0

NOP

This is a rung comment. 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.

Bits

1

This is an address comment for input 110/00 SYM_110_00 <_010:I.Data[0].0>

This is an address comment for output 010/00 SYM_010_00 <_010:O.Data[0].0>

This is an address comment for input 110/00 SYM_110_00 <_010:I.Data[0].0>

This program was converted using the DHRIO option. The racks get converted to
full racks on a universal Remote I/O. The PLC2 does not save the rack
information so there is no way to know what hardware actually exists. Bits in
the rack range will be remapped to I/O. Bits outside of this range will point
to the INT array.

2

Instrction comment with no address comment SYM_110_01 <_010:I.Data[0].1>

Comment for 010/01 SYM_010_01 <_010:O.Data[0].1>

L

I/O bit addresses use the format: _RRS:I.Data[M].B where:
RR = Rack in octal
S = Starting module group
I = Input (or O for output)
M = Module group offset
B = Terminal number in decimal

3

Comment for 110/02 SYM_110_02 <_010:I.Data[0].2>

Comment for 010/02 SYM_010_02 <_010:O.Data[0].2>

U

This section of rungs show how bits are converted.

4
Comment for 110/03 SYM_110_03 <_010:I.Data[0].3>
Comment for 010/03 SYM_010_03 <_010:O.Data[0].3>

5
Comment for 110/04 SYM_110_04 <_010:I.Data[0].4>
Comment for 010/04 SYM_010_04 <_010:O.Data[0].4>
L

6
Comment for 110/05 SYM_110_05 <_010:I.Data[0].5>
Comment for 010/05 SYM_010_05 <_010:O.Data[0].5>
U

7
Comment for 110/06 SYM_110_06 <_010:I.Data[0].6>
Comment for 010/06 SYM_010_06 <_010:O.Data[0].6>

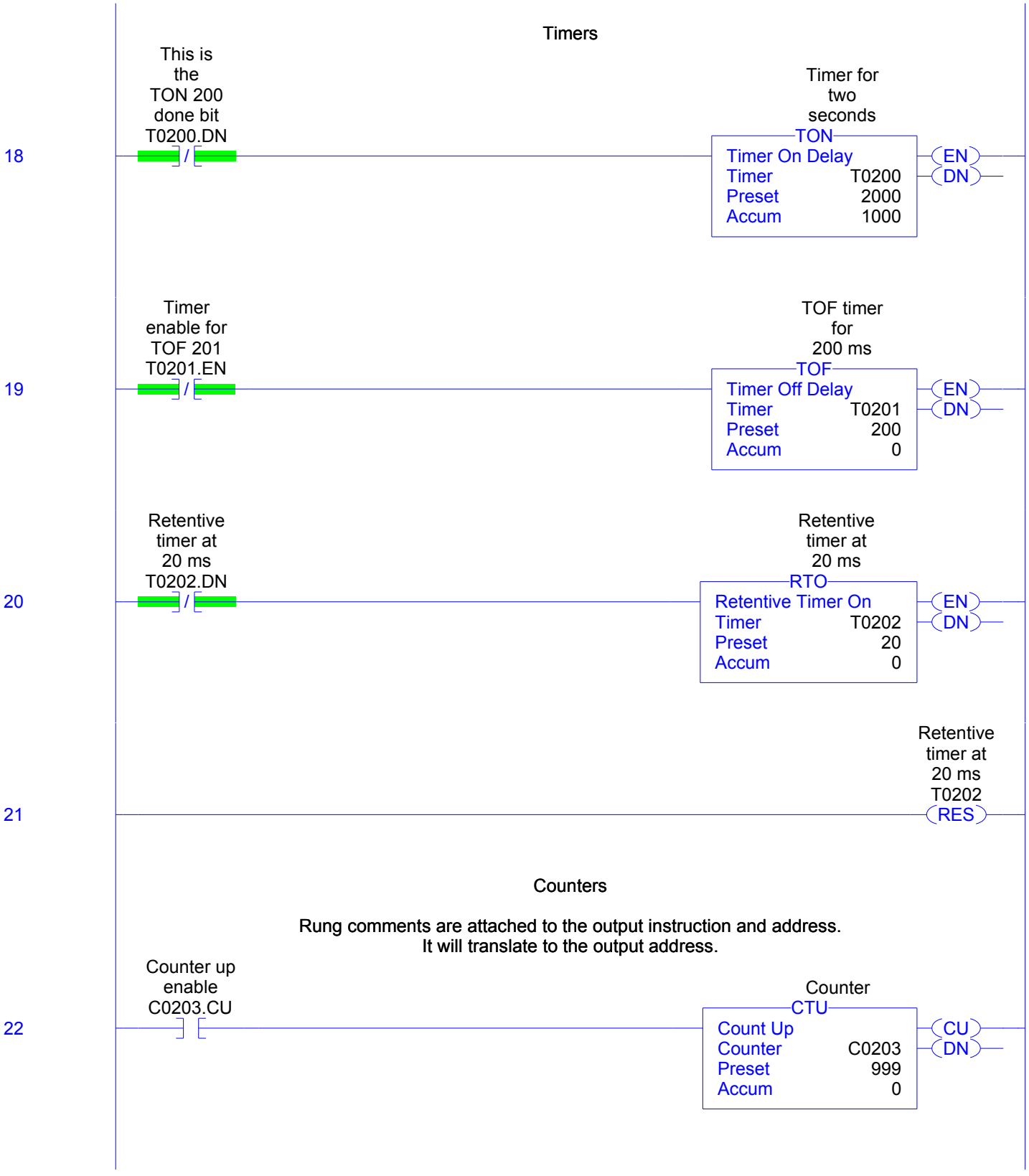
The next three rungs will AFI.

