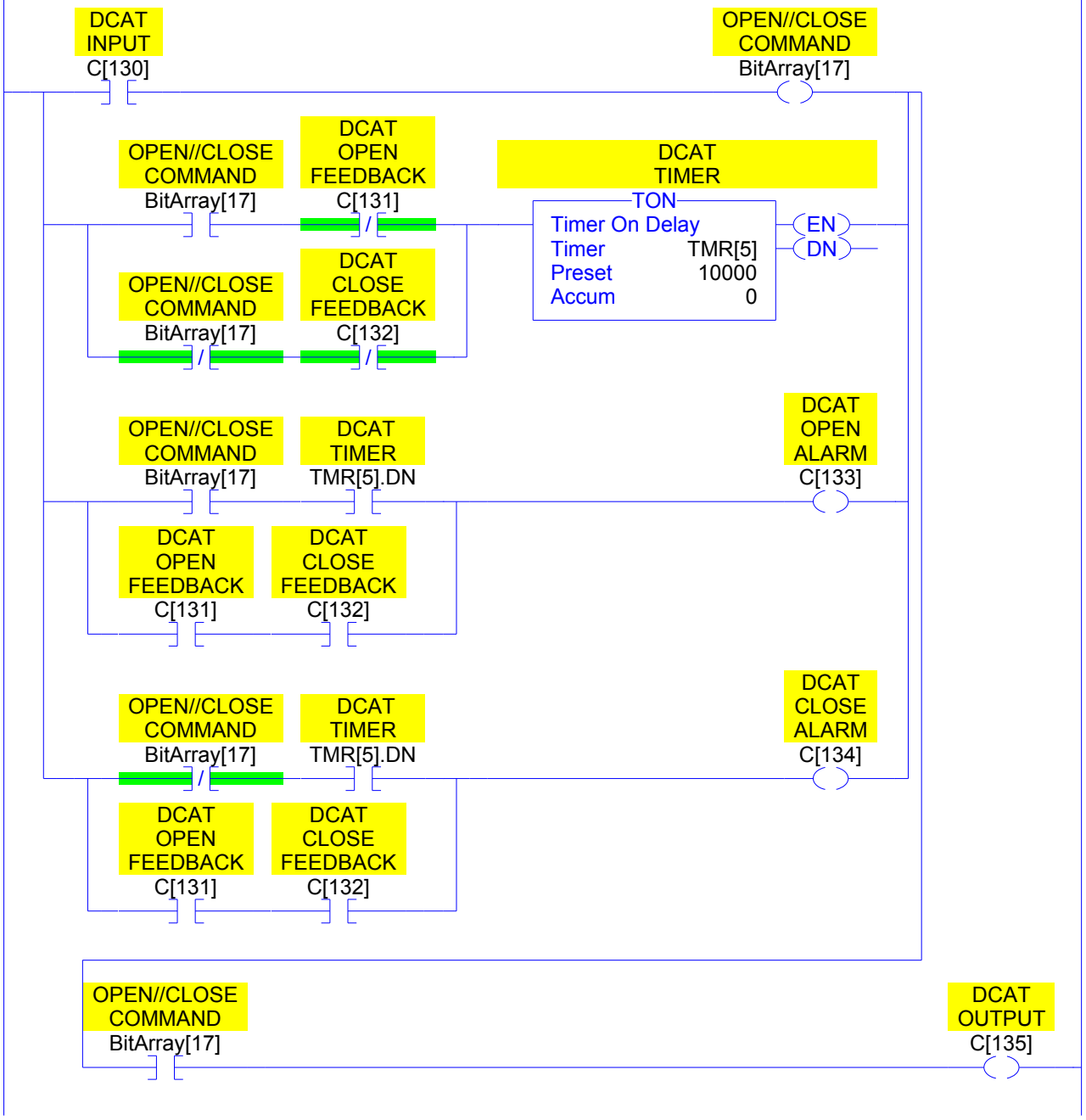
TXOR ENABLED
C[107]


SUBROUTINE
ONE
19
TTOW ENABLED NOT DONE
C[111]
RESET
BitArray[16]
TTOW INDEX
V_230.EN
TTOW ENABLE
C[110]
RESET
BitArray[16]
TTOW INDEX
FAL
File Arith/Logical
Control V_230
Length 6
Position 0
Mode 1
Dest V[231]
0
Expression V[232+V_230.POS]
EN
DN
ER
RESET
BitArray[16]
TTOW INDEX
V_230
RES
TTOW INDEX
V_230.DN
TTOW ENABLED NOT DONE
C[111]
MISCELLANEOUS
20
DSET ENABLE
C[120]
DSET SOURCE
SSV
Set System Value
Class Name WALLCLOCKTIME
Instance Name
Attribute Name DateTime
Source V[250]
0
DSET ENABLED
C[121]
21
DCMP ENABLE
C[122]
DCMP SOURCE
EQU
Equal
Source A V[255]
0
Source B ?
??
DCMP ENABLED
C[123]


SUBROUTINE
ONE
TSET
ENABLE
C[124]
TSET
SOURCE
SSV
Set System Value
Class Name WALLCLOCKTIME
Instance Name
Attribute Name DateTime
Source V[260]
0
TSET
ENABLED
C[125]
22
TCMP
ENABLE
C[126]
TCMP
SOURCE
EQU
Equal
Source A V[265]
0
Source B ?
??
TCMP
ENABLED
C[127]
23

