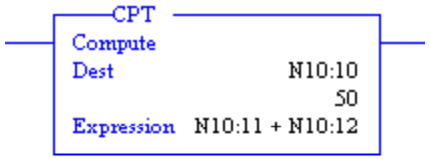
Pertemuan ke-12

Math Instructions

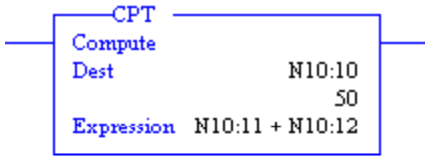
Objektif

- Memahami operasi dari instruksi Compute kemudian mengaplikasikannya ke dalam PLC
- Membuat program PLC dengan menggunakan instruksi math



CPT – Compute (1)

- CPT – Compute digunakan untuk menghitung secara general dua nilai dari source
- Penghitungan yang dilakukan tergantung dari expression yang digunakan
- Expression yang digunakan terdiri dari :
  - Penjumlahan +
  - Pengurangan –
  - Pengalian *
  - Pembagian |
  - Negatif -
  - Pengakaran SQR
  - Eksponensial **

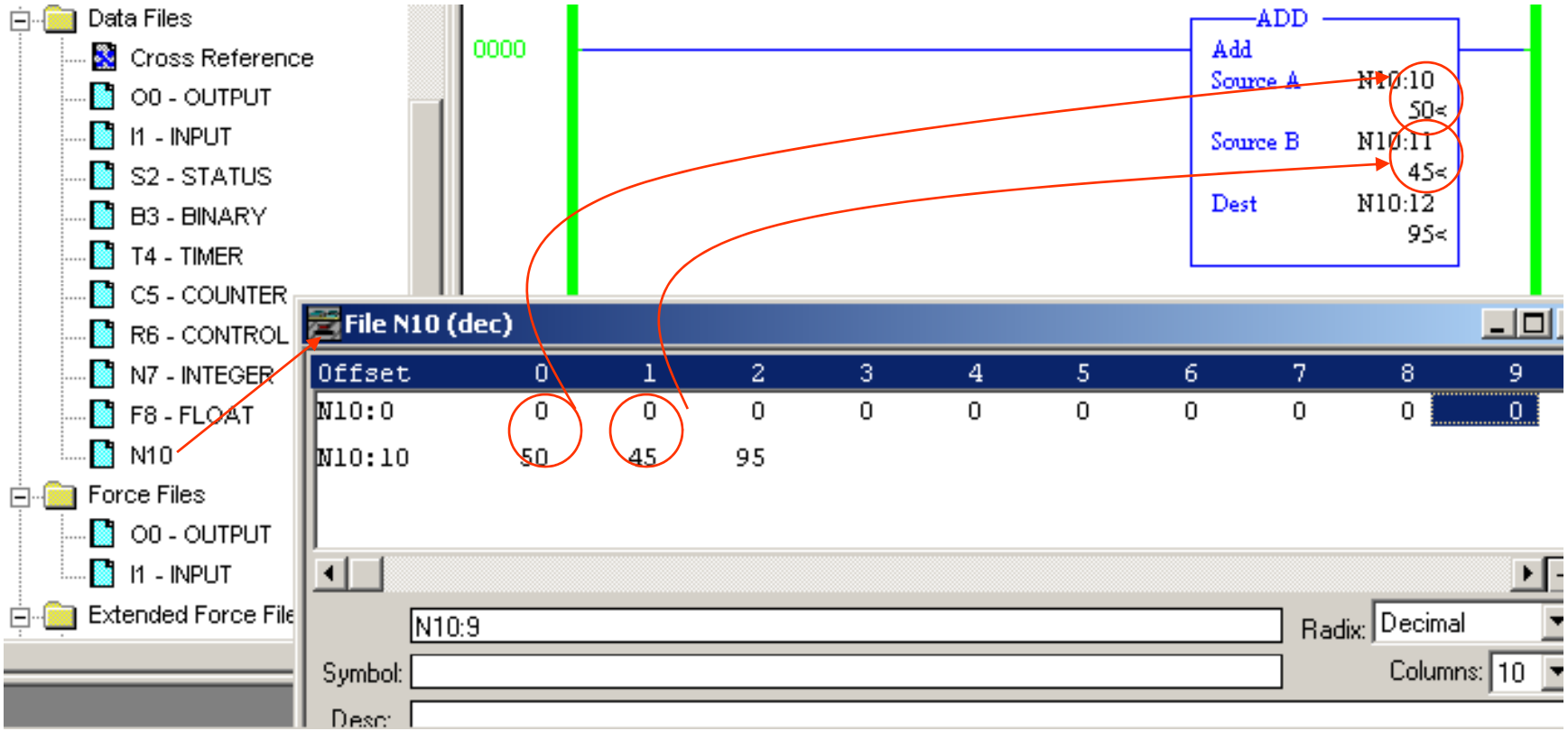


CPT – Compute (1)

- Bisa Juga untuk operasi-operasi trigonometri, operasi-operasinya meliputi :
  - ARC – Arc Cosinus
  - ASN – Arc Sinus
  - ARN – Arc Tangent
  - COS – Cosinus
  - SIN – Sinus
  - TAN - Tangent Bitwise :
  - AND – Bitwise and
  - OR – Bitwise or
  - XOR – Bitwise xor
  - NOT – Bitwise not
- Conversion :
  - FRD – Convert from BCD to Binary
  - TOD – Convert from Binary to BCD
  - DEG – Convert from radians to degree
  - RAD – Convert from degree to radians

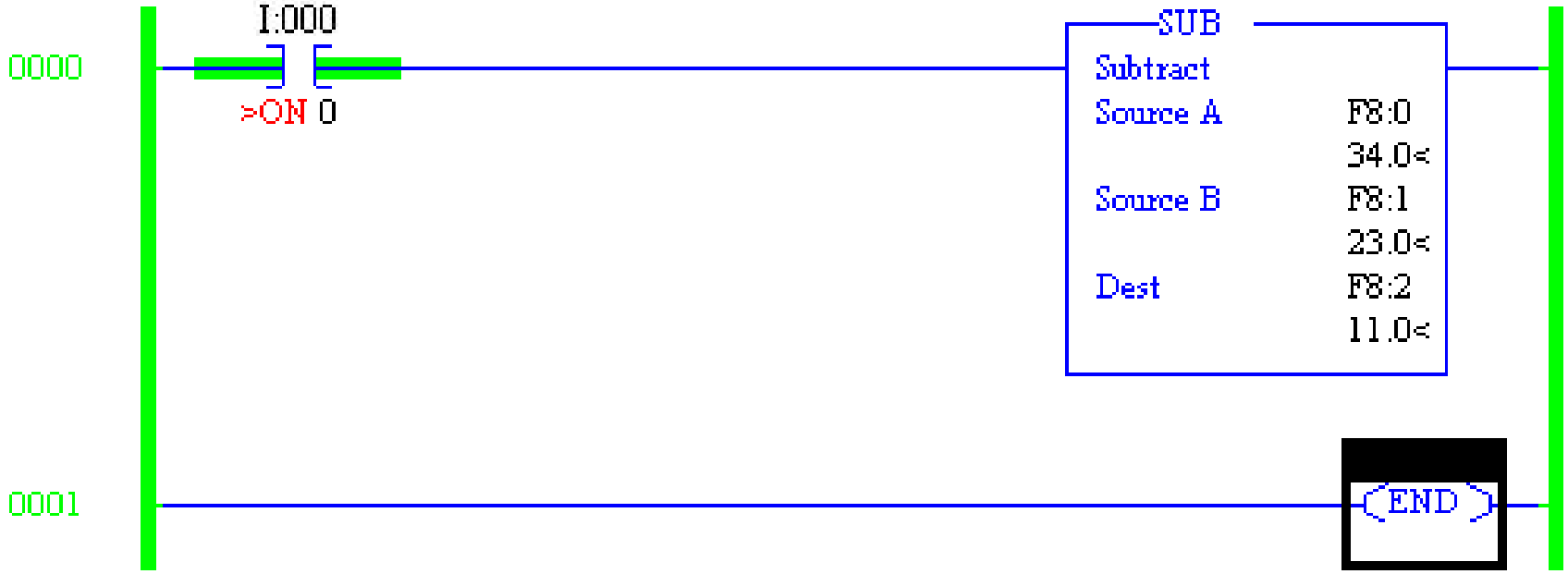
ADD – Add

- ADD – Add, operator yang berfungsi untuk menjumlahkan nilai dari source A dengan nilai di source B
- Contoh Pemrograman ADD :



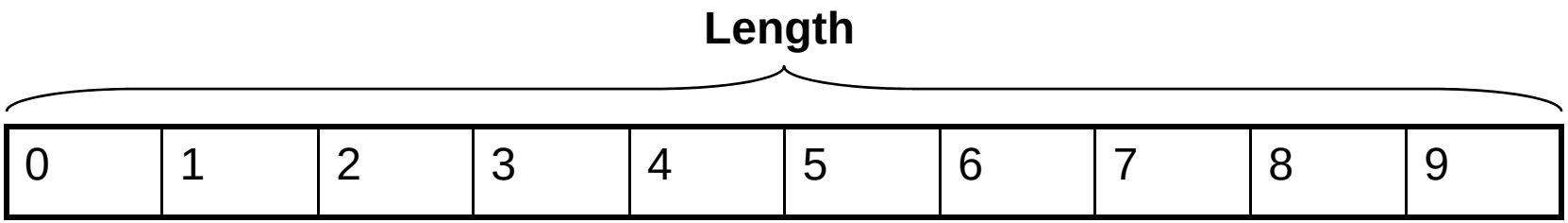
SUB – Subtract

- SUB - Subtract, SUB menghitung operasi pengurangan dari data source A oleh data source B, kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh pemrograman SUB :



AVE - Average

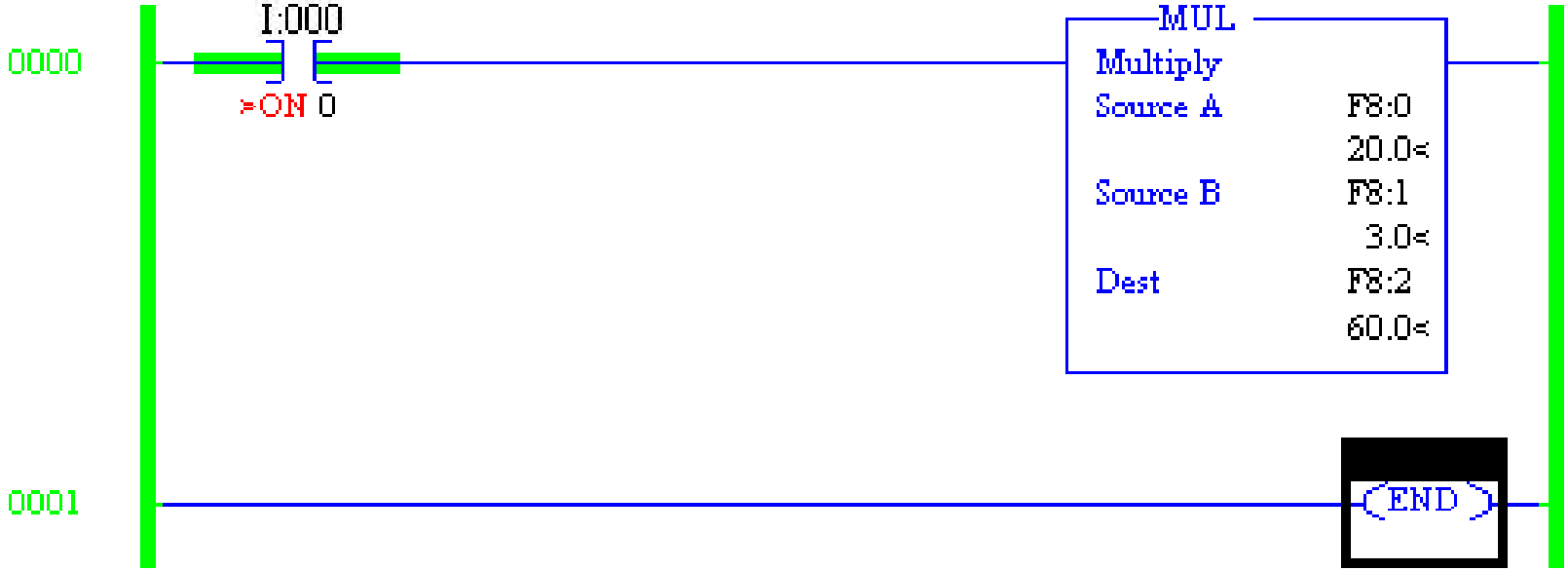
- AVE – Average, AVE digunakan untuk menghitung nilai rata-rata dari satu set kumpulan nilai, nilai dijumlahkan mulai dari posisi yang ditentukan sampai batas terakhir length, hasil penjumlahan kemudian dirata-rata dan disimpan di file tujuan (destination)