8
AFI
Comment for 110/07 SYM_110_07 <_010:I.Data[0].7>
AFI
Comment for 010/07 SYM_010_07 <_010:O.Data[0].7>
L

9
AFI
Comment for 110/10 SYM_110_10 <_010:I.Data[0].8>
Comment for 010/10 SYM_010_10 <_010:O.Data[0].8>
U

10
Comment for 110/11 SYM_110_11 <_010:I.Data[0].9>
AFI
Comment for 010/11 SYM_010_11 <_010:O.Data[0].9>


11
Comment for 110/12 SYM_110_12 <_010:I.Data[0].10>
Comment for 010/12 SYM_010_12 <_010:O.Data[0].10>
L
12
Comment for 110/13 SYM_110_13 <_010:I.Data[0].11>
Comment for 010/13 SYM_010_13 <_010:O.Data[0].11>
U
13
Comment for 110/14 SYM_110_14 <_010:I.Data[0].12>
Comment for 010/14 SYM_010_14 <_010:O.Data[0].12>
14
Comment for 110/15 SYM_110_15 <_010:I.Data[0].13>
Comment for 010/15 SYM_010_15 <_010:O.Data[0].13>
L
15
Comment for 110/16 SYM_110_16 <_010:I.Data[0].14>
Comment for 010/16 SYM_010_16 <_010:O.Data[0].14>
U
16
Comment for 110/17 SYM_110_17 <_010:I.Data[0].15>
Comment for 010/17 SYM_010_17 <_010:O.Data[0].15>
17
This address is outside the rack range M[40].0
Comment M[40].1
This commt is at the word level 150 M[104].0
This commt is at the bit level, overrides the word M[104].1
Comment M[168].0
Comment M[168].1
Comment M[232].15


Timers
18
This is the TON 200 done bit T0200.DN
Timer for two seconds
TON
Timer On Delay
Timer T0200
Preset 2000
Accum 1000
EN
DN
19
Timer enable for TOF 201 T0201.EN
TOF timer for 200 ms
TOF
Timer Off Delay
Timer T0201
Preset 200
Accum 0
EN
DN
20
Retentive timer at 20 ms T0202.DN
Retentive timer at 20 ms
RTO
Retentive Timer On
Timer T0202
Preset 20
Accum 0
EN
DN
21
Retentive timer at 20 ms T0202
RES
Counters
Rung comments are attached to the output instruction and address.
It will translate to the output address.
22
Counter up enable C0203.CU
Counter
CTU
Count Up
Counter C0203
Preset 999
Accum 0
CU
DN

Counters

Rung comments are attached to the output instruction and address.
It will translate to the output address.

23

Counter down enable C0203.CD

Counter

CTD Count Down Counter C0203
Preset 999
Accum 0

CD
DN

Counters

Rung comments are attached to the output instruction and address.
It will translate to the output address.

24

Counter done C0203.DN

Counter C0203
RES

Block Transfers

25

Block Transfer Write control block H0205.DN

Block Transfer Write control block H0205.EN

Block Transfer Write control block

MSG
Type - Block Transfer Write
Message Control H0205 ...

EN
DN
ER

26

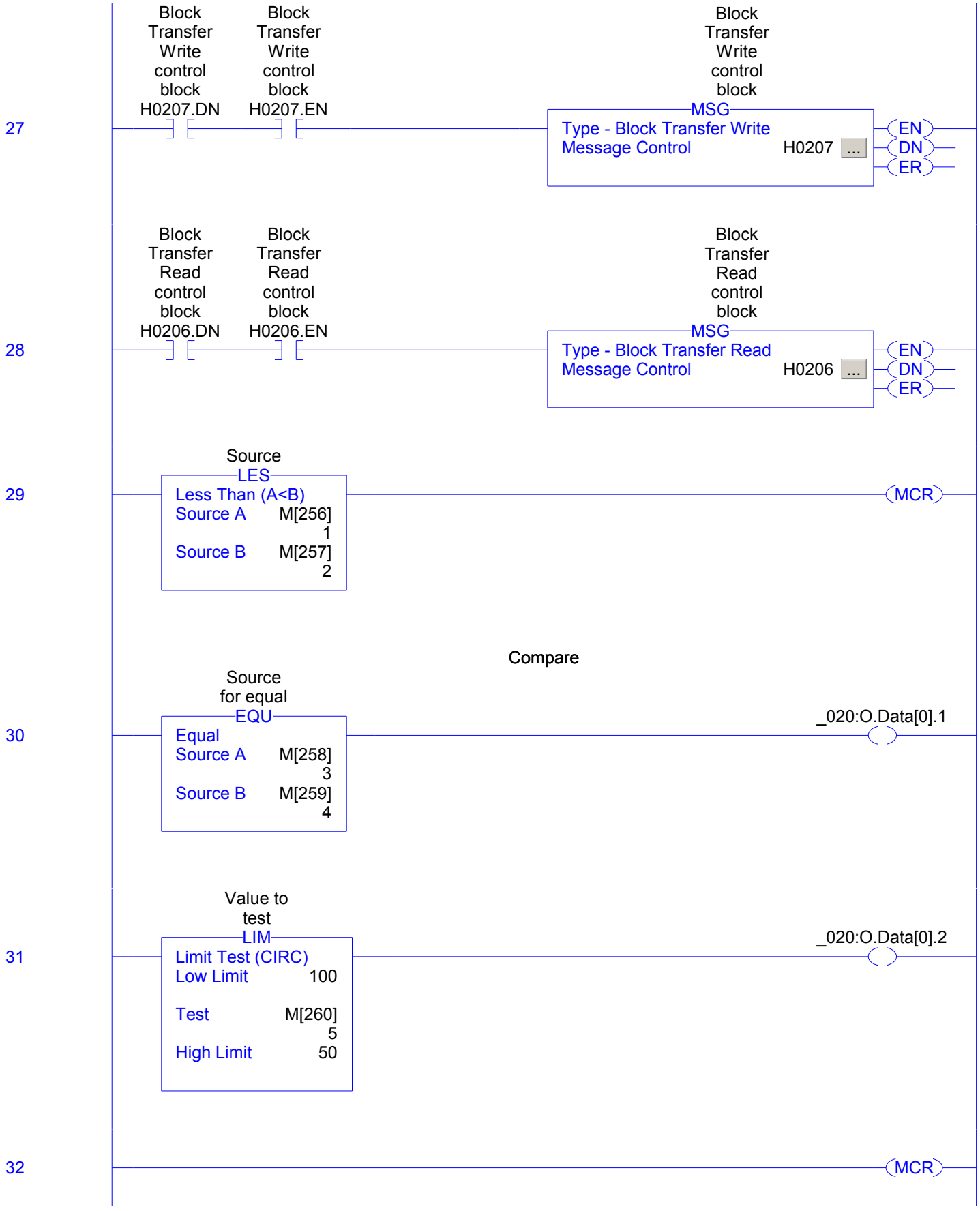
Block Transfer Read control block H0204.DN

Block Transfer Read control block H0204.EN

Block Transfer Read control block

MSG
Type - Block Transfer Read
Message Control H0204 ...