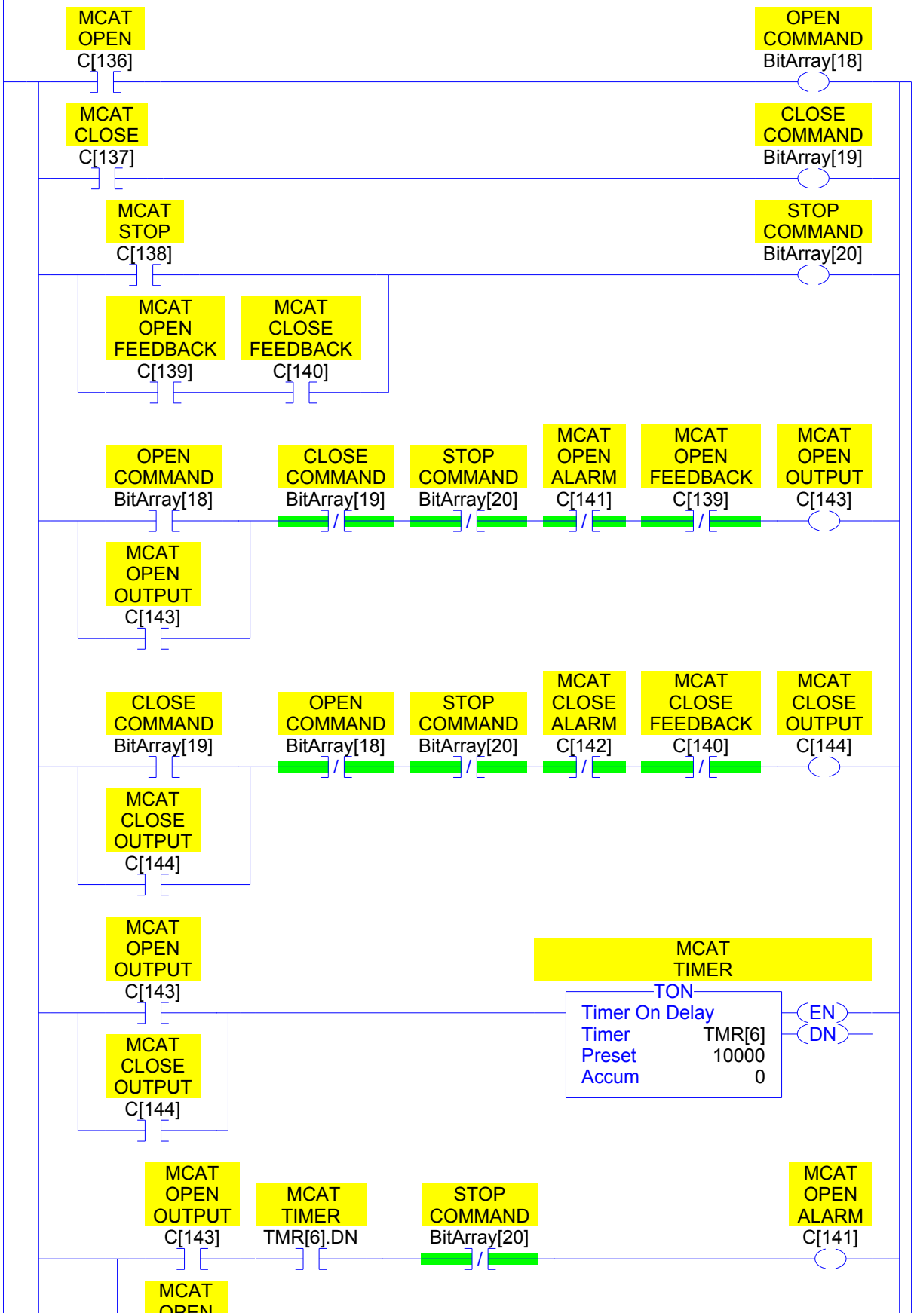
SUBROUTINE
ONE

24

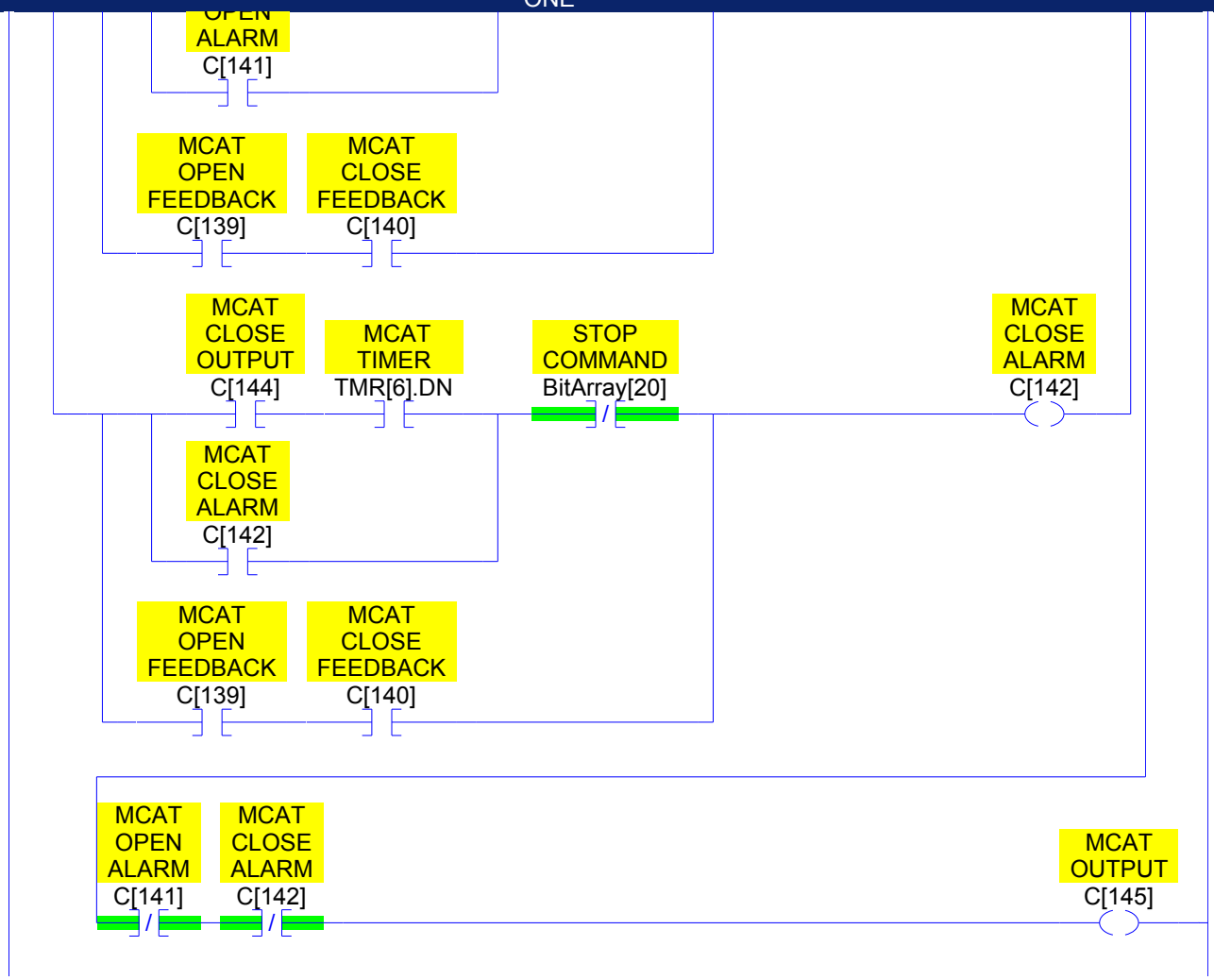


SUBROUTINE
ONE

25

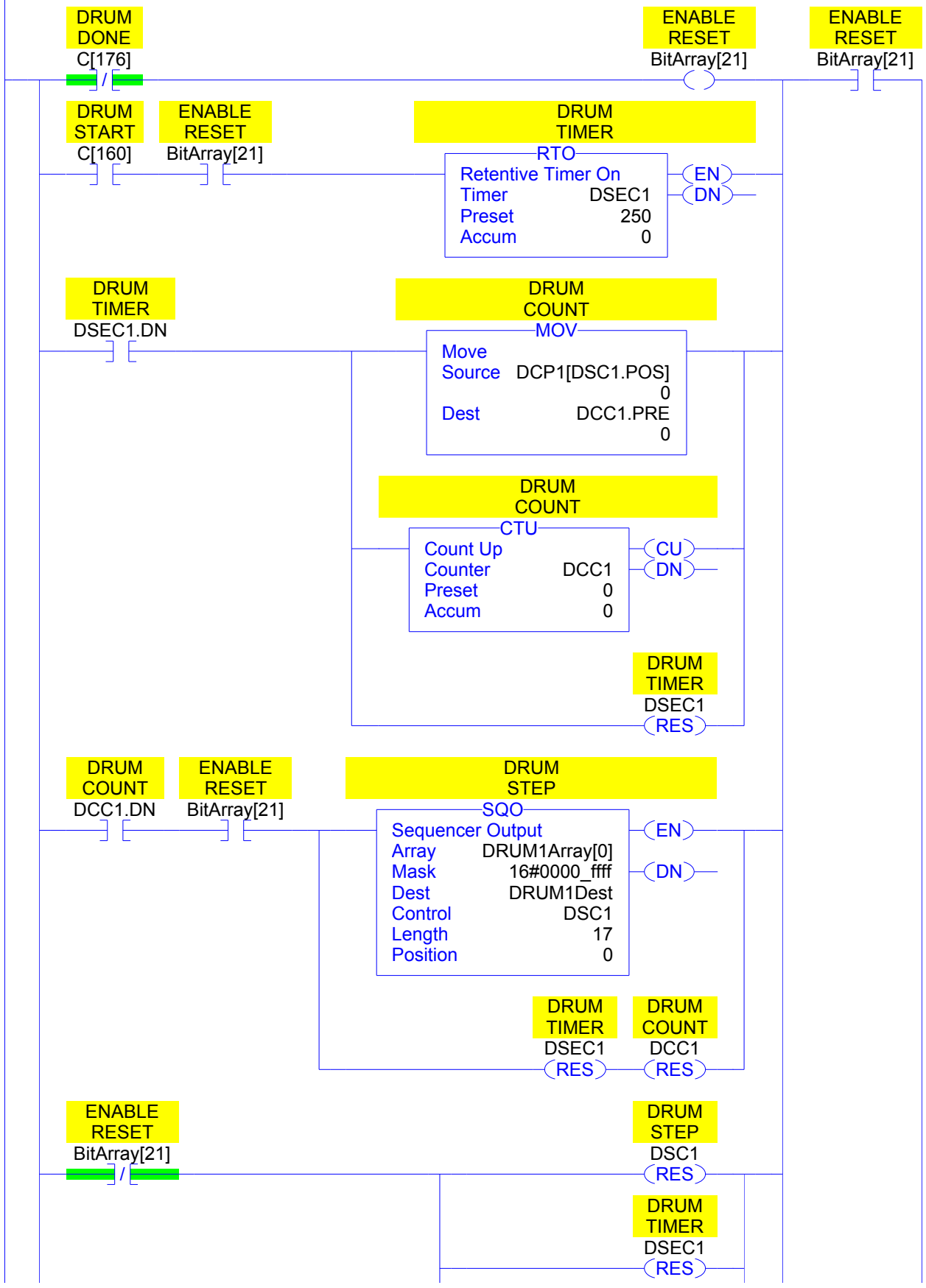


SUBROUTINE
ONE

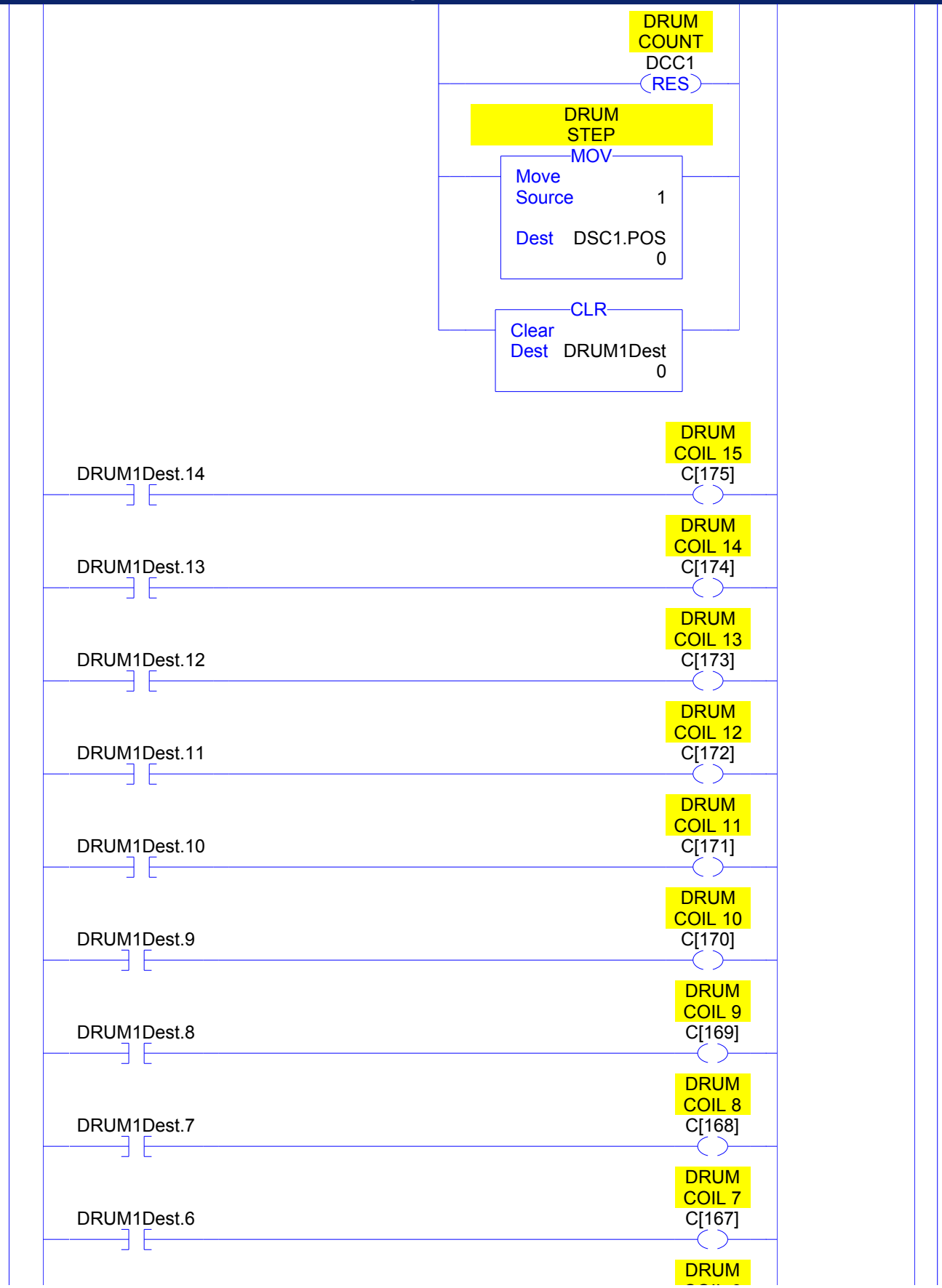


SUBROUTINE
ONE

26