N7:1 = 10 (nilai)
Posisi = 0
Length = 9 (pembagi)
N7:0 = 1 (hasil)

MUL – Multiple

- MUL – Multiple, MUL digunakan untuk mengoperasikan perkalian pada data source A dengan data source B kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman MUL :

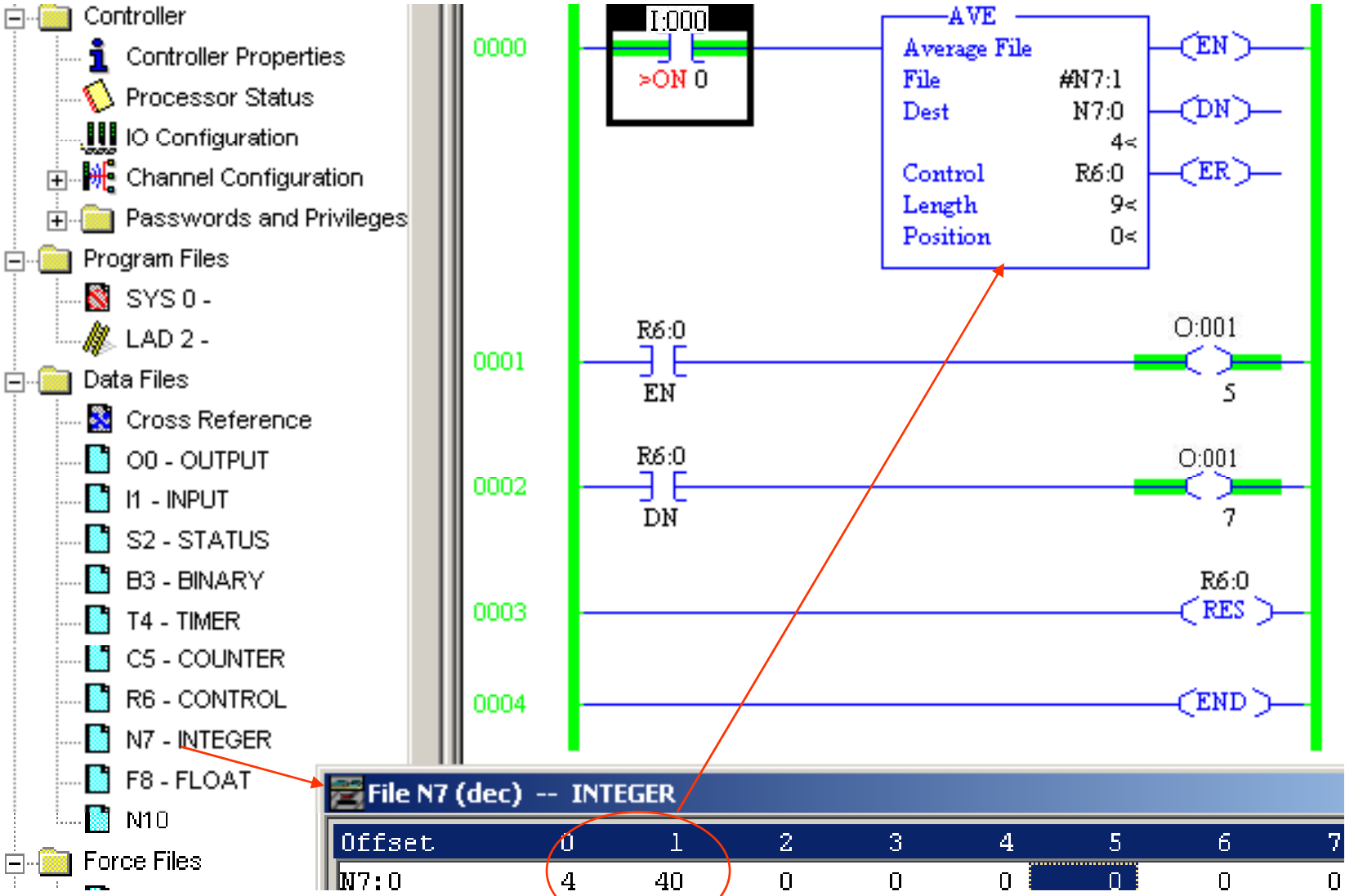


NEG - Negate

- NEG – Negate, digunakan untuk mengoperasikan negasi pada data source kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman NEG :

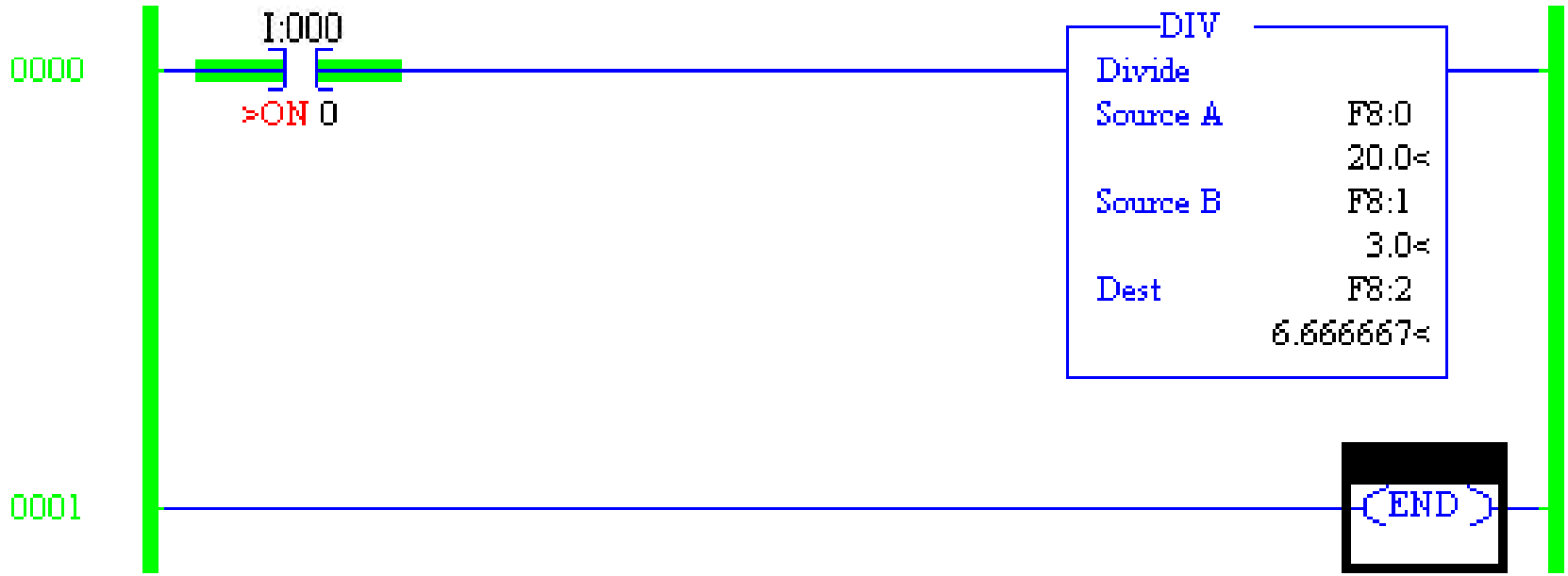


Contoh Pemrograman AVE



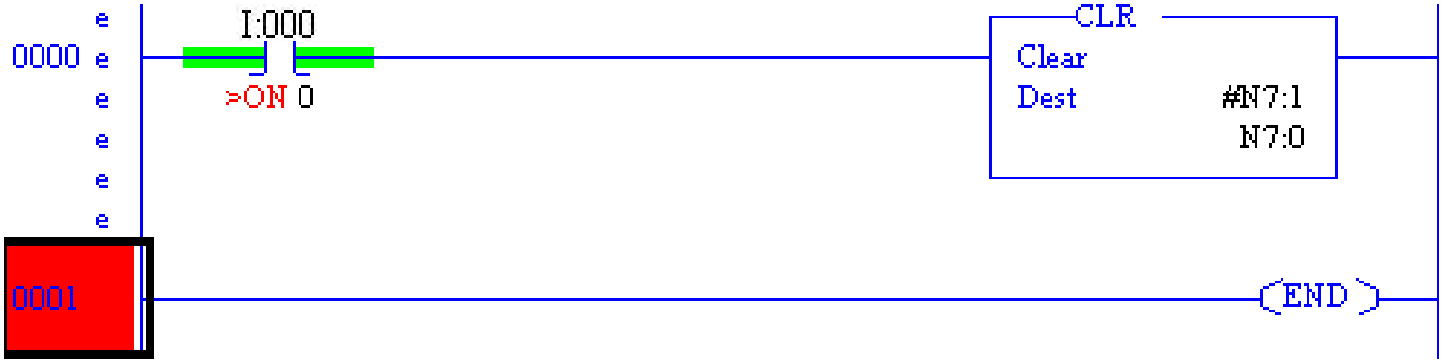
DIV – Divide

- DIV – Divide, DIV digunakan untuk membagi data source A dengan data source B kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman DIV :



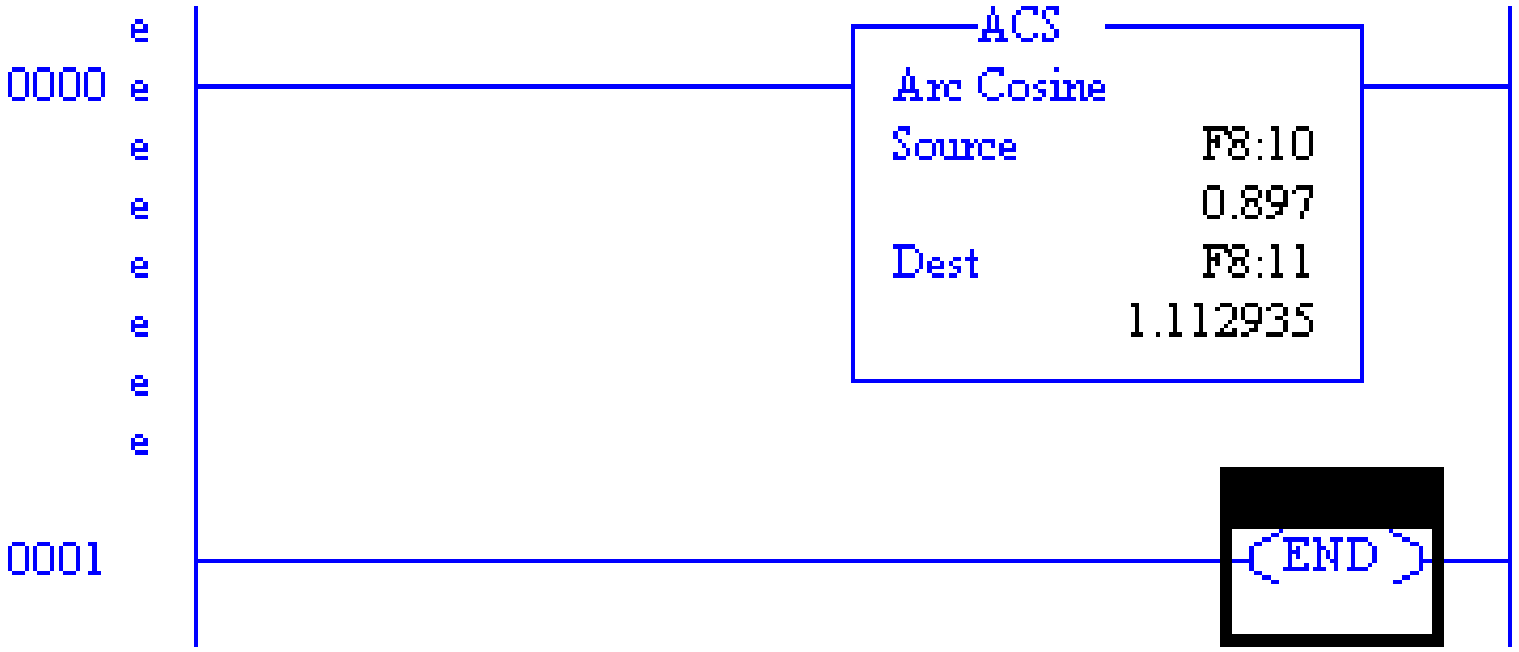
CLR – Clear

- CLR – Clear, CLR digunakan untuk menset semua bit ke nol
- Contoh pemrograman CLR :