EN
DN
ER


27
Block Transfer Write control block H0207.DN
Block Transfer Write control block H0207.EN
Block Transfer Write control block
MSG
Type - Block Transfer Write
Message Control H0207
EN
DN
ER
28
Block Transfer Read control block H0206.DN
Block Transfer Read control block H0206.EN
Block Transfer Read control block
MSG
Type - Block Transfer Read
Message Control H0206
EN
DN
ER
29
Source
LES
Less Than (A<B)
Source A M[256] 1
Source B M[257] 2
MCR
Compare
30
Source for equal
EQU
Equal
Source A M[258] 3
Source B M[259] 4
_020:O.Data[0].1
31
Value to test
LIM
Limit Test (CIRC)
Low Limit 100
Test M[260] 5
High Limit 50
_020:O.Data[0].2
32
MCR

Math

33

Sum
ADD
Add
Source A M[262] 7
Source B M[263] 8
Dest M[264] 15

34

Difference
SUB
Subtract
Source A M[265] 9
Source B M[266] 10
Dest M[267] 4001

35

Product
MUL
Multiply
Source A M[268] 11
Source B M[269] 12
Dest M[270] 0

MUL
Multiply
Source A M[268] 11
Source B M[269] 12
Dest M[271] 132

36

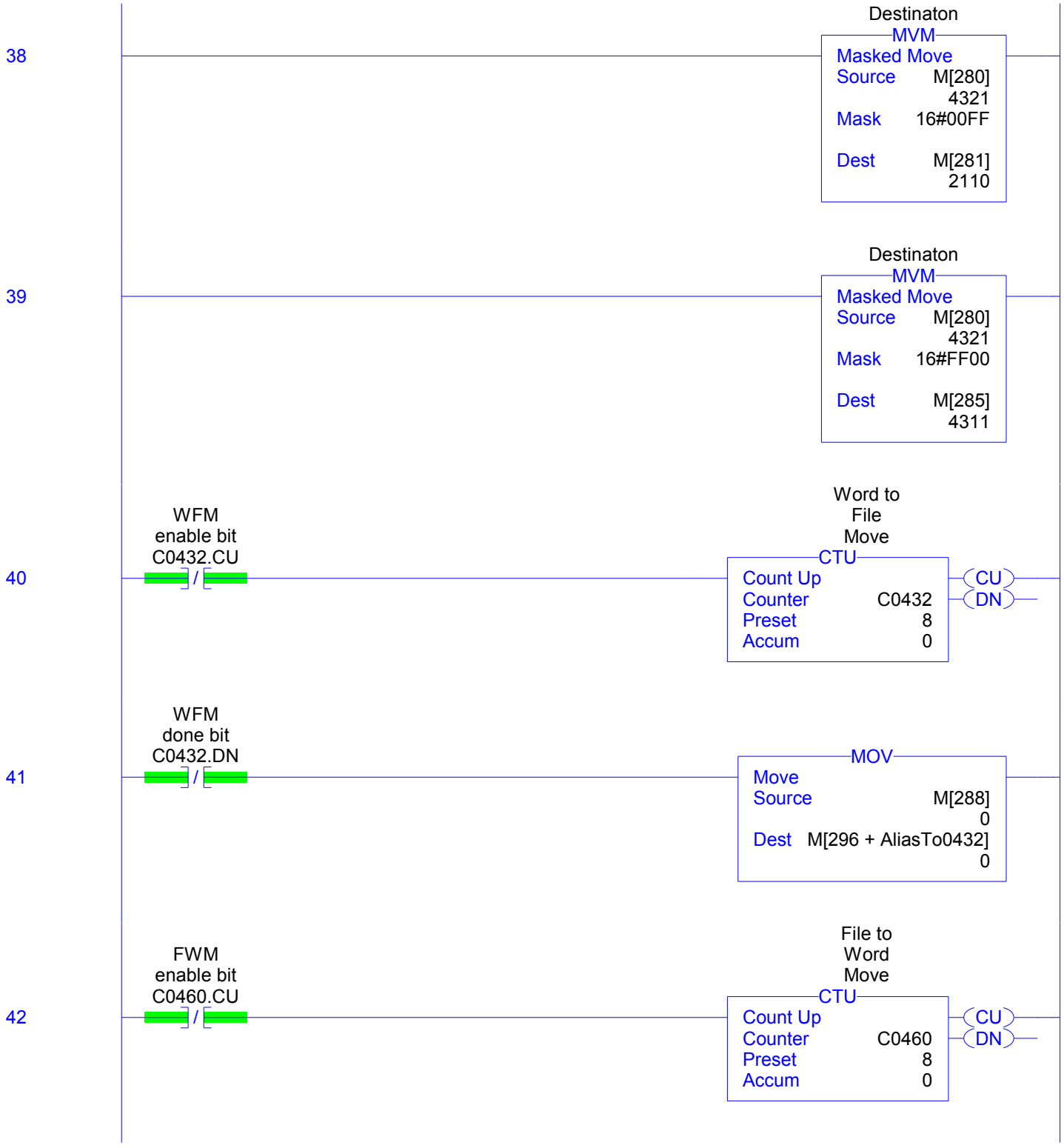
Quotient
DIV
Divide
Source A M[272] 13
Source B M[273] 14
Dest M[274] 0