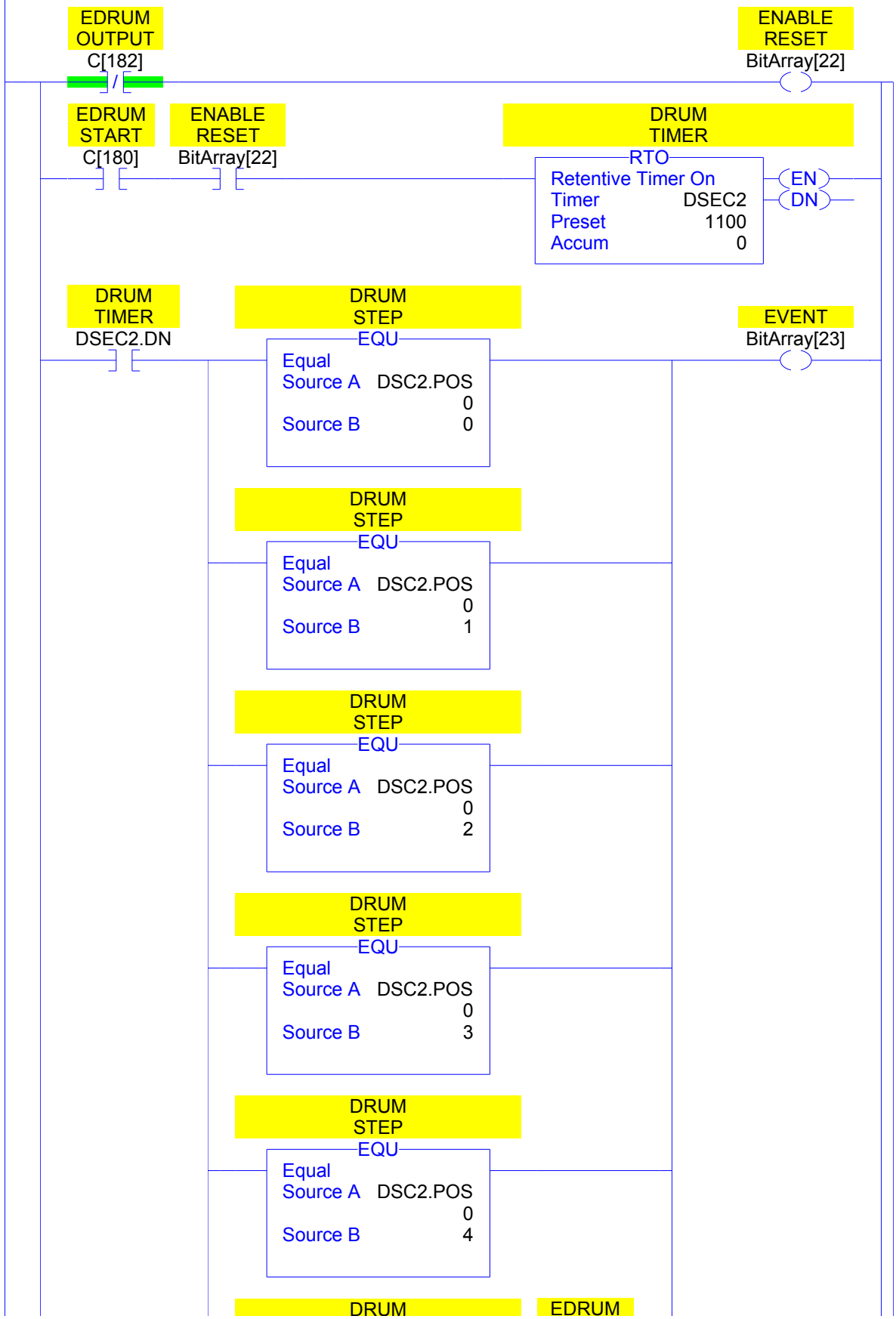
SUBROUTINE
ONE



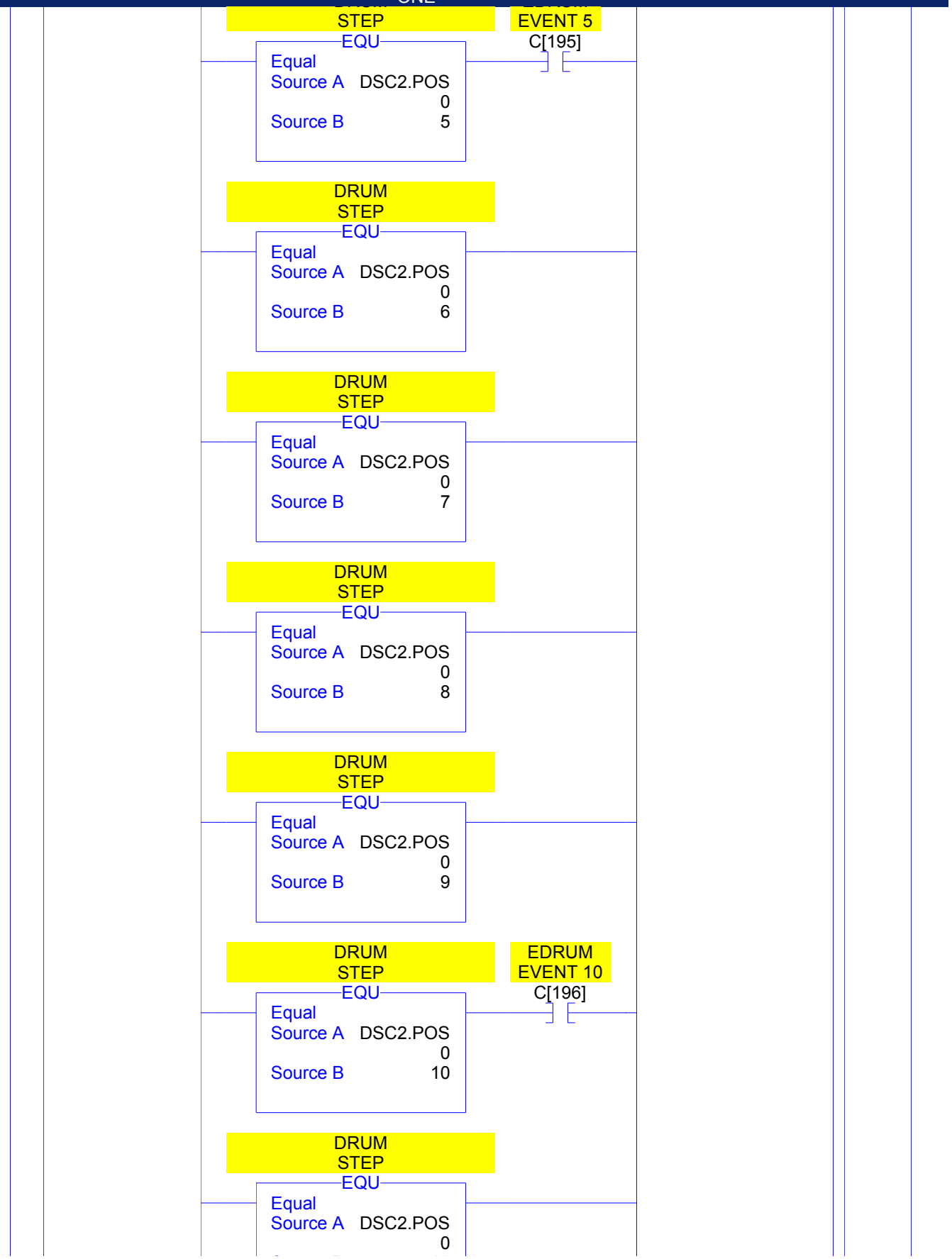
SUBROUTINE
ONE

SUBROUTINE
ONE

27



SUBROUTINE
ONE



SUBROUTINE
ONE

Source B 11

DRUM
STEP
EQU
Equal
Source A DSC2.POS
0
Source B 12

DRUM
STEP
EQU
Equal
Source A DSC2.POS