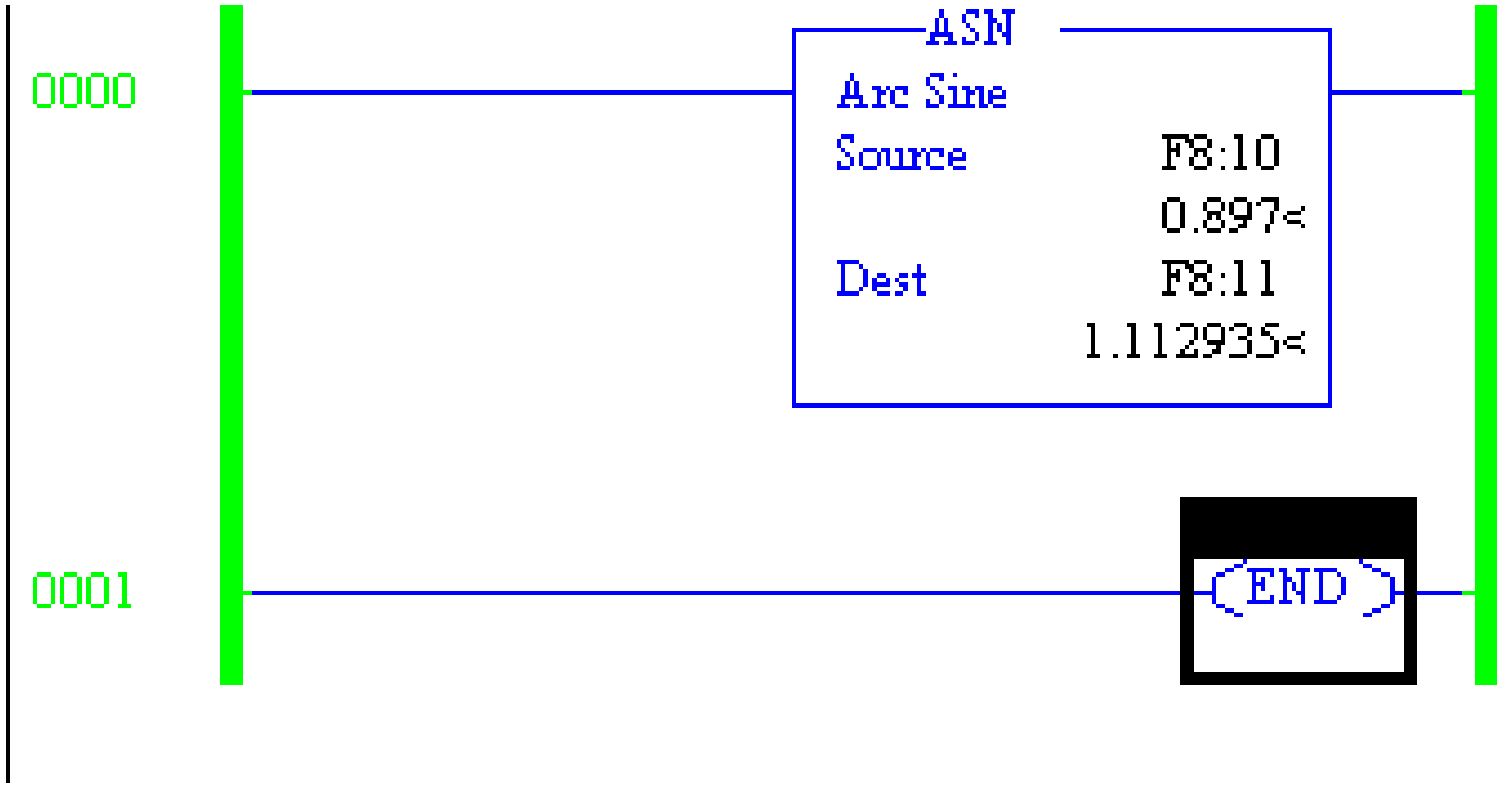
ACS – Arc Cosinus

- ACS – Arc Cosinus, digunakan untuk menghitung arc cosinus di source file dan menyimpan hasilnya di memori tujuan (destination)
- Contoh pemrograman ARC :



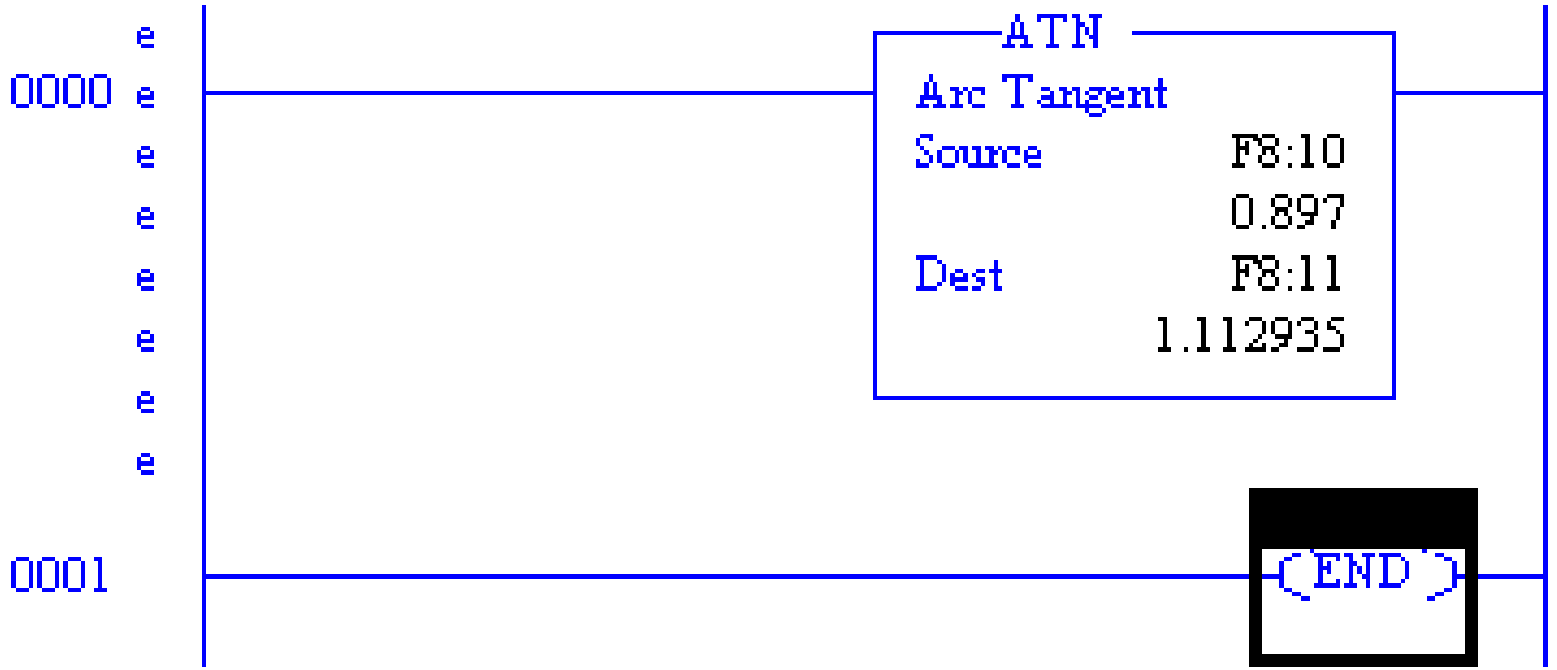
ASN – Arc Sinus

- ASN – Arc Sinus, ASN digunakan untuk mengoperasikan arc sinus dari data source A kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman ASN :



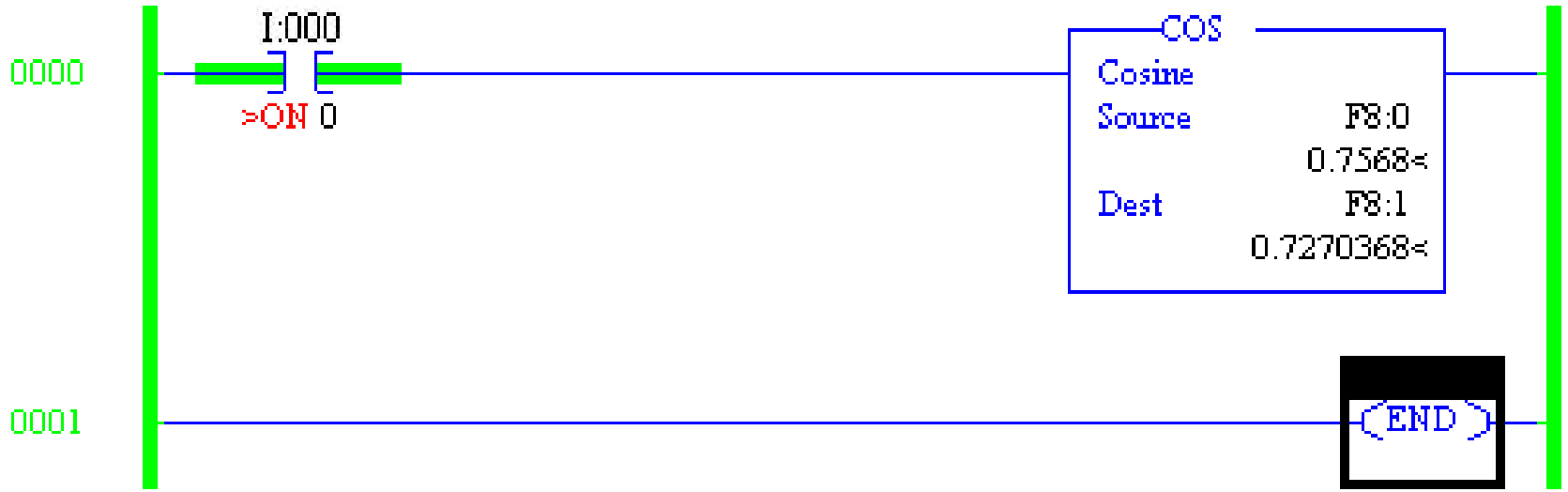
ATN – Arc Tangent

- ATN – Arc Tangent, ATN digunakan untuk mengoperasikan arc tangent dari data source A kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman ATN :



COS – Cosinus

- COS – Cosinus, COS digunakan untuk mengoperasikan Cosinus dari data source kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman COS :



SIN – Sine

- SIN – Sine, SIN digunakan untuk mengoperasikan sinus dari data source kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman SIN :



LN – Natural Log

- LN – Natural Log, LN digunakan untuk mengoperasikan natural log dengan data source kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman LN :



LOG - Logarithm

- LOG – Logarithm, LOG digunakan untuk mengoperasikan logaritma pada data source kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman LOG :