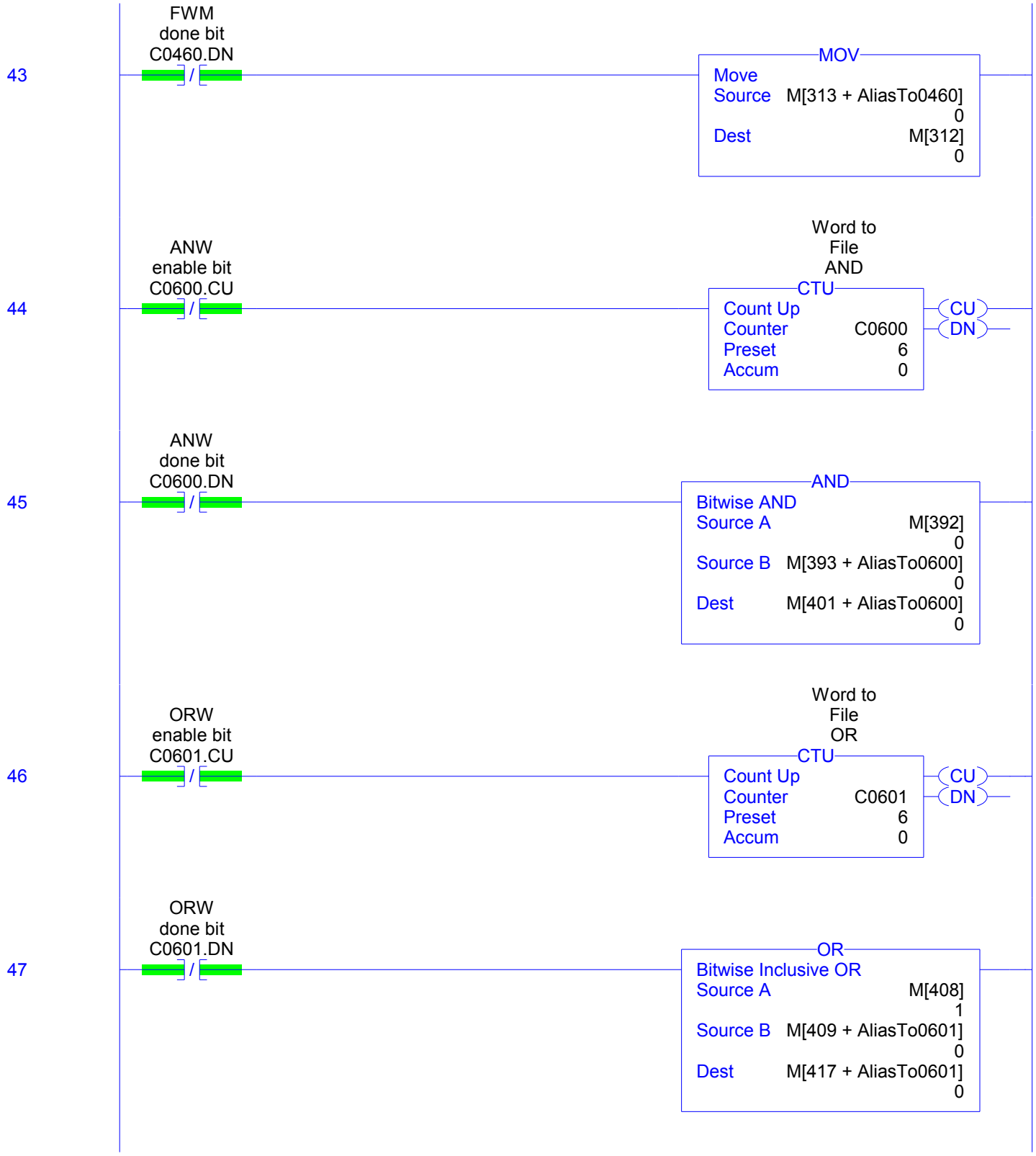
DIV
Divide
Source A M[272] 13
Source B M[273] 14
Dest M[275] 928