0
Source B 13

DRUM
STEP
EQU
Equal
Source A DSC2.POS
0
Source B 14

DRUM
STEP
EQU
Equal
Source A DSC2.POS
0
Source B 15

EDRUM
EVENT 15
C[197]

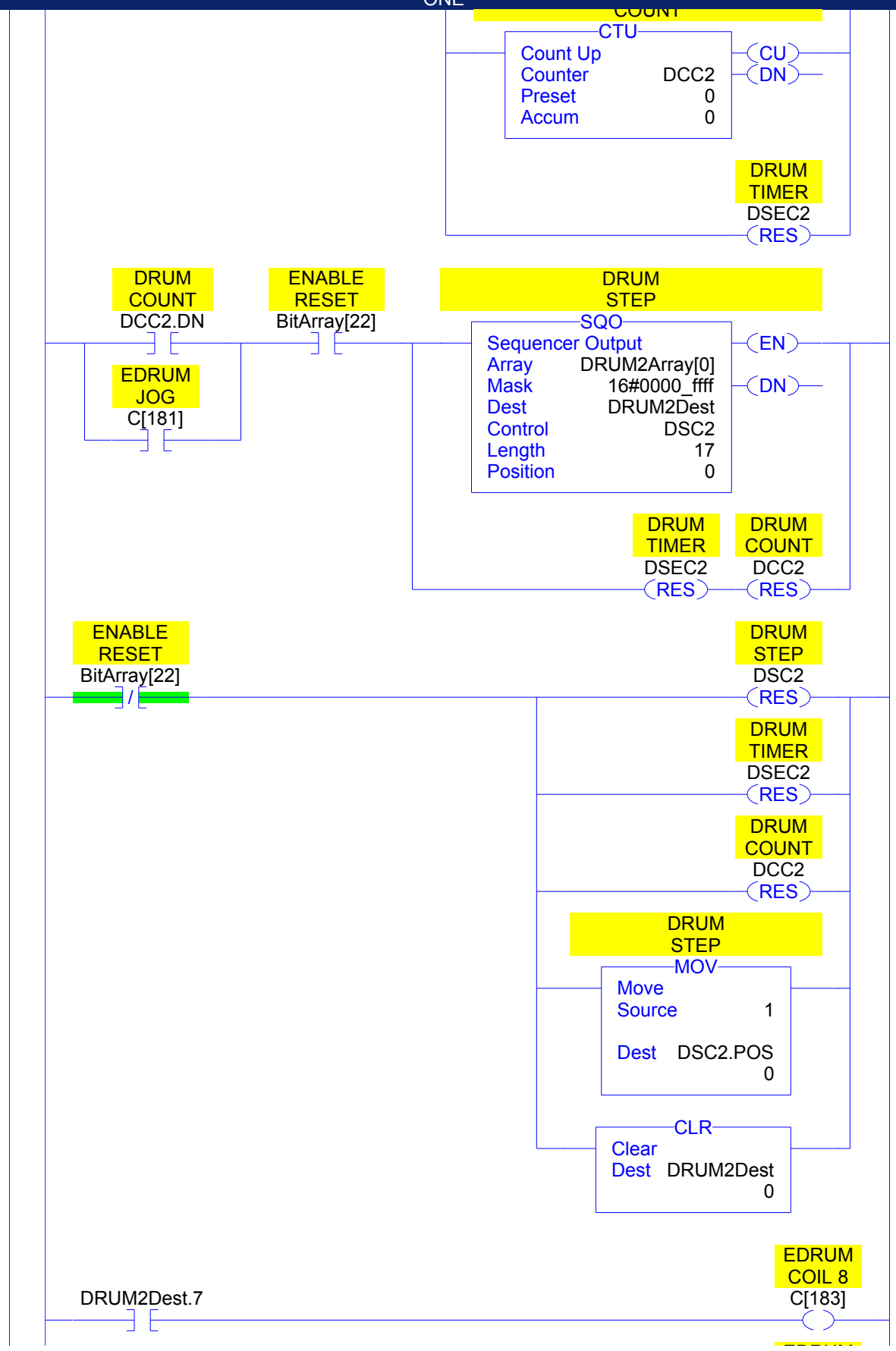
DRUM
STEP
EQU
Equal
Source A DSC2.POS
0
Source B 16

EVENT
BitArray[23]