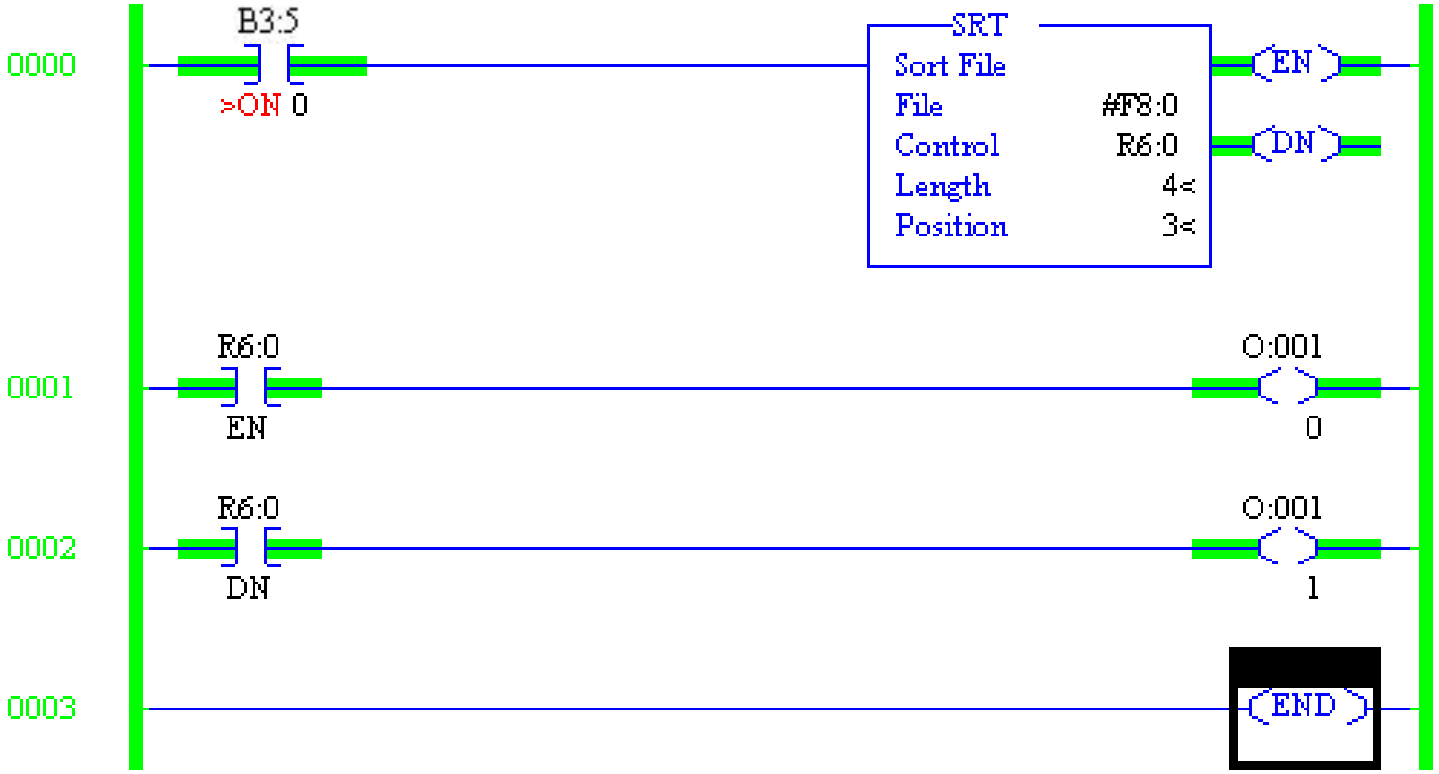
SQR – Square Root

- SQR – Square Root, SQR digunakan untuk mengoperasikan akar kuadrat dari data source kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman SQR :



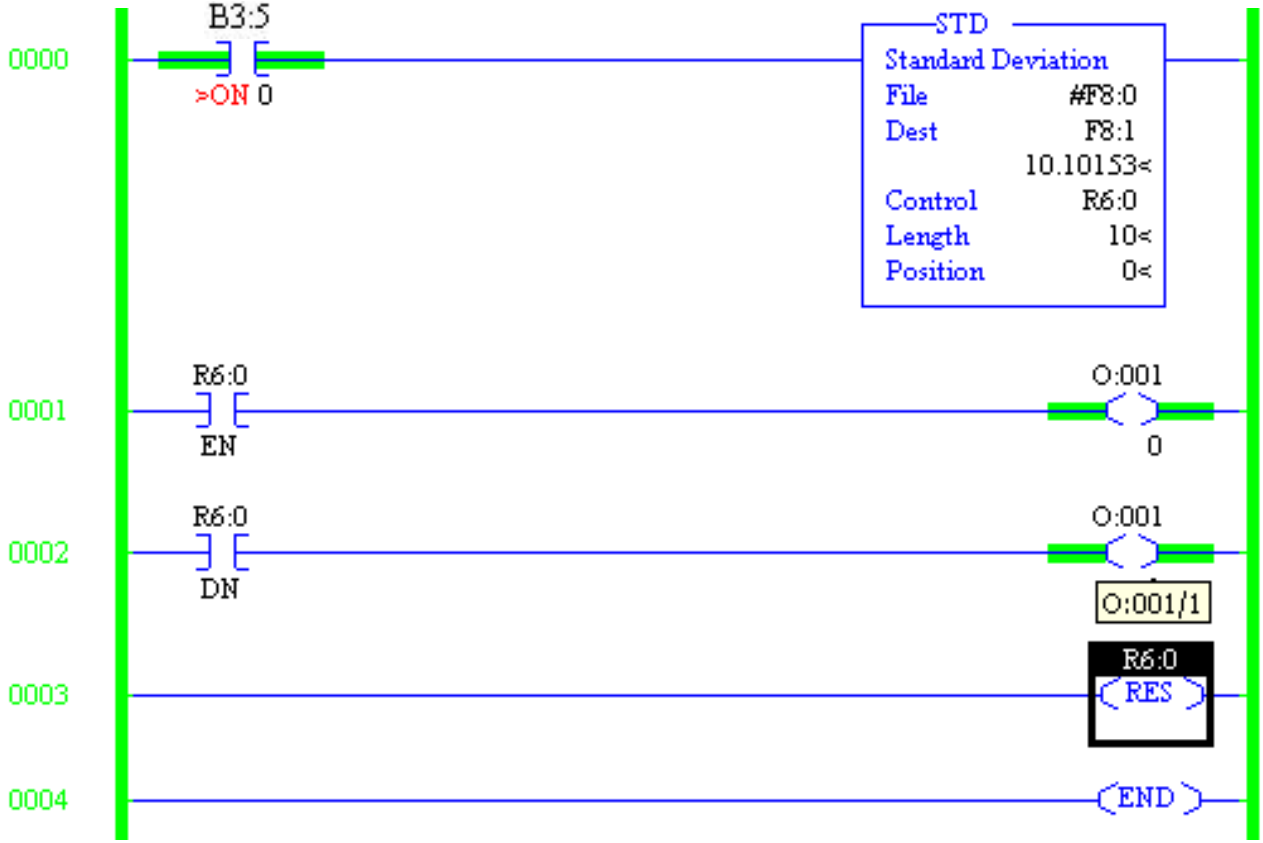
SRT – Sort File

- SRT – Sort File, operasi ini digunakan untuk mengatur kumpulan nilai dari suatu set data dalam suatu urutan, dari yang terkecil manjadi yang terbesar
- Contoh pemrograman SRT :



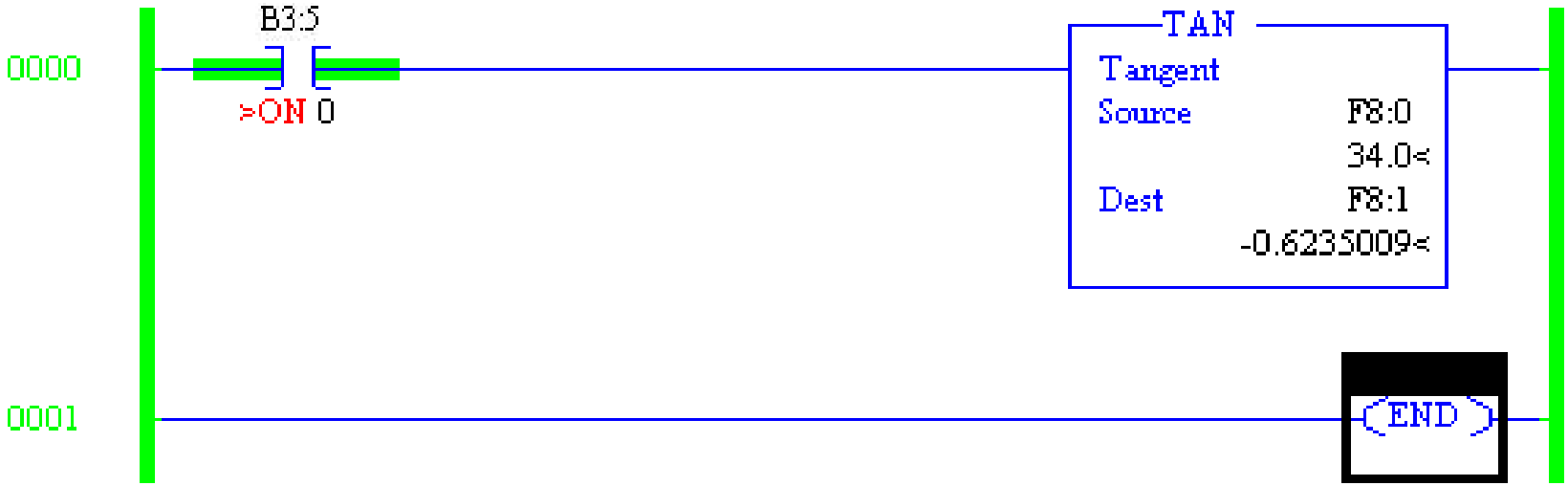
STD – Standard Deviation

- STD – Standard Deviation, STD menghitung deviasi standar dari sejumlah data dari source, kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh pemrograman STD :



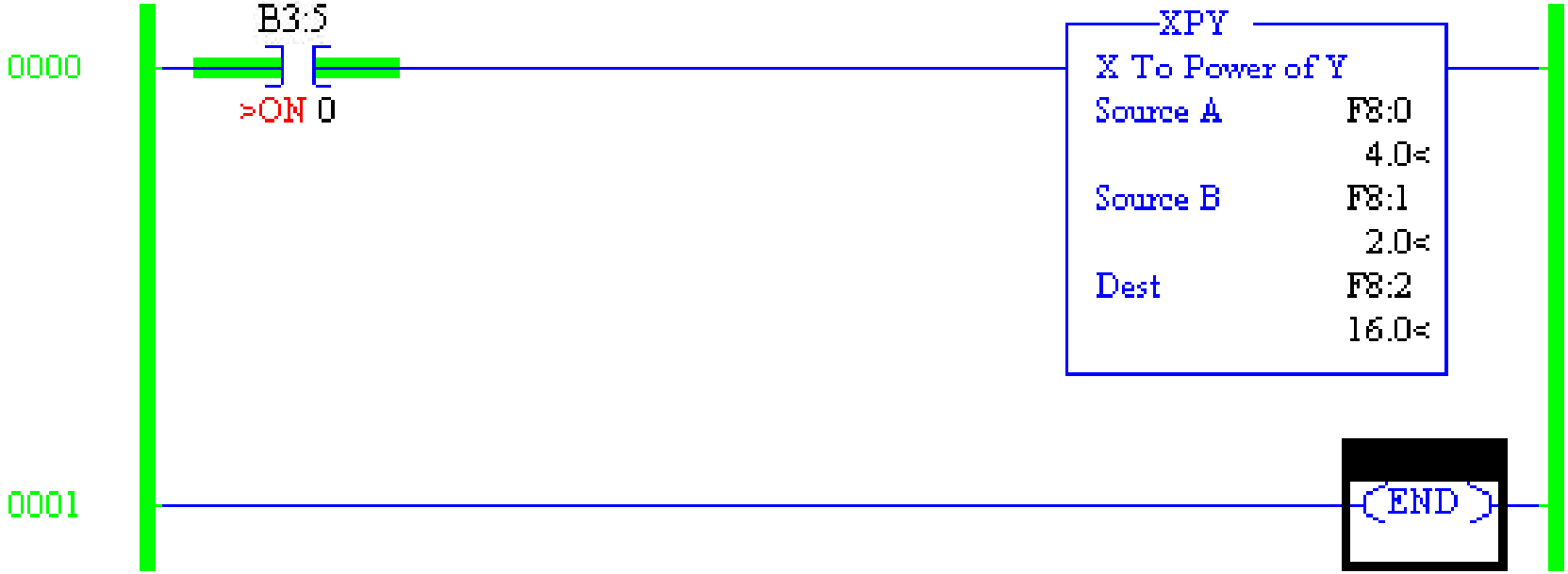
TAN - Tangent

- TAN – Tangent, TAN digunakan untuk mengoperasikan tangent dari data source kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman TAN :