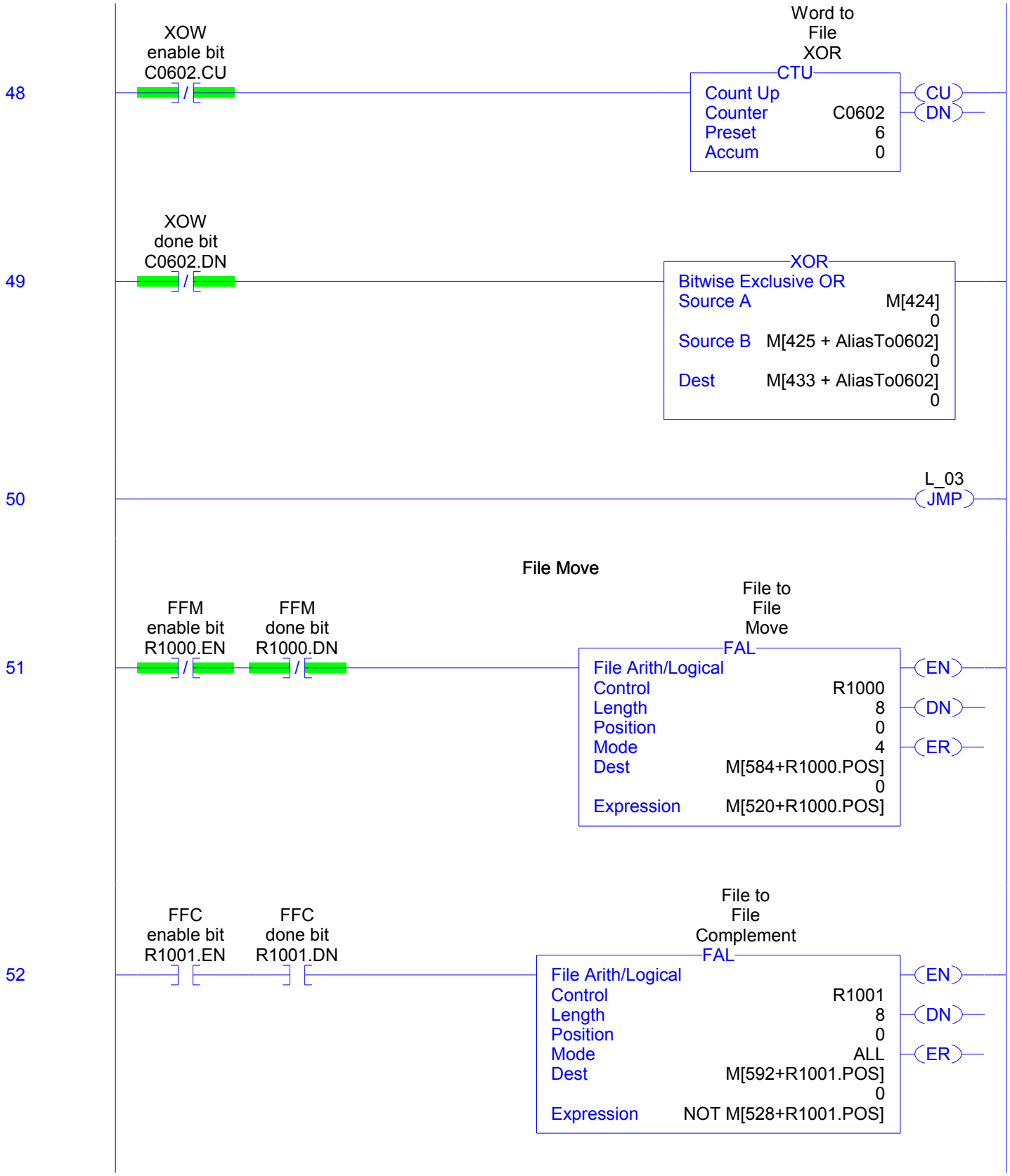
Data Move

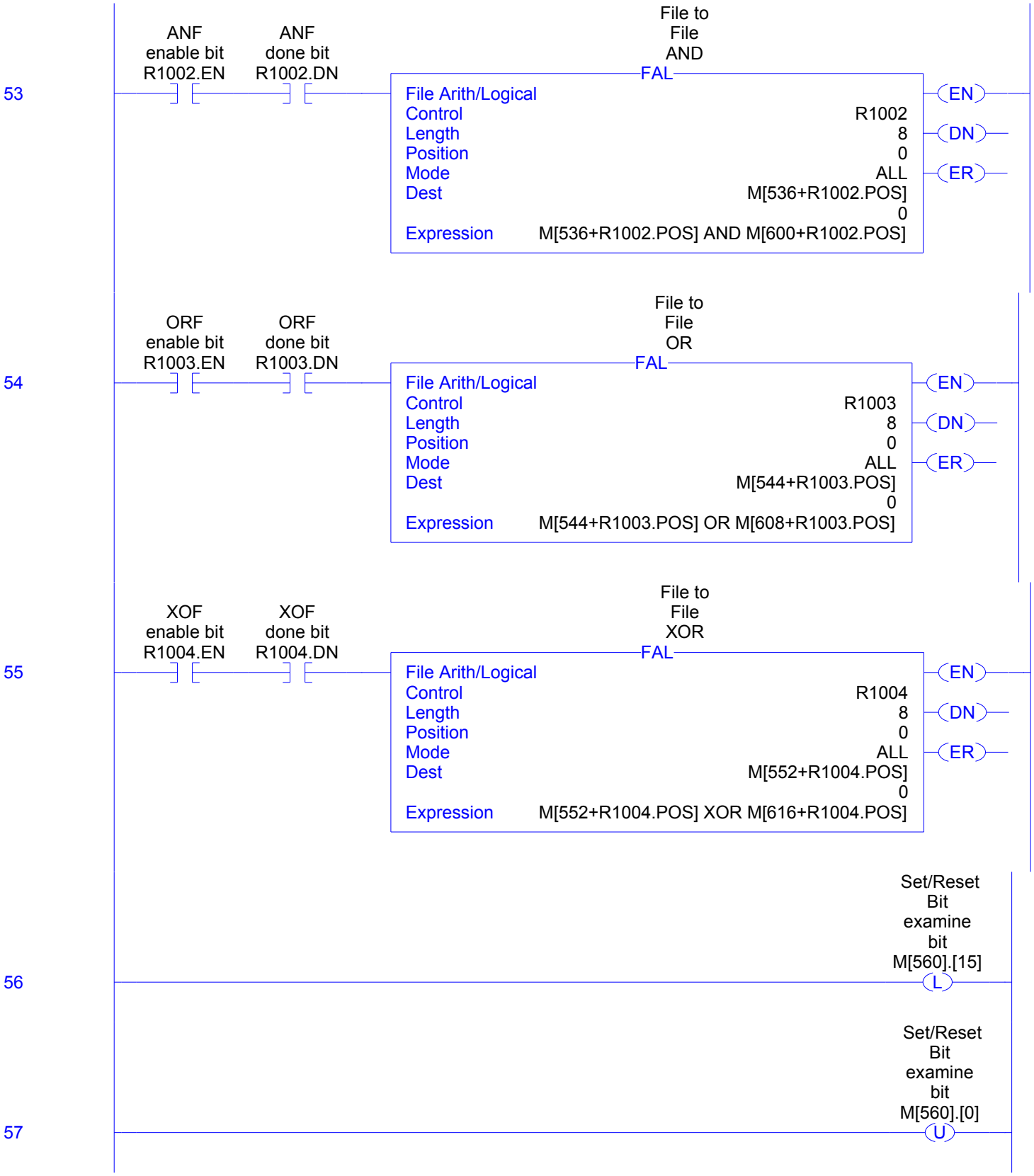
37