DRUM
COUNT
MOV
Move
Source DCP2[DSC2.POS]
0
Dest DCC2.PRE
0

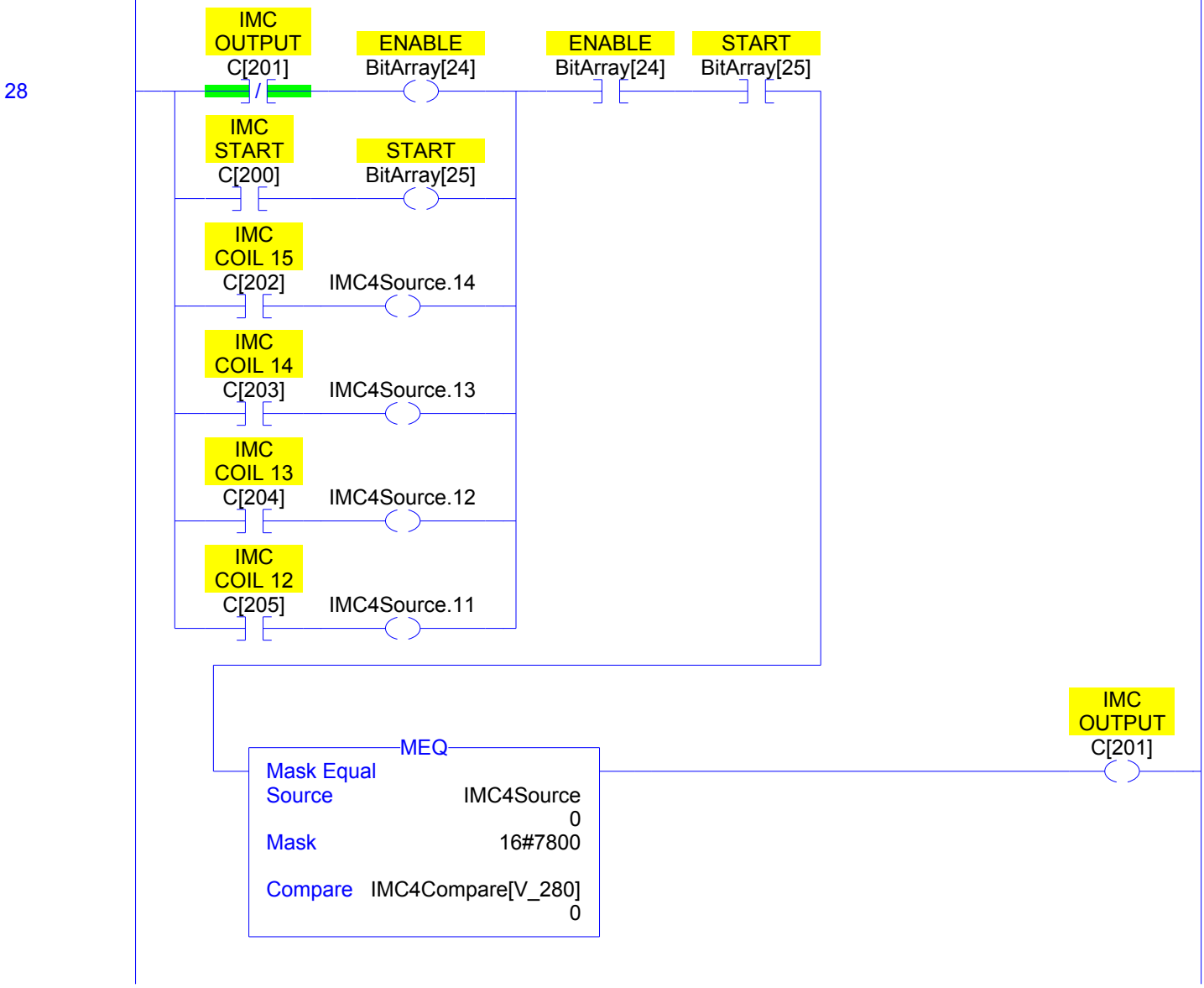
DRUM
COUNT

SUBROUTINE
ONE



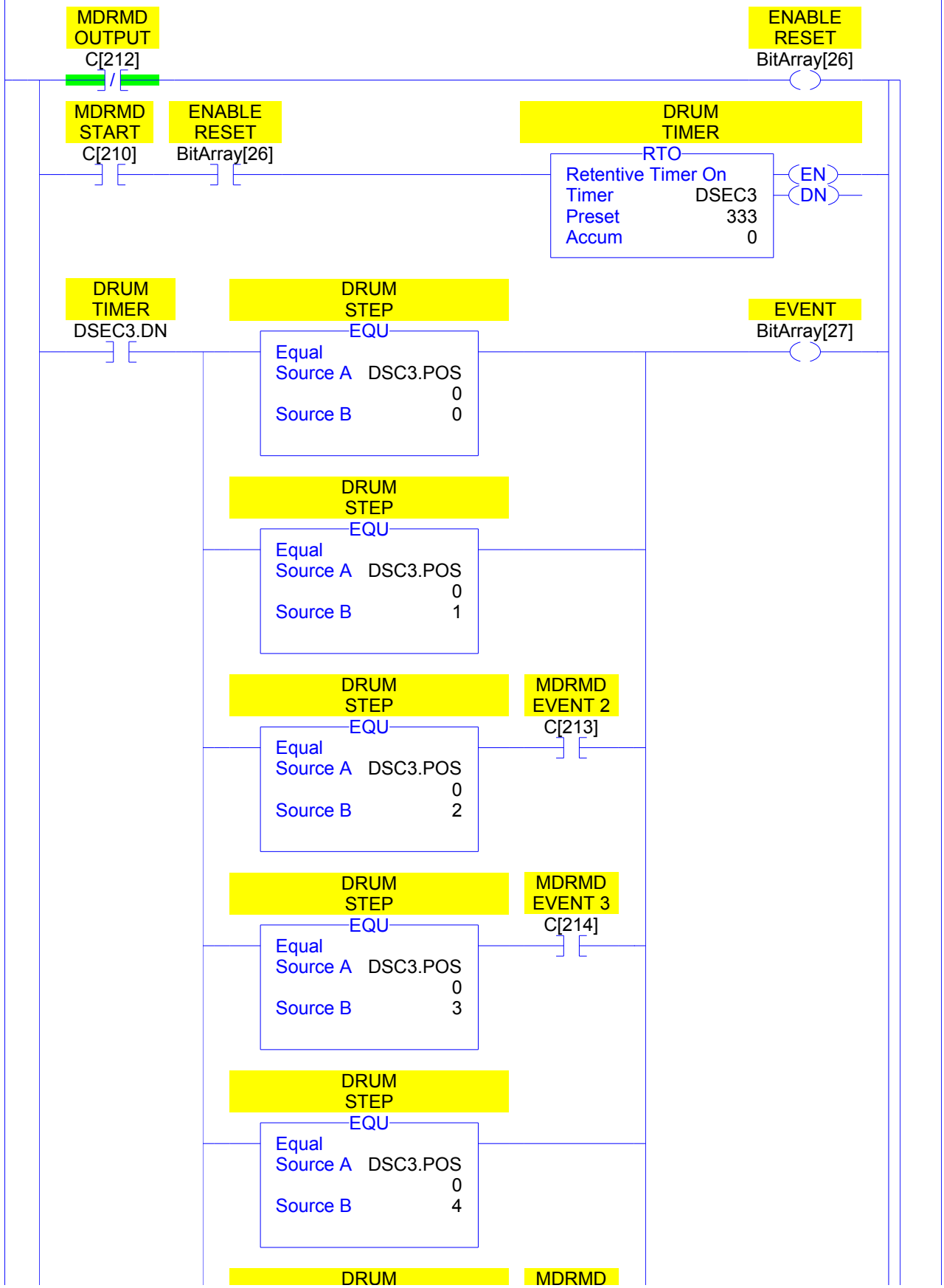
SUBROUTINE
ONE

SUBROUTINE
ONE

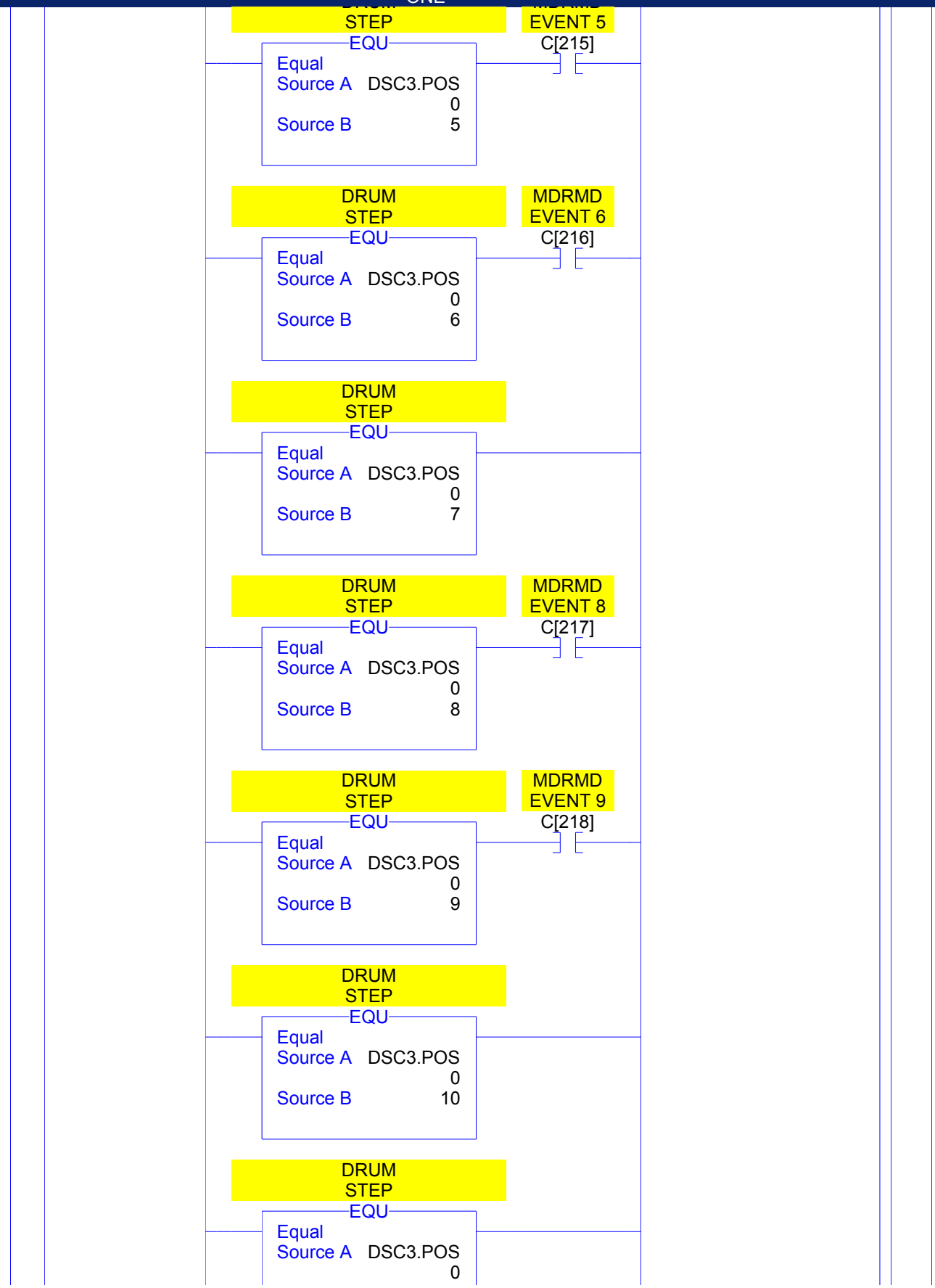


SUBROUTINE
ONE

29



SUBROUTINE
ONE



SUBROUTINE
ONE

Source B 11

DRUM
