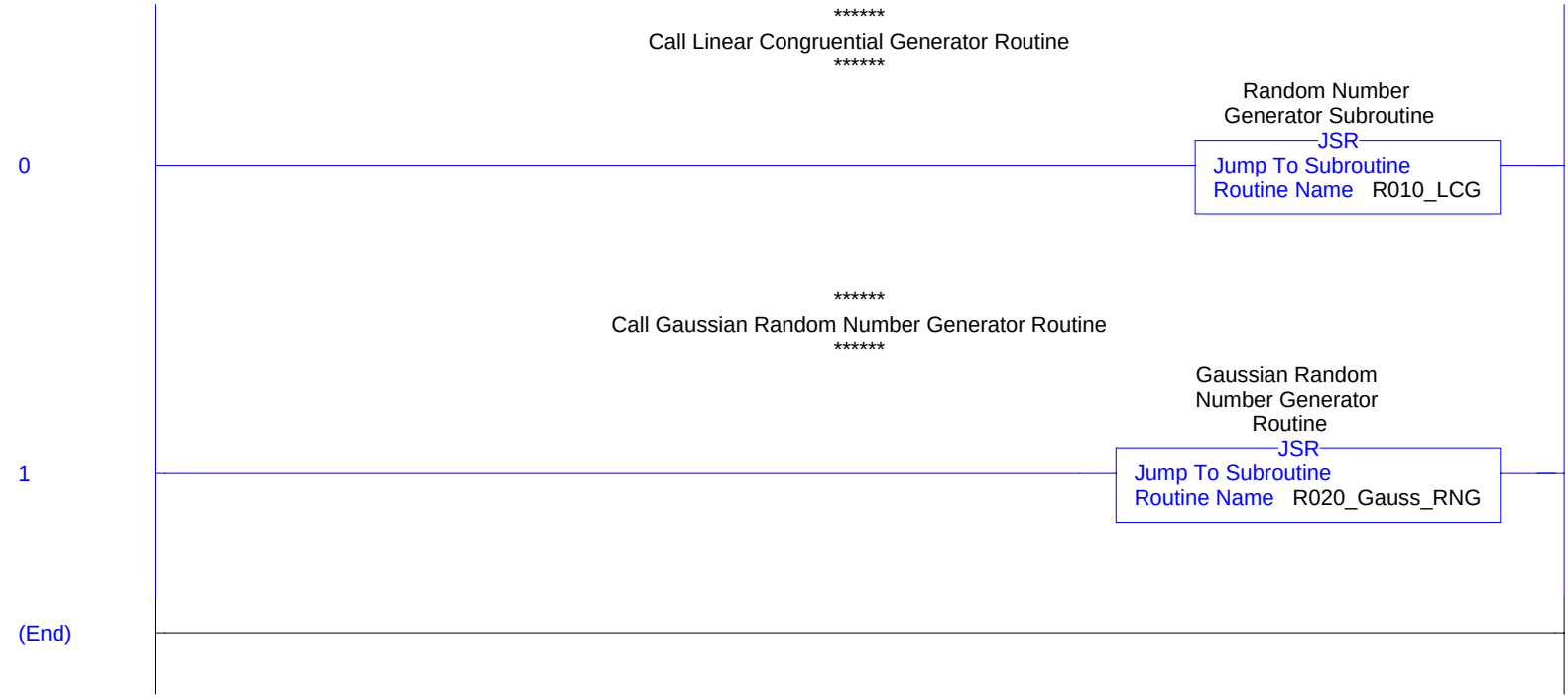