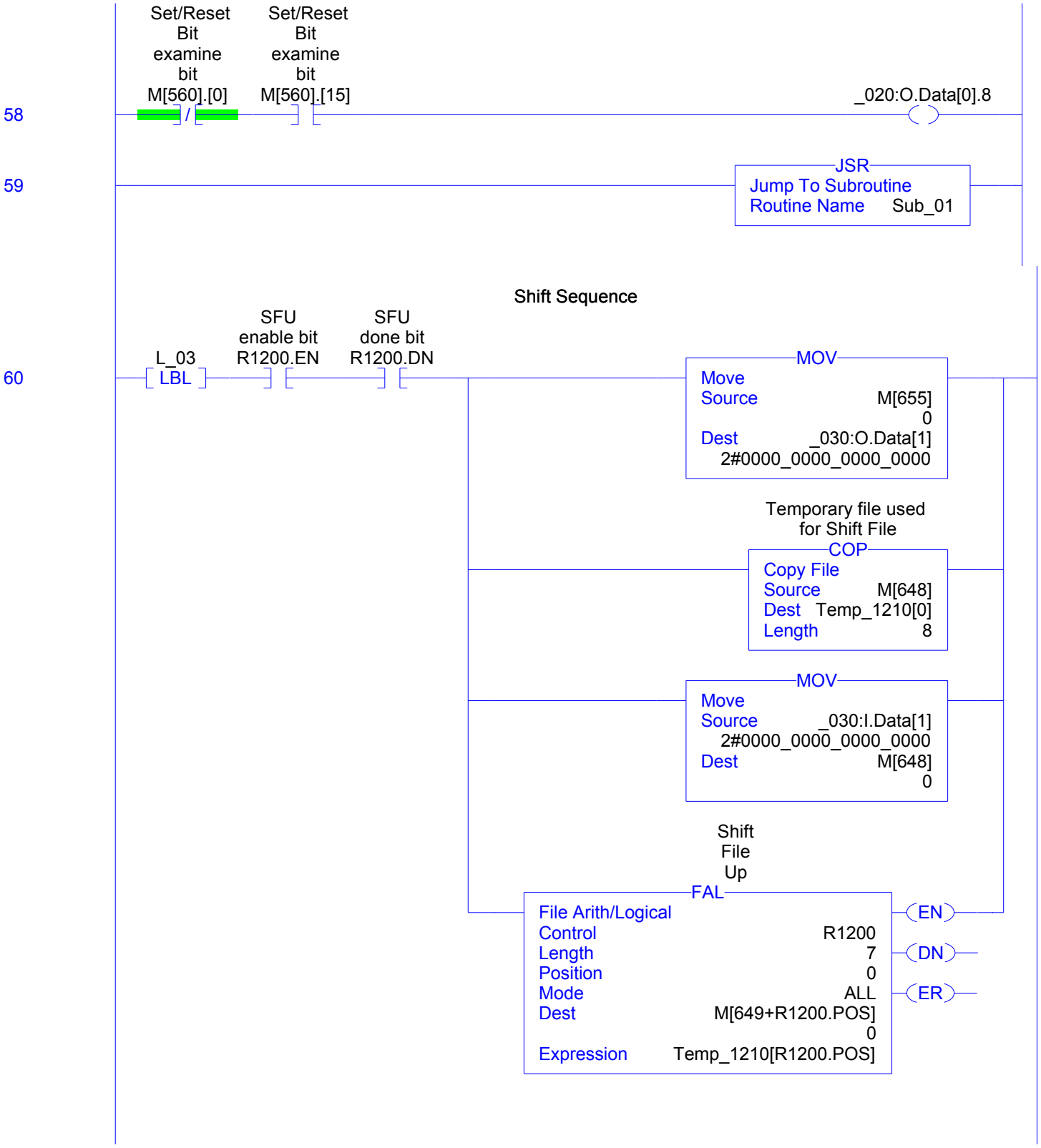
Destinaton
MOV
Move
Source M[276] 30
Dest M[277] 30