XPY – X To Power of Y

- XPY – X To Power of Y, XPY digunakan untuk mengoperasikan X pangkat Y, data source A adalah X sedangkan data source B adalah Y kemudian menyimpan hasilnya di file tujuan (destination)
- Contoh Pemrograman XPY :

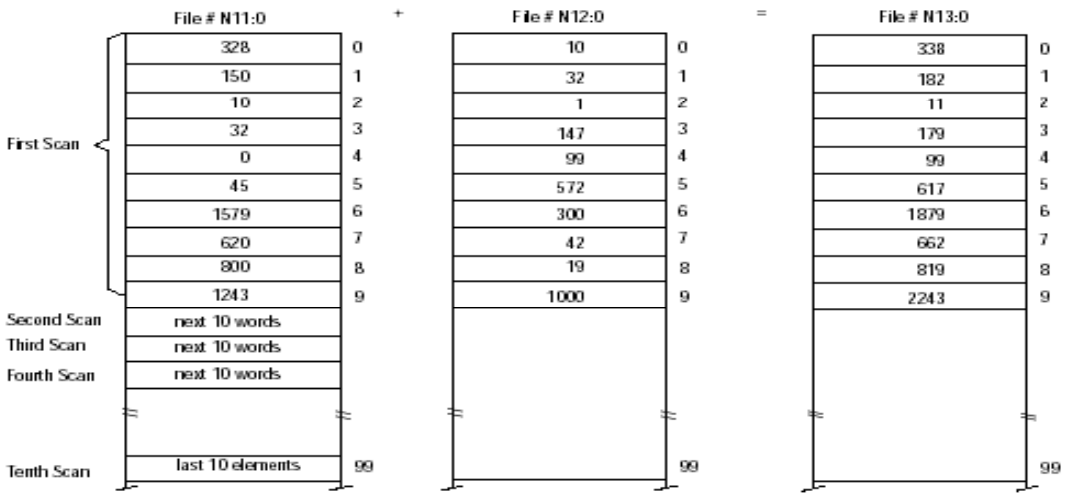


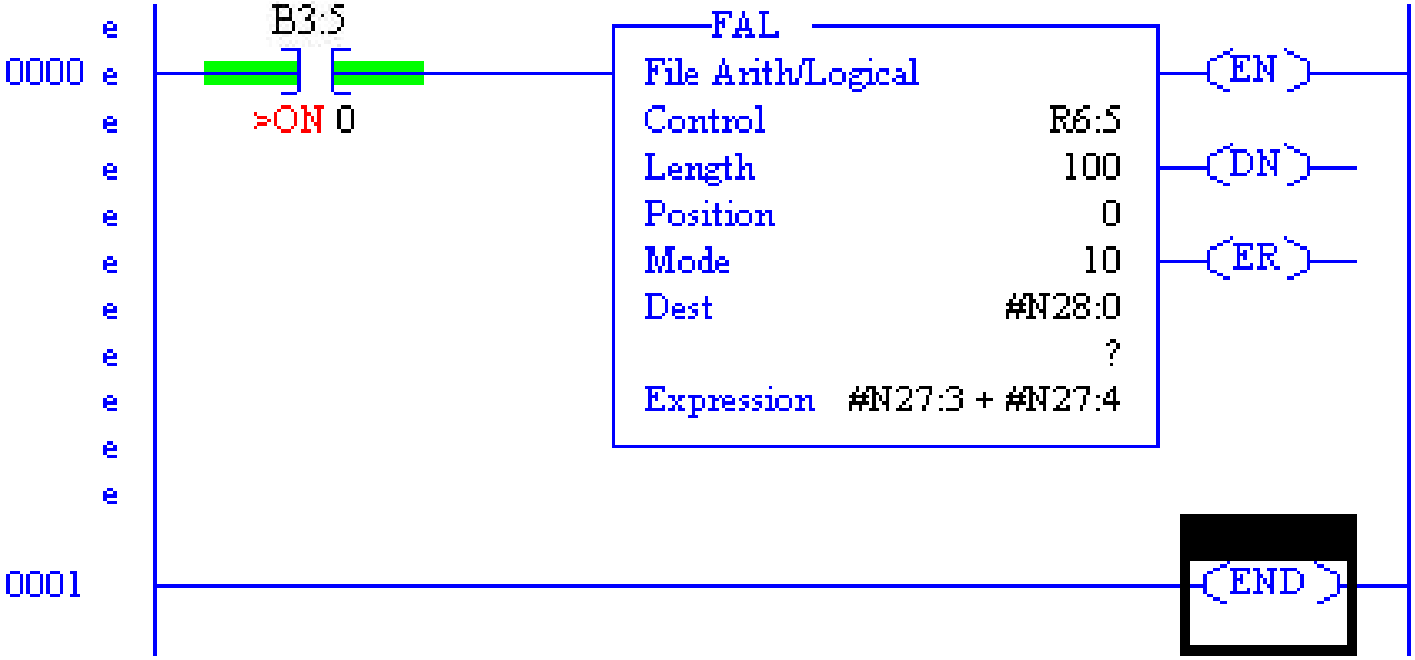
FAL Arithmetic Operations

Operator:	Meaning:	Operator:	Meaning:
+	add	|	divide
–	subtract	–	negate
*	multiply	0	clear

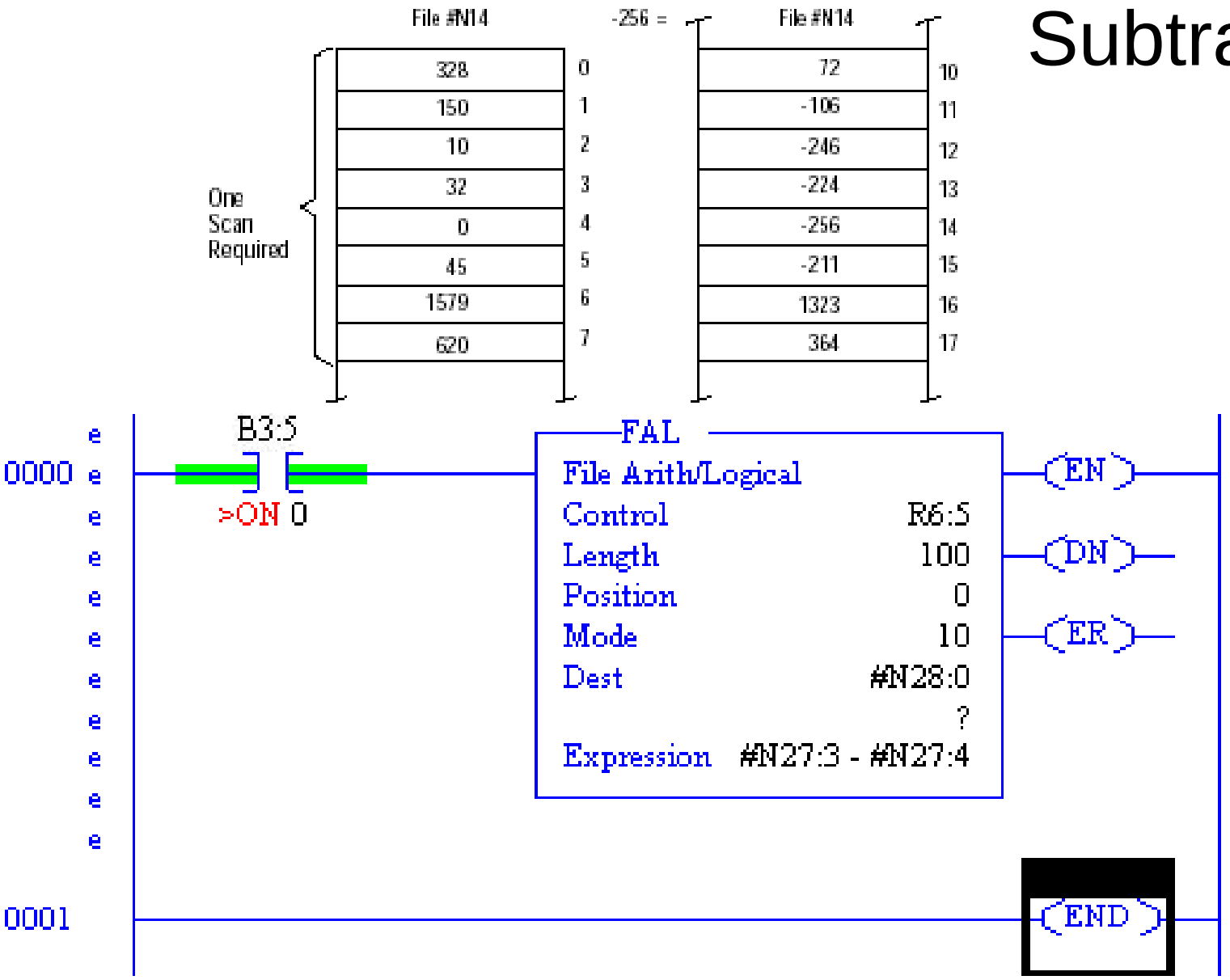
Type of File:	Range Stored in Word:
bit	–32,768 to +32,767 for integers
integer	–32,768 to +32,767
timer	0 to +32,767
counter	–32,768 to +32,767
control	0 to +32,767
floating point	$\pm 1.1754944e^{-38}$ to $\pm 3.4028237e^{+38}$

Add

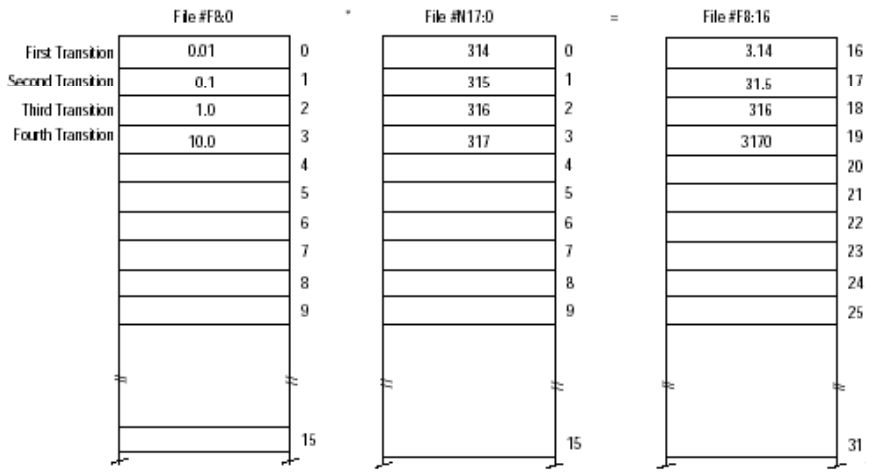

File # N11:0
File # N12:0
File # N13:0
First Scan
Second Scan
Third Scan
Fourth Scan
Tenth Scan
next 10 words
last 10 elements
328 150 10 32 0 45 1579 620 800 1243
10 32 1 147 99 572 300 42 19 1000
338 182 11 179 99 617 1879 662 819 2243