STEP

EQU
Equal
Source A DSC3.POS
0
Source B 12

DRUM
STEP

EQU
Equal
Source A DSC3.POS
0
Source B 13

DRUM
STEP

EQU
Equal
Source A DSC3.POS
0
Source B 14

DRUM
STEP

EQU
Equal
Source A DSC3.POS
0
Source B 15

DRUM
STEP

EQU
Equal
Source A DSC3.POS
0
Source B 16

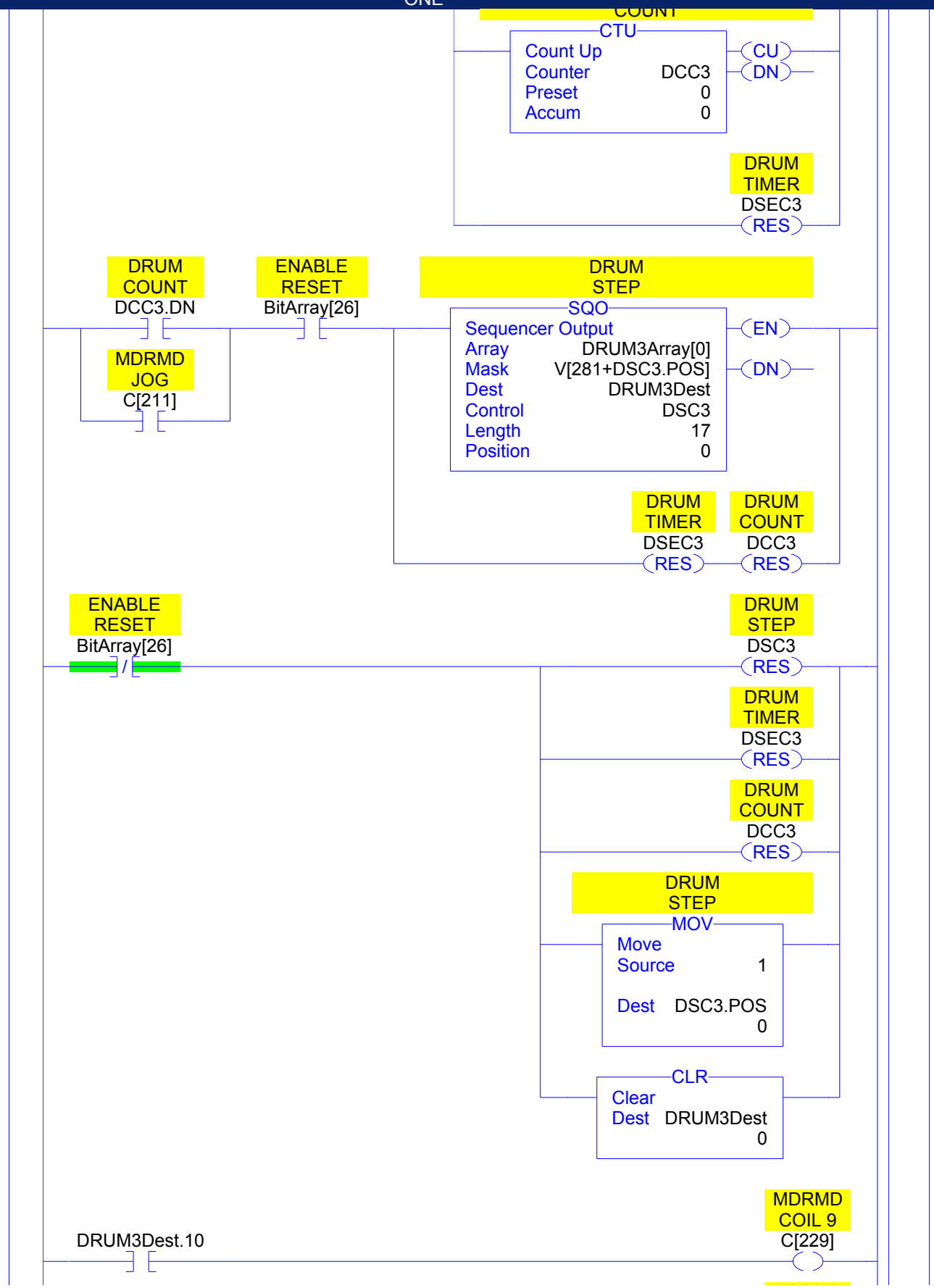
EVENT
BitArray[27]