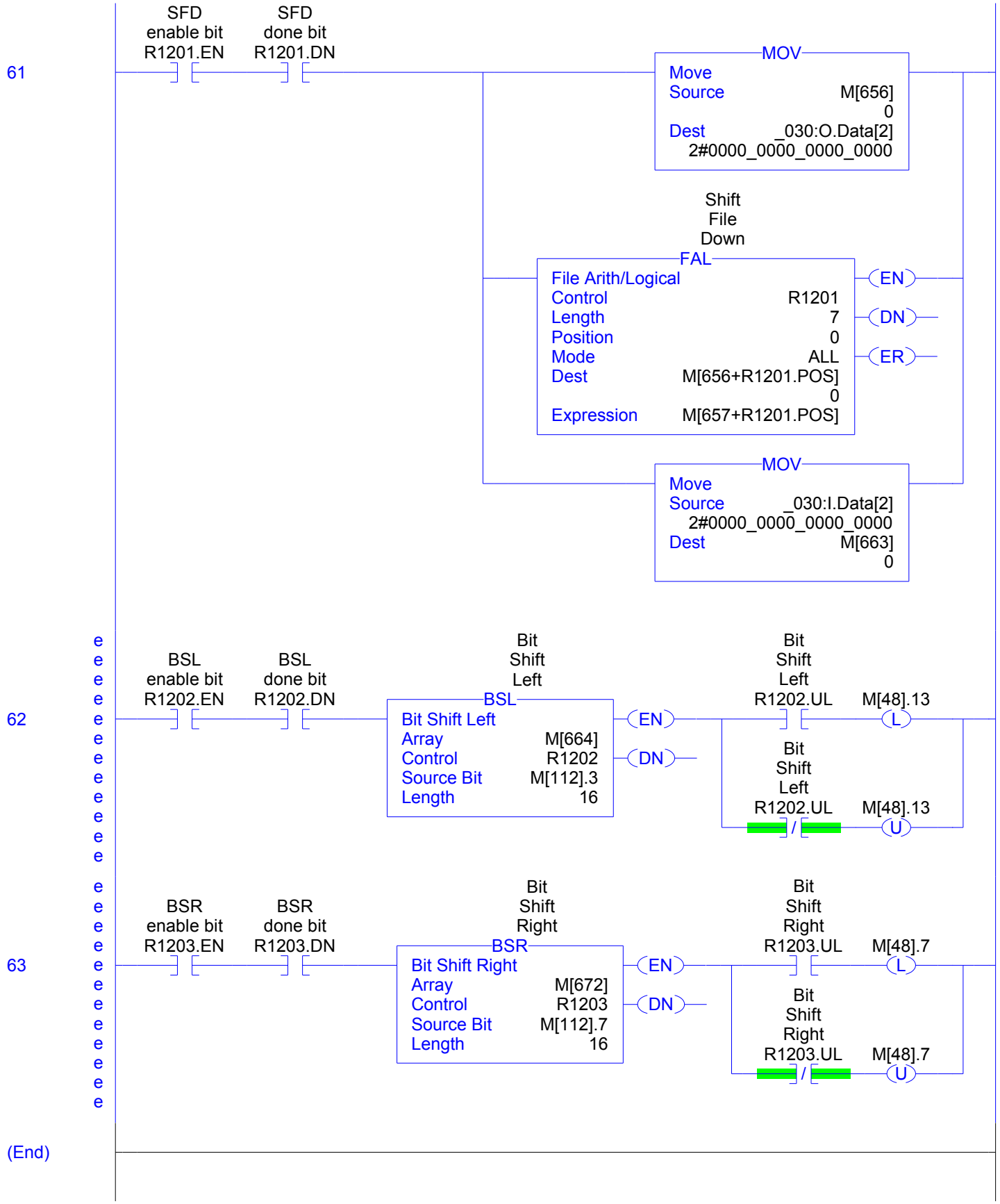

38
Destinaton
MVM
Masked Move
Source M[280]
4321
Mask 16#00FF
Dest M[281]
2110
39
Destinaton
MVM
Masked Move
Source M[280]
4321
Mask 16#FF00
Dest M[285]
4311
40
WFM
enable bit
C0432.CU
Word to
File
Move
CTU
Count Up
Counter C0432
Preset 8
Accum 0
CU
DN
41
WFM
done bit
C0432.DN
MOV
Move
Source M[288]
0
Dest M[296 + AliasTo0432]
0
42
FWM
enable bit
C0460.CU
File to
Word
Move
CTU
Count Up
Counter C0460
Preset 8
Accum 0
CU
DN

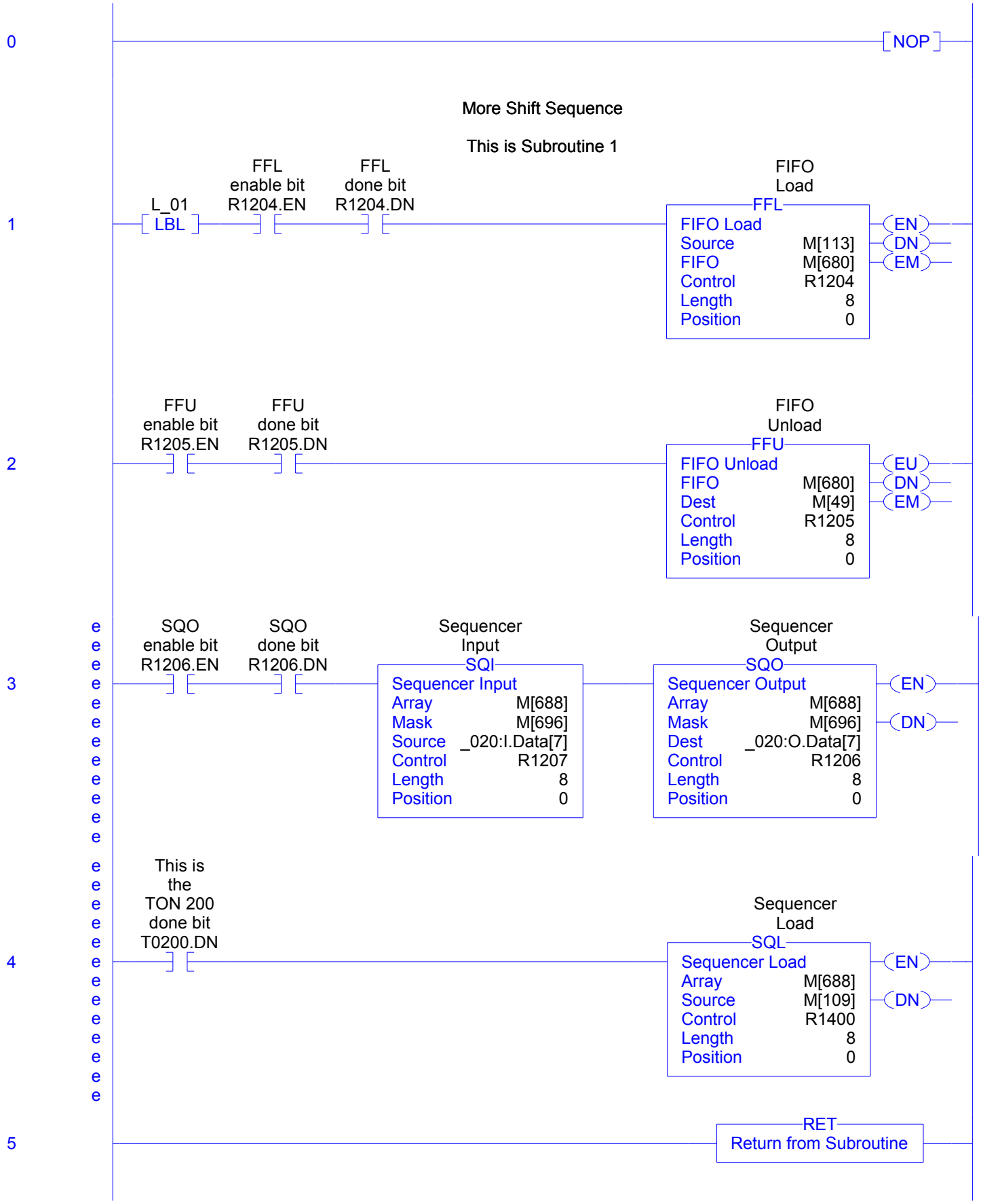

43
FWM
done bit
C0460.DN
MOV
Move
Source M[313 + AliasTo0460]
0
Dest M[312]
0
44
ANW
enable bit
C0600.CU
Word to
File
AND
CTU
Count Up
Counter C0600
Preset 6
Accum 0
CU
DN
45
ANW
done bit
C0600.DN
AND
Bitwise AND
Source A M[392]
0
Source B M[393 + AliasTo0600]
0
Dest M[401 + AliasTo0600]
0
46
ORW
enable bit
C0601.CU
Word to
File
OR
CTU
Count Up
Counter C0601
Preset 6
Accum 0
CU
DN
47
ORW
done bit
C0601.DN
OR
Bitwise Inclusive OR
Source A M[408]
1
Source B M[409 + AliasTo0601]
0
Dest M[417 + AliasTo0601]
0