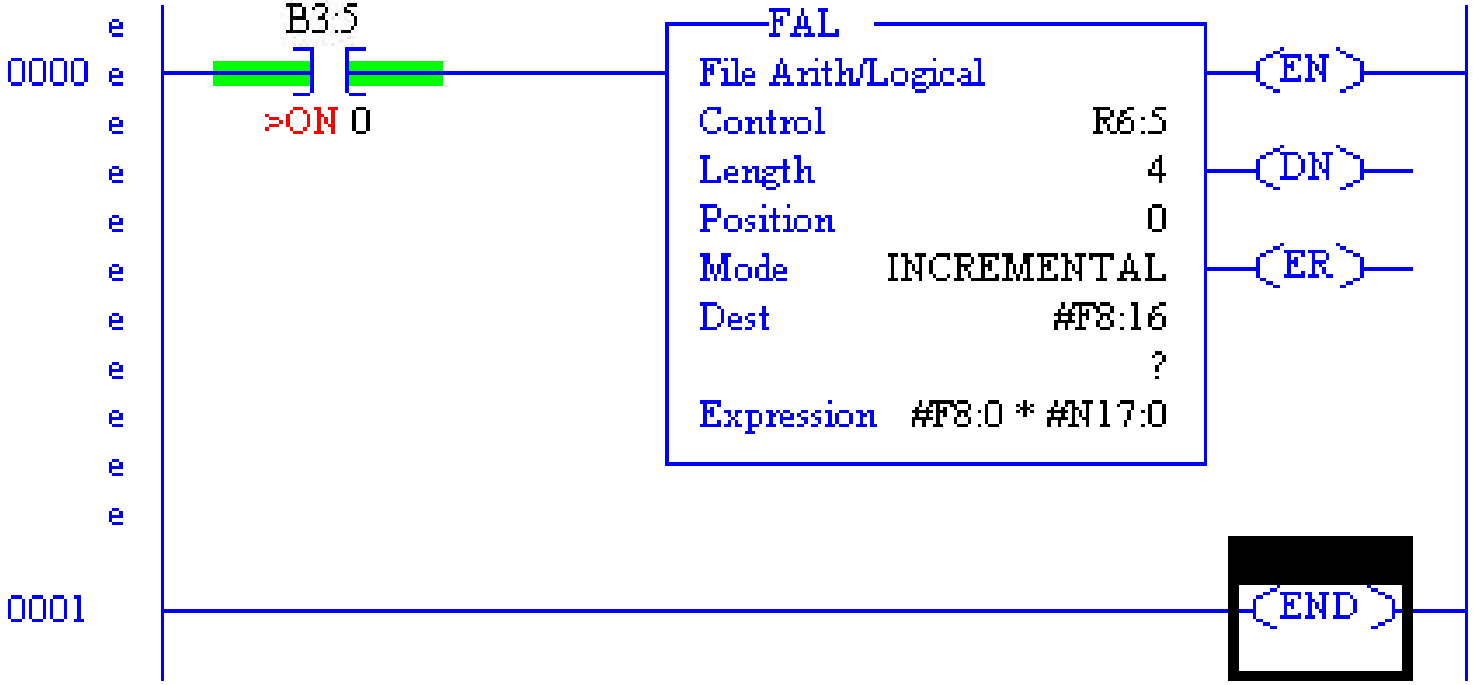

B3:5
>ON 0
FAL
File Arith/Logical
Control R6:5
Length 100
Position 0
Mode 10
Dest #N28:0
?
Expression #N27:3 + #N27:4
EN
DN
ER
0000
0001
END

Subtract

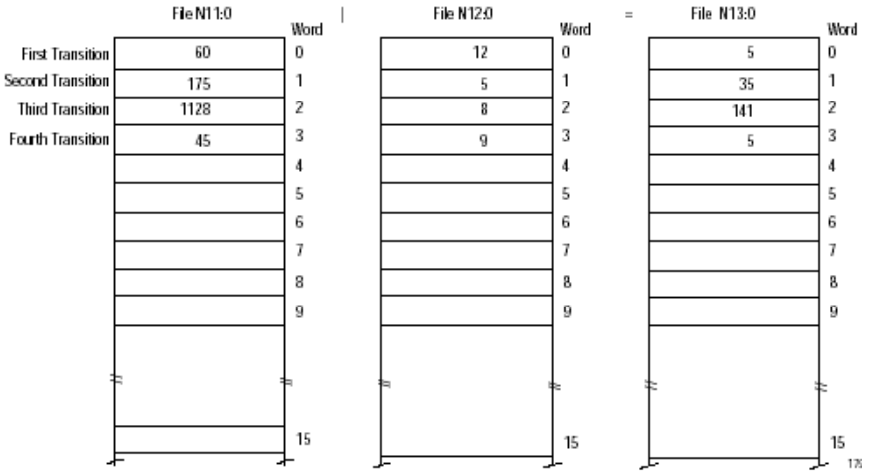

File #N14
-256 =
File #N14
One Scan Required
328
150
10
32
0
45
1579
620
72
-106
-246
-224
-256
-211
1323
364
0
1
2
3
4
5
6
7
10
11
12
13
14
15
16
17
B3:5
>ON 0
FAL
File Arith/Logical
Control R6:5
Length 100
Position 0
Mode 10
Dest #N28:0
?
Expression #N27:3 - #N27:4
EN
DN
ER
0000
0001
END

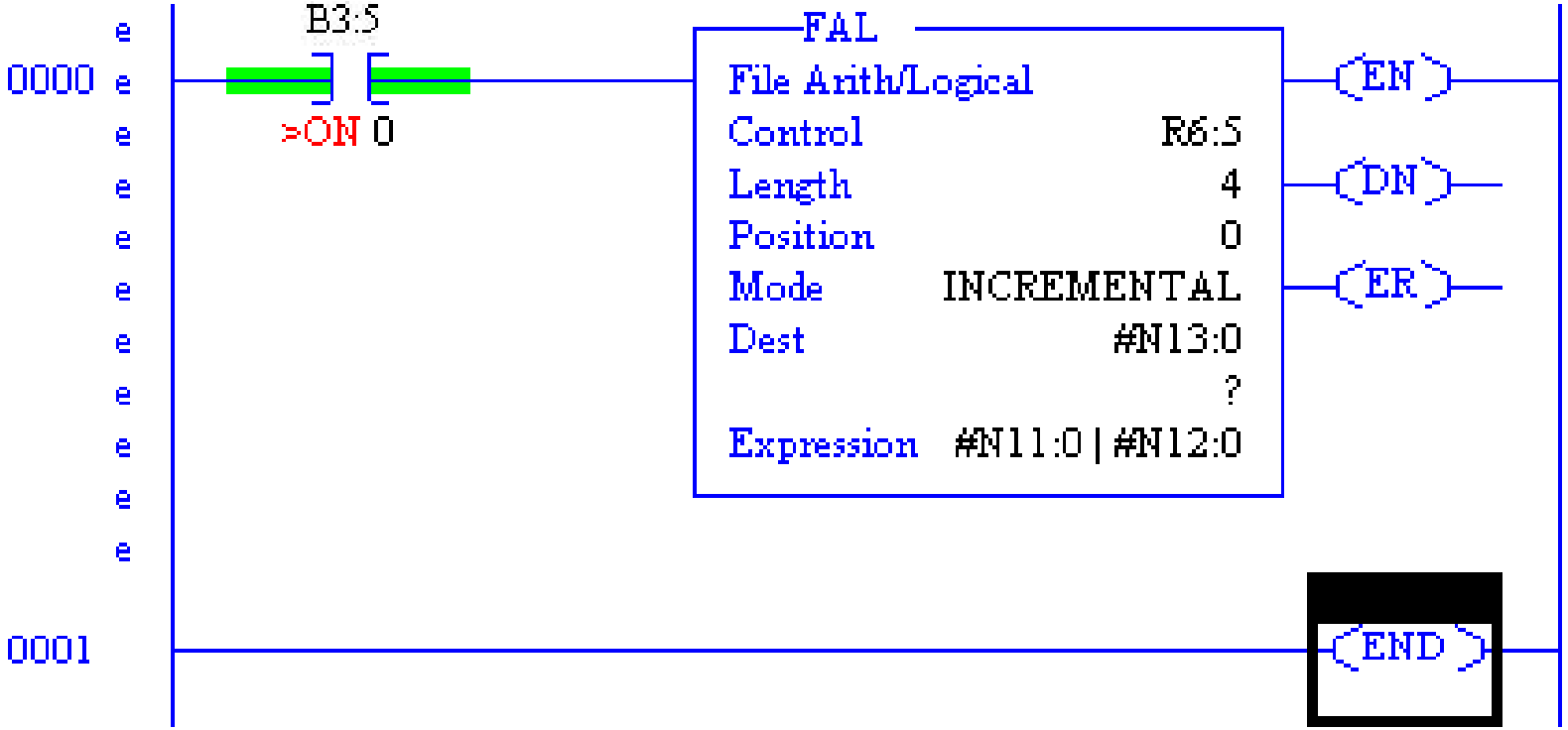
Multiple


File #F8:0
First Transition
Second Transition
Third Transition
Fourth Transition
0.01
0.1
1.0
10.0
File #N17:0
314
315
316
317
File #F8:16
3.14
31.5
316
3170

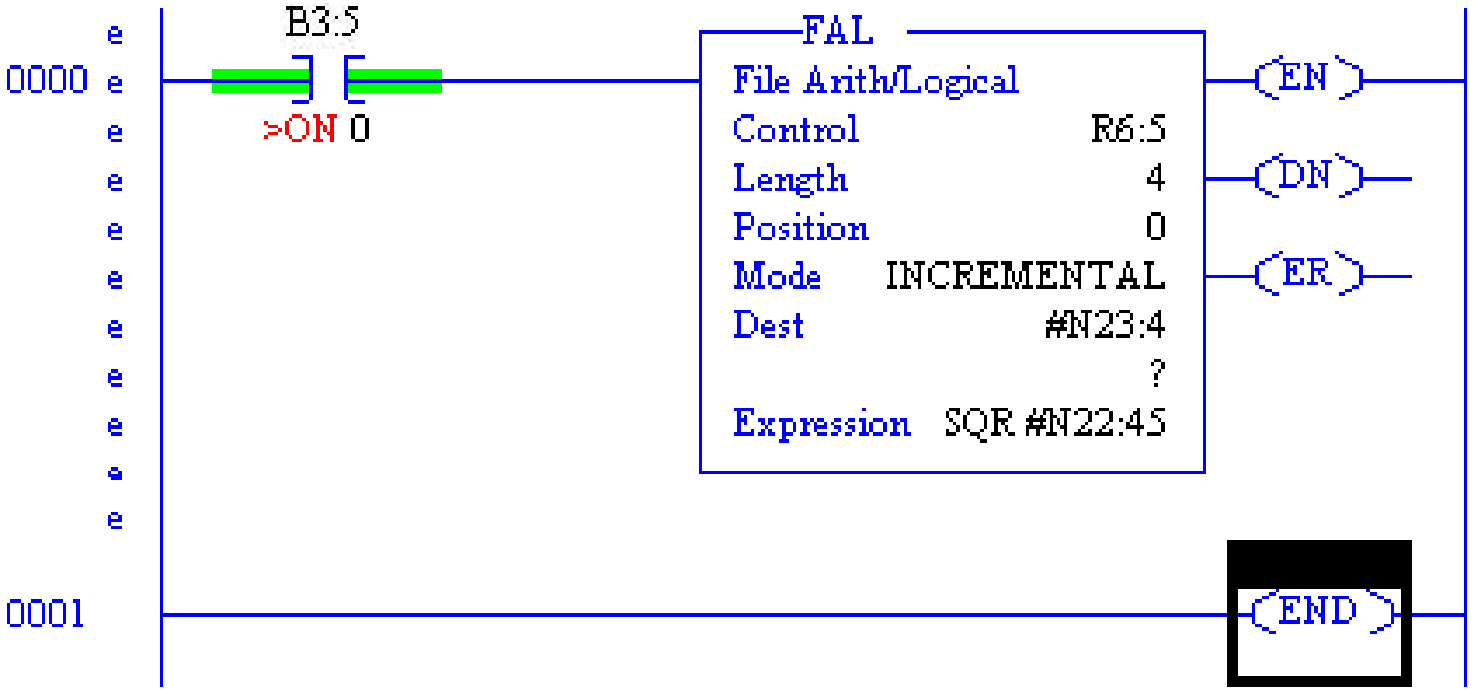

B3:5
>ON 0
FAL
File Arith/Logical
Control R6:5
Length 4
Position 0
Mode INCREMENTAL
Dest #F8:16
?
Expression #F8:0 * #N17:0
EN
DN
ER
0000
0001
END

Divide


File N11:0
Word
First Transition 60 0
Second Transition 175 1
Third Transition 1128 2
Fourth Transition 45 3
4
5
6
7
8
9
15
File N12:0
Word
12 0
5 1
8 2
9 3
4
5
6
7
8
9
15
File N13:0
Word
5 0
35 1
141 2
5 3
4
5
6
7
8
9
15


B3:5
>ON 0
FAL
File Arith/Logical
Control R6:5
Length 4
Position 0
Mode INCREMENTAL
Dest #N13:0
?
Expression #N11:0 | #N12:0
EN
DN
ER
0000
0001
END