DRUM
COUNT

MOV
Move
Source DCP3[DSC3.POS]
0
Dest DCC3.PRE
0

DRUM

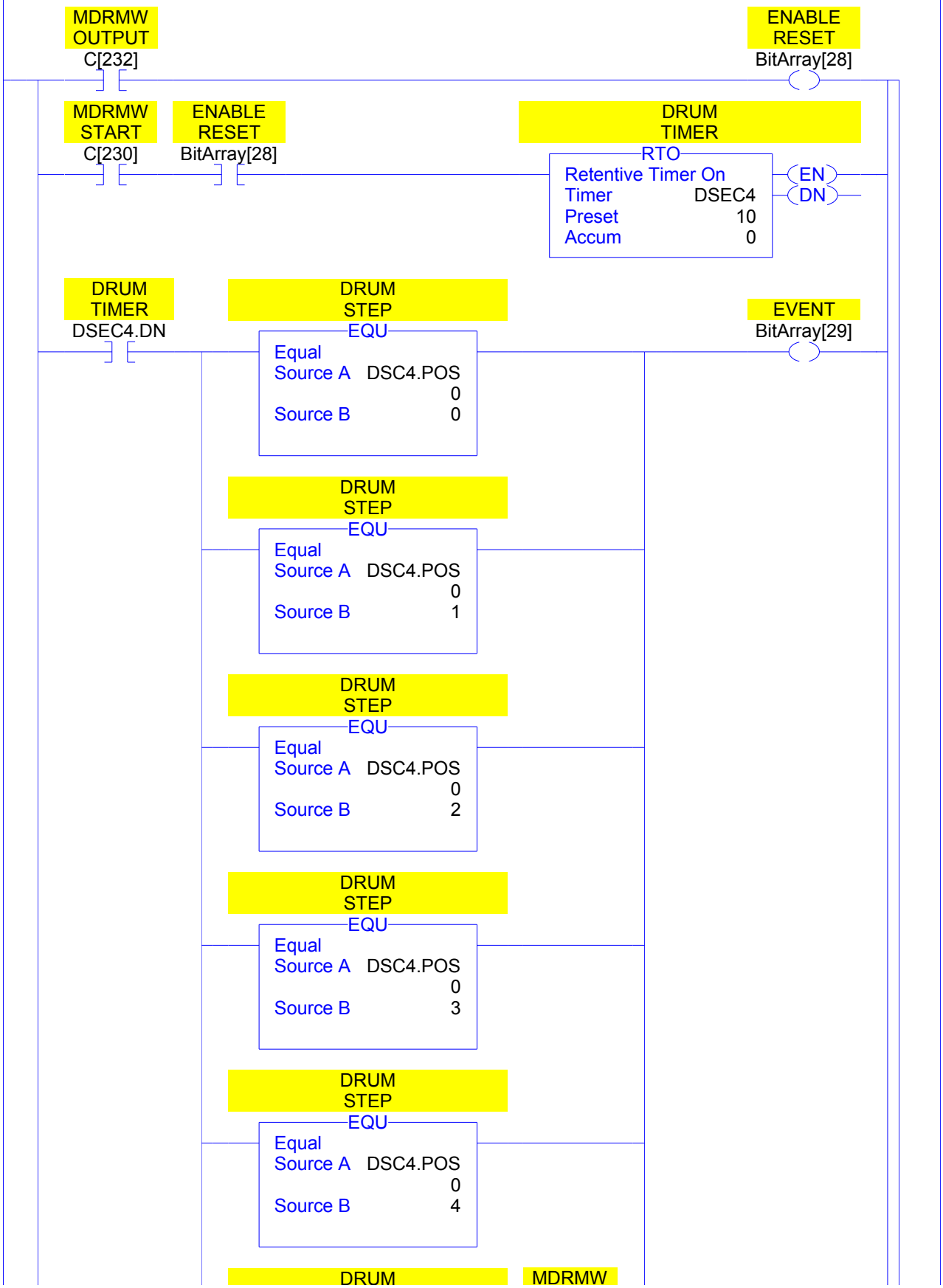
SUBROUTINE
ONE



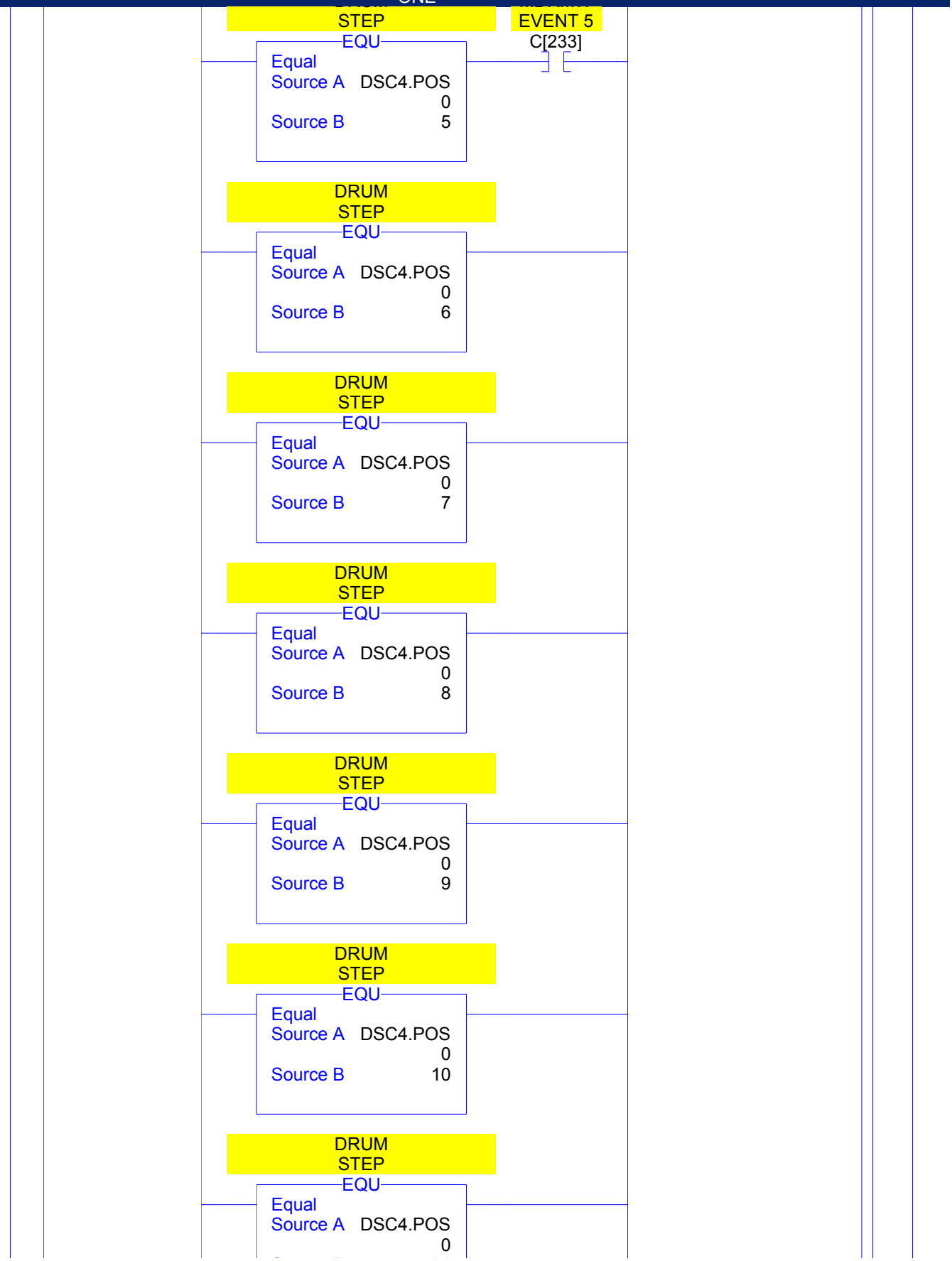
SUBROUTINE
ONE

SUBROUTINE
ONE

30



SUBROUTINE
ONE



SUBROUTINE
ONE

Source B 11

DRUM
STEP

EQU
Equal
Source A DSC4.POS
0
Source B 12

DRUM
STEP

EQU
Equal
Source A DSC4.POS
0
Source B 13

DRUM
STEP

EQU
Equal
Source A DSC4.POS
0
Source B 14

DRUM
STEP

EQU
Equal
Source A DSC4.POS
0
Source B 15

DRUM
STEP

EQU
Equal
Source A DSC4.POS
0
Source B 16

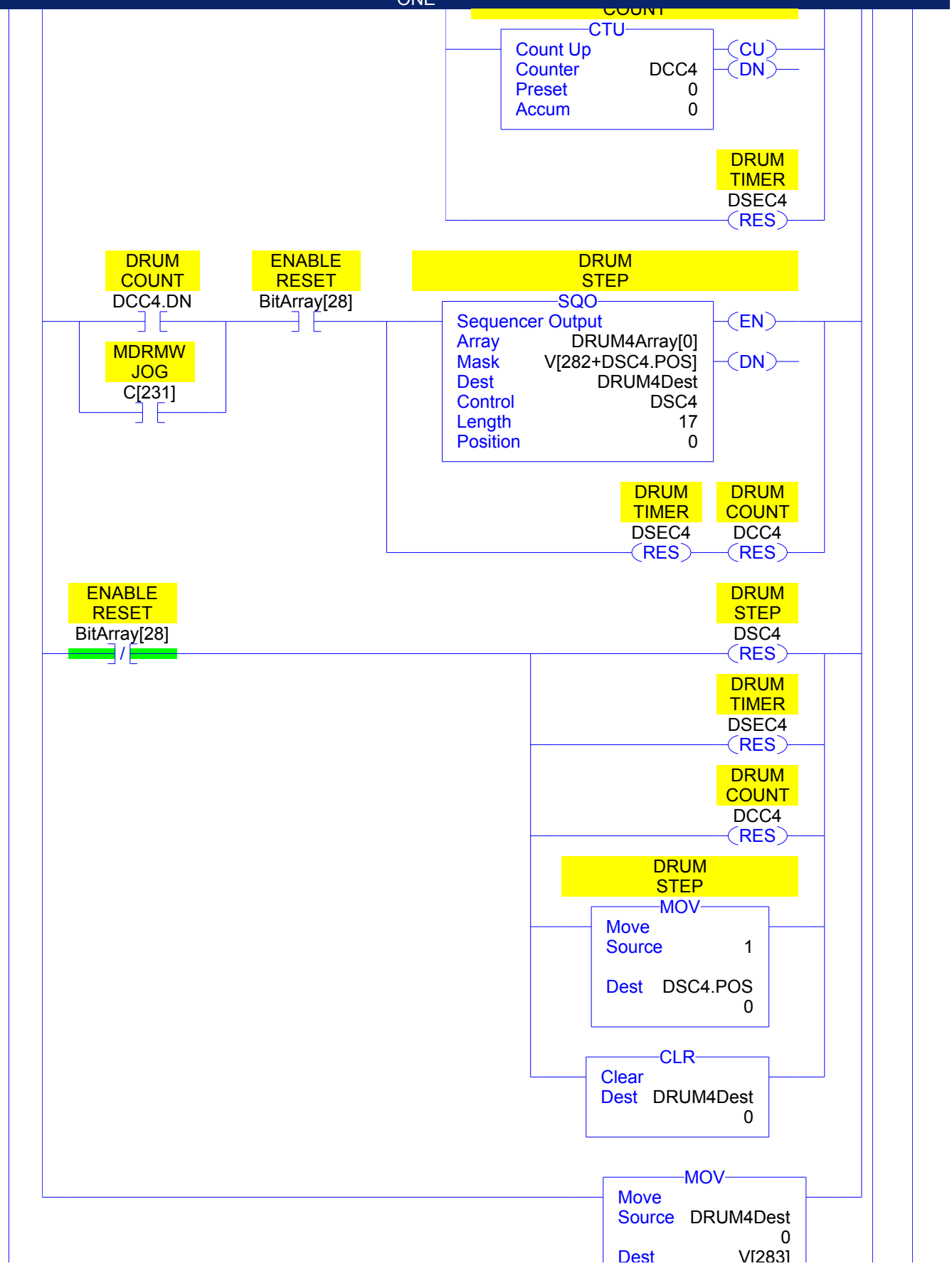
EVENT
BitArray[29]

DRUM
COUNT

MOV
Move
Source DCP4[DSC4.POS]
0
Dest DCC4.PRE
0

DRUM
COUNT

SUBROUTINE
ONE



SUBROUTINE
ONE

0

ENABLE RESET
BitArray[28]