48
XOW
enable bit
C0602.CU
Word to
File
XOR
CTU
Count Up
Counter C0602
Preset 6
Accum 0
CU
DN
49
XOW
done bit
C0602.DN
XOR
Bitwise Exclusive OR
Source A M[424]
0
Source B M[425 + AliasTo0602]
0
Dest M[433 + AliasTo0602]
0
50
L_03
JMP
File Move
51
FFM
enable bit
R1000.EN
FFM
done bit
R1000.DN
File to
File
Move
FAL
File Arith/Logical
Control R1000
Length 8
Position 0
Mode 4
Dest M[584+R1000.POS]
0
Expression M[520+R1000.POS]
EN
DN
ER
52
FFC
enable bit
R1001.EN
FFC
done bit
R1001.DN
File to
File
Complement
FAL
File Arith/Logical
Control R1001
Length 8
Position 0
Mode ALL
Dest M[592+R1001.POS]
0
Expression NOT M[528+R1001.POS]
EN
DN
ER


53
ANF enable bit R1002.EN
ANF done bit R1002.DN
File to File AND
FAL
File Arith/Logical
Control R1002
Length 8
Position 0
Mode ALL
Dest M[536+R1002.POS] 0
Expression M[536+R1002.POS] AND M[600+R1002.POS]
EN
DN
ER
54
ORF enable bit R1003.EN
ORF done bit R1003.DN
File to File OR
FAL
File Arith/Logical
Control R1003
Length 8
Position 0
Mode ALL
Dest M[544+R1003.POS] 0
Expression M[544+R1003.POS] OR M[608+R1003.POS]
EN
DN
ER
55
XOF enable bit R1004.EN
XOF done bit R1004.DN
File to File XOR
FAL
File Arith/Logical
Control R1004
Length 8
Position 0
Mode ALL
Dest M[552+R1004.POS] 0
Expression M[552+R1004.POS] XOR M[616+R1004.POS]
EN
DN
ER
56
Set/Reset Bit examine bit M[560].[15]
L
57
Set/Reset Bit examine bit M[560].[0]
U


Set/Reset Bit examine bit M[560].[0]
Set/Reset Bit examine bit M[560].[15]
_020:O.Data[0].8
58
59
JSR
Jump To Subroutine
Routine Name Sub_01
Shift Sequence
L_03
LBL
SFU enable bit R1200.EN
SFU done bit R1200.DN
60
MOV
Move
Source M[655]
0
Dest _030:O.Data[1]
2#0000_0000_0000_0000
Temporary file used for Shift File
COP
Copy File
Source M[648]
Dest Temp_1210[0]
Length 8
MOV
Move
Source _030:I.Data[1]
2#0000_0000_0000_0000
Dest M[648]
0
Shift File Up
FAL
File Arith/Logical
Control R1200
Length 7
Position 0
Mode ALL
Dest M[649+R1200.POS]
0
Expression Temp_1210[R1200.POS]
EN
DN
ER


61
SFD
enable bit
R1201.EN
SFD
done bit
R1201.DN
MOV
Move
Source M[656]
0
Dest _030:O.Data[2]
2#0000_0000_0000_0000
Shift
File
Down
FAL
File Arith/Logical
Control R1201
Length 7
Position 0
Mode ALL
Dest M[656+R1201.POS]
0
Expression M[657+R1201.POS]
EN
DN
ER
MOV
Move
Source _030:I.Data[2]
2#0000_0000_0000_0000
Dest M[663]
0
62
BSL
enable bit
R1202.EN
BSL
done bit
R1202.DN
Bit
Shift
Left
BSL
Bit Shift Left
Array M[664]
Control R1202
Source Bit M[112].3
Length 16
EN
DN
Bit
Shift
Left
R1202.UL
M[48].13
L
Bit
Shift
Left
R1202.UL
M[48].13
U
63
BSR
enable bit
R1203.EN
BSR
done bit
R1203.DN
Bit
Shift
Right
BSR
Bit Shift Right
Array M[672]
Control R1203
Source Bit M[112].7
Length 16
EN
DN
Bit
Shift
Right
R1203.UL
M[48].7
L
Bit
Shift
Right
R1203.UL
M[48].7
U
(End)


0
NOP
More Shift Sequence
This is Subroutine 1
1
L_01
LBL
FFL enable bit R1204.EN
FFL done bit R1204.DN
FIFO Load
FFL
FIFO Load
Source M[113]
FIFO M[680]
Control R1204
Length 8
Position 0
EN
DN
EM
2
FFU enable bit R1205.EN
FFU done bit R1205.DN
FIFO Unload
FFU
FIFO Unload
FIFO M[680]
Dest M[49]
Control R1205
Length 8
Position 0
EU
DN
EM
3
SQO enable bit R1206.EN
SQO done bit R1206.DN
Sequencer Input
SQI
Sequencer Input
Array M[688]
Mask M[696]
Source _020:I.Data[7]
Control R1207
Length 8
Position 0
Sequencer Output
SQO
Sequencer Output
Array M[688]
Mask M[696]
Dest _020:O.Data[7]
Control R1206
Length 8
Position 0
EN
DN
4
This is the TON 200 done bit T0200.DN
Sequencer Load
SQL
Sequencer Load
Array M[688]
Source M[109]
Control R1400
Length 8
Position 0
EN
DN
5
RET
Return from Subroutine


(End)