Square Root

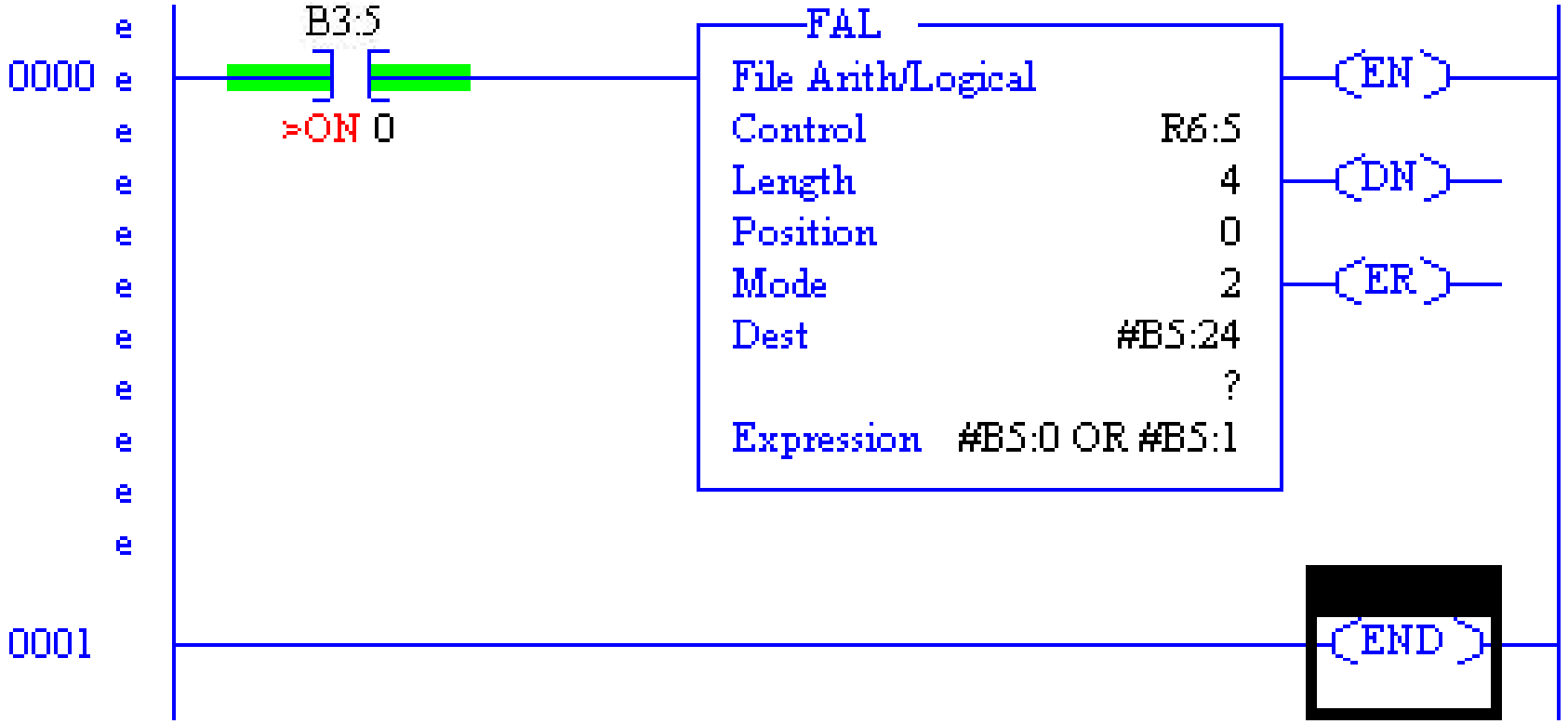

B3:5
>ON 0
FAL
File Arith/Logical
Control R6:5
Length 4
Position 0
Mode INCREMENTAL
Dest #N23:4
?
Expression SQR #N22:45
EN
DN
ER
0000
0001
END

FAL Logic Operation

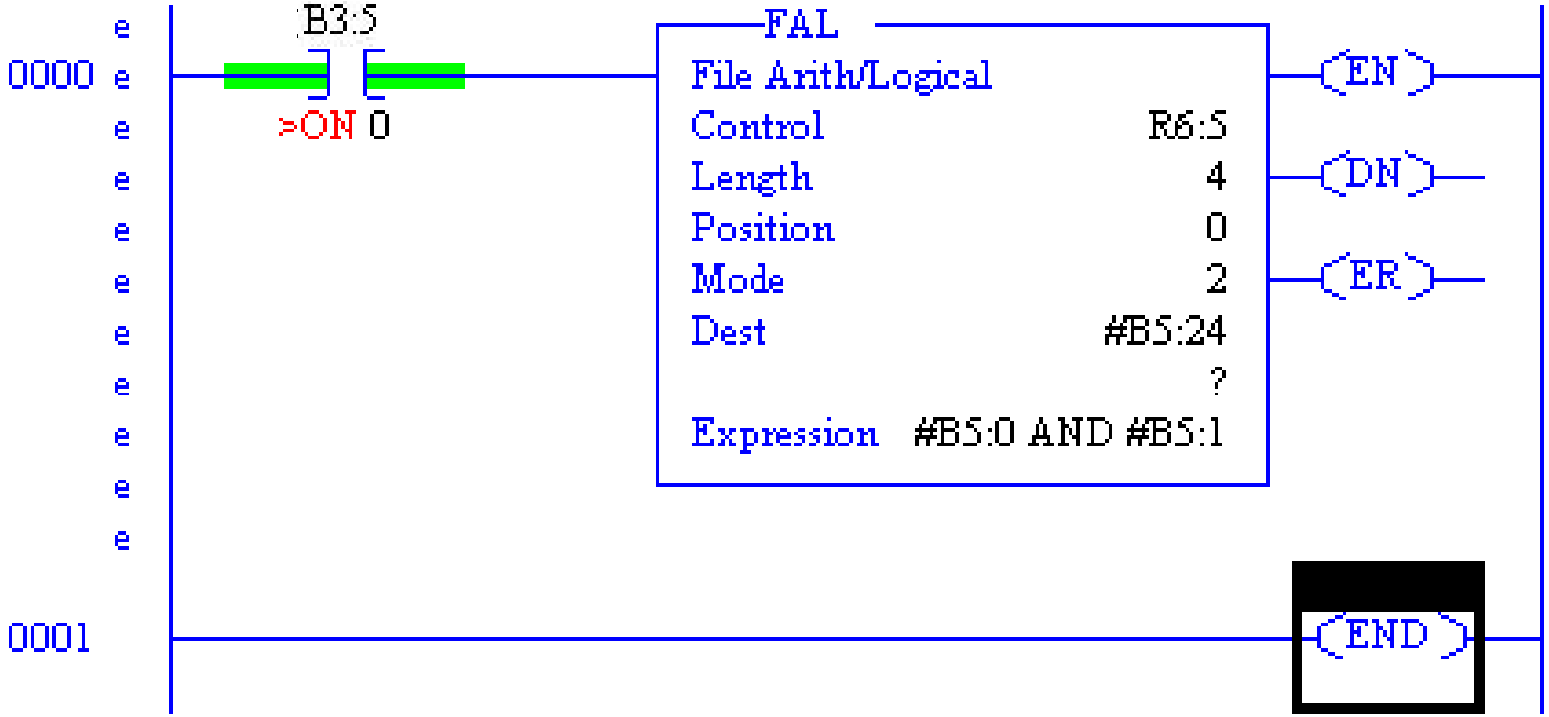
- AND
- OR
- XOR
- NOT

OR

	File I:000	Word	or	File B3	Word	=	File B5	Word
First Scan	0000000000000000	0		1010101010101010	6		1010101010101010	24
	1111111111111111	1		1111111100000000	7		1111111111111111	25
Second Scan	1111000011110000	2		0000000000000000	8		1111000011110000	26
	1010101010101010	3		1100110011001100	9		1110111011101110	27
Third Scan		4			10			28
		5			11			29



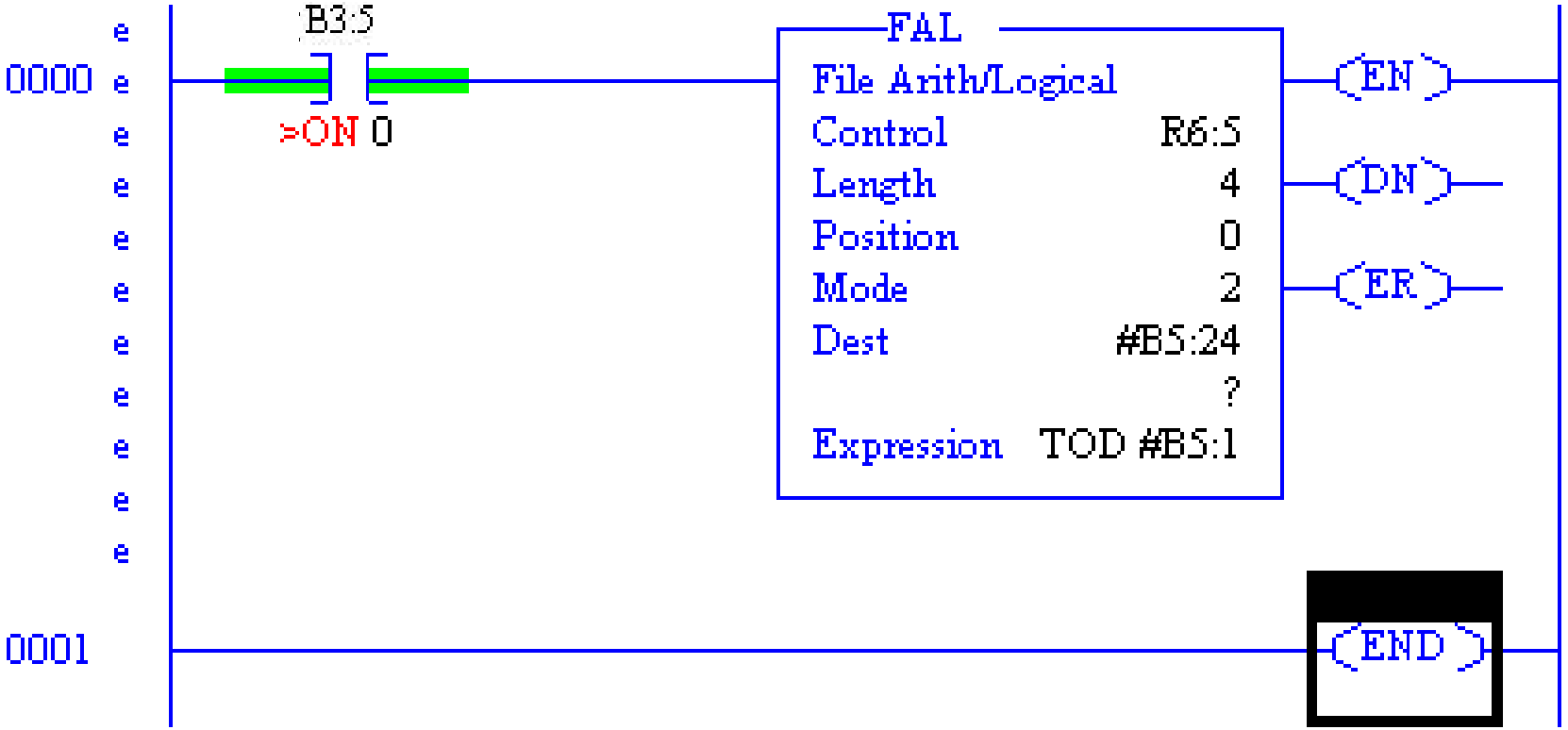
AND


B3:5
>ON 0
FAL
File Arith/Logical
Control R6:5
Length 4
Position 0
Mode 2
Dest #B5:24
?
Expression #B5:0 AND #B5:1
EN
DN
ER
0000
0001
END