DRUM STEP
DSC4.DN
/

MDRMW OUTPUT
C[232]

31

SHRB ENABLE RESET
C[242]
/

ENABLE
BitArray[30]

ENABLE
BitArray[30]

SHRB DATA
C[241]

DATA
BitArray[31]

SHRB CLOCK
C[240]

ENABLE
BitArray[30]

BSL
Bit Shift Left
Array C[249]
Control SHRB5
Source Bit BitArray[31]
Length 20
EN
DN

ENABLE
BitArray[30]
/

SHRB5
RES

SHRB5.UL

SHRB OUTPUT
C[243]

The TEXT box will change into
a rung comment.

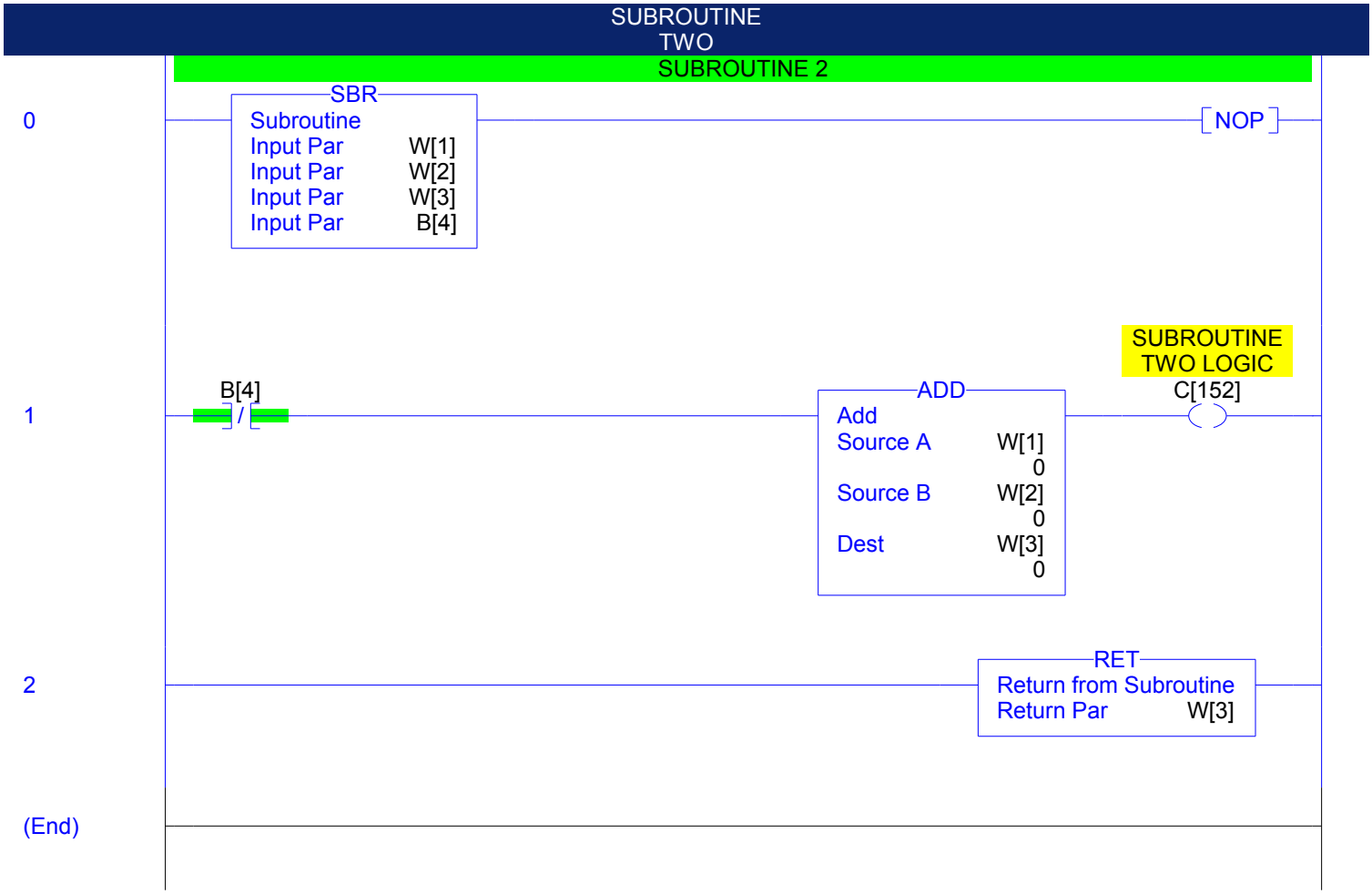
32

NOP

33

RET
Return from Subroutine

(End)


SUBROUTINE
TWO
SUBROUTINE 2
0
SBR
Subroutine
Input Par W[1]
Input Par W[2]
Input Par W[3]
Input Par B[4]
NOP
SUBROUTINE
TWO LOGIC
C[152]
1
B[4]
/
ADD
Add
Source A W[1]
0
Source B W[2]
0
Dest W[3]
0
2
RET
Return from Subroutine
Return Par W[3]
(End)