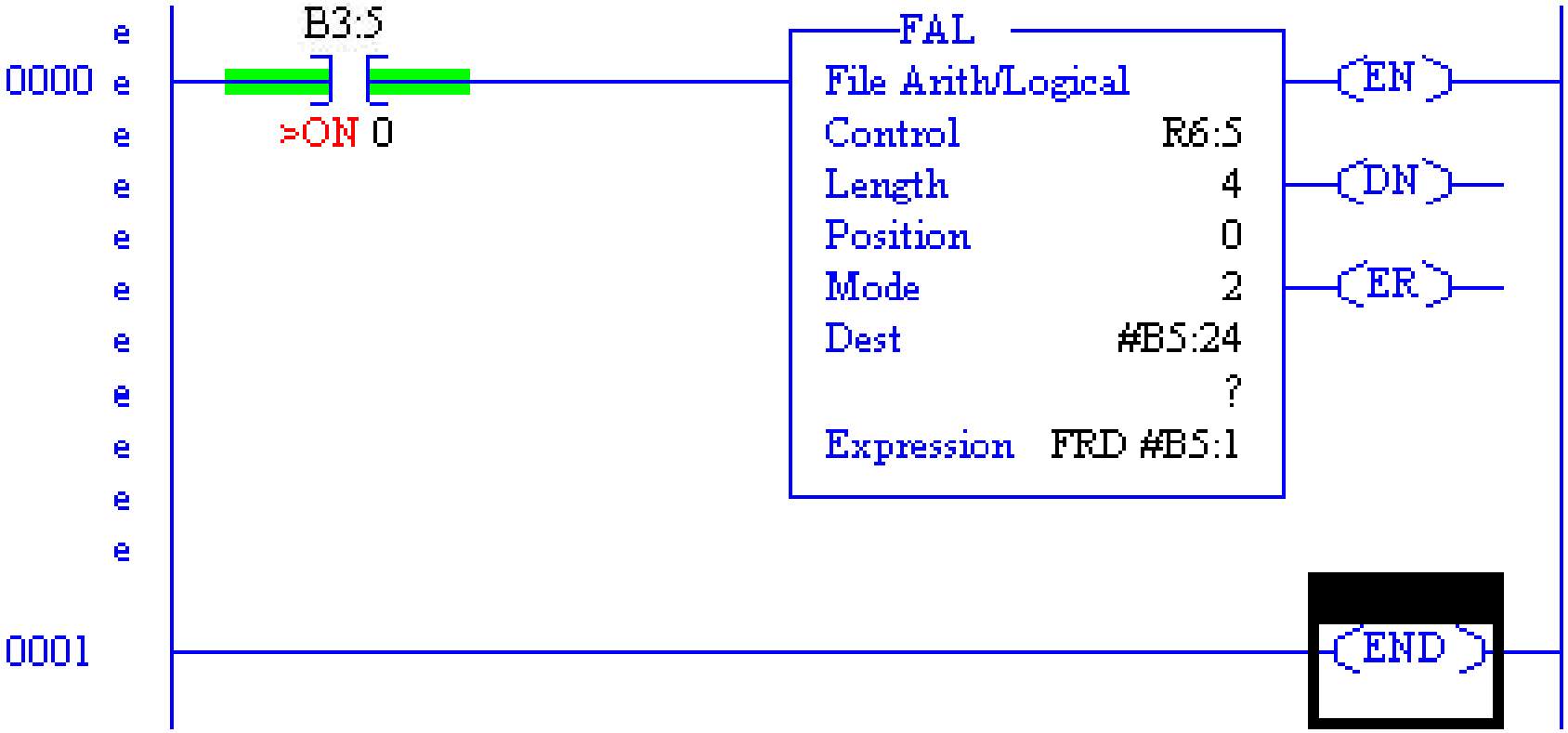
FAL Convert Operation

- TOD – Convert to BCD
- FRD – Convert from BCD

TOD – Convert to BCD



FRD – Convert from BCD

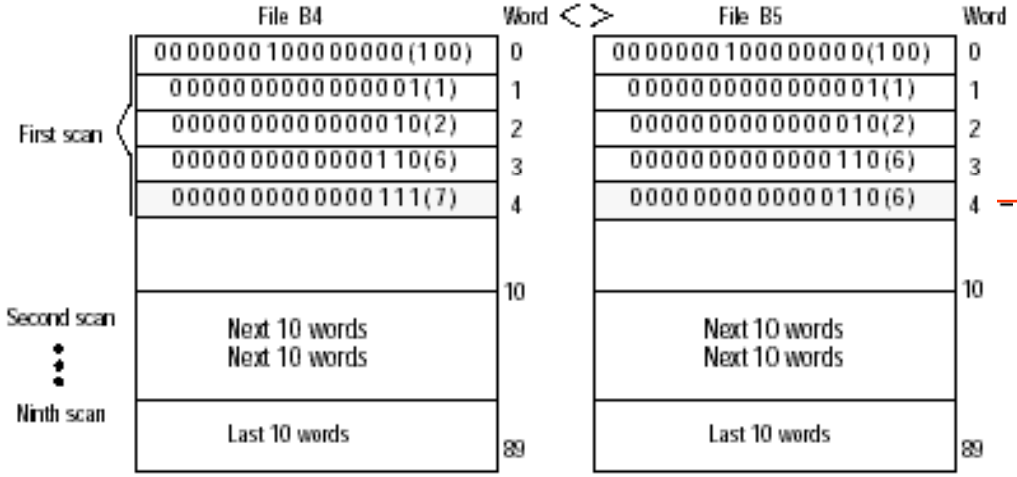


FSC – File Search and Compare

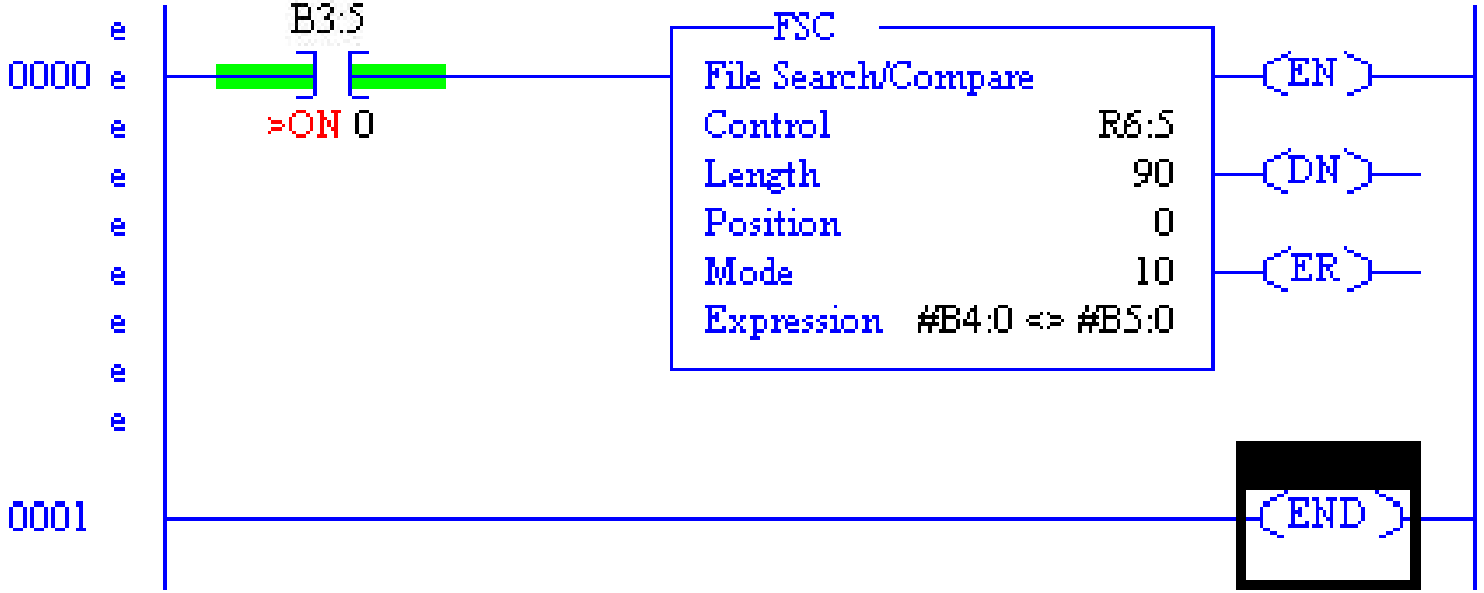
- Instruksi FSC memungkinkan operasi pencarian dan pembandingan (search and compare) hampir sama dengan operasi CMP termasuk ekspresi kompleksnya. Perbedaannya adalah instruksi FSC bisa mengoperasikan persamaan logika pada file-file sementara CMP hanya bisa beroperasi pada word tunggal. Juga instruksi FSC adalah instruksi output sementara CMP adalah instruksi input

Comparison:	Example Expression:
Search Equal	#N50:0 = #N51:0
Search Not Equal	#N52:0 <> N52:11
Search Less Than	#B3:100 < #N53:0
Search Less Than or Equal	#F60:0 <= F60:12
Search Greater Than	#N54:0 > 256
Search Greater Than or Equal	F60:10 >= #N61:0

Contoh Search not Equal

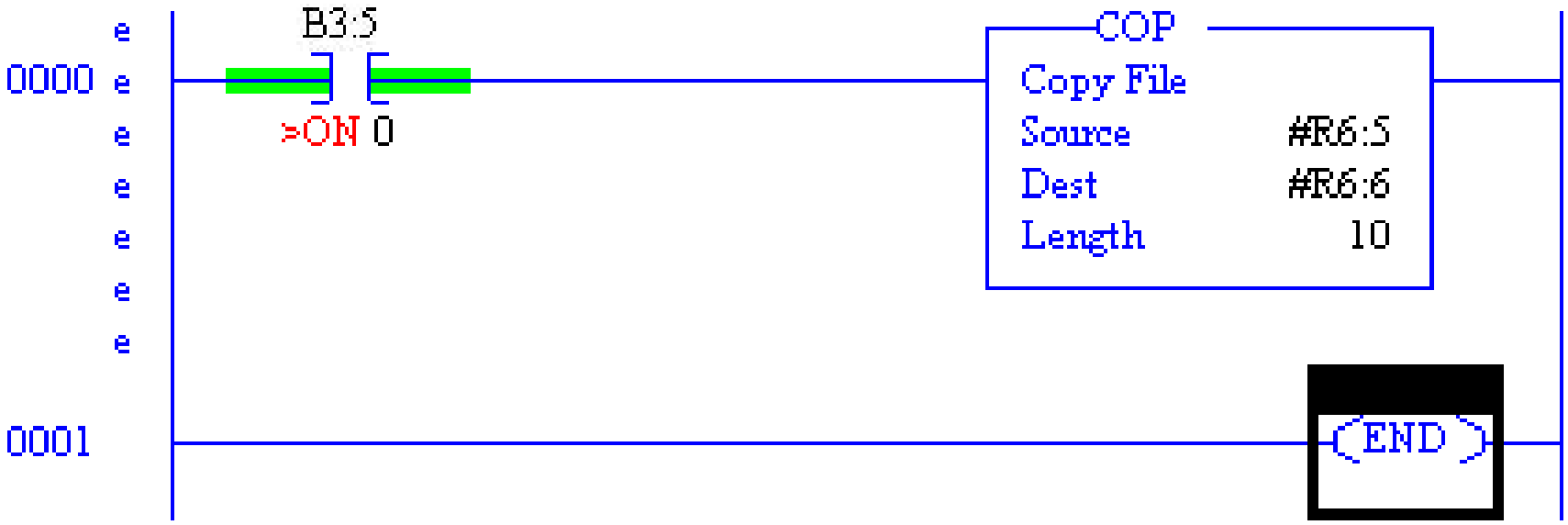


Processor berhenti dan men-set bit found dan inhibit, untuk meneruskan program reset inhibit bit



COP – Copy File

- COP – Copy File, Instruksi COP berfungsi untuk mengkopi suatu file danmenyimpannya di file tujuan (destination)
- Contoh Program COP :



FLL - File Fill

- FLL - File Fill, Instruksi FLL adalah instruksi output digunakan untuk mengisi file dengan nilai yang berasal dari source. Instruksi FLL tidak menggunakan status bit. Jika perlu, programlah parallel output yang menggunakan alamat penyimpanan
- Instruksi FLL tidak menulis file secara parsial. Data yang tersimpan akan hilang. Jika file tujuan berupa word (seperti integer file) harus dispesifikasikan dulu panjang dari strukturnya.
- Contoh Program FLL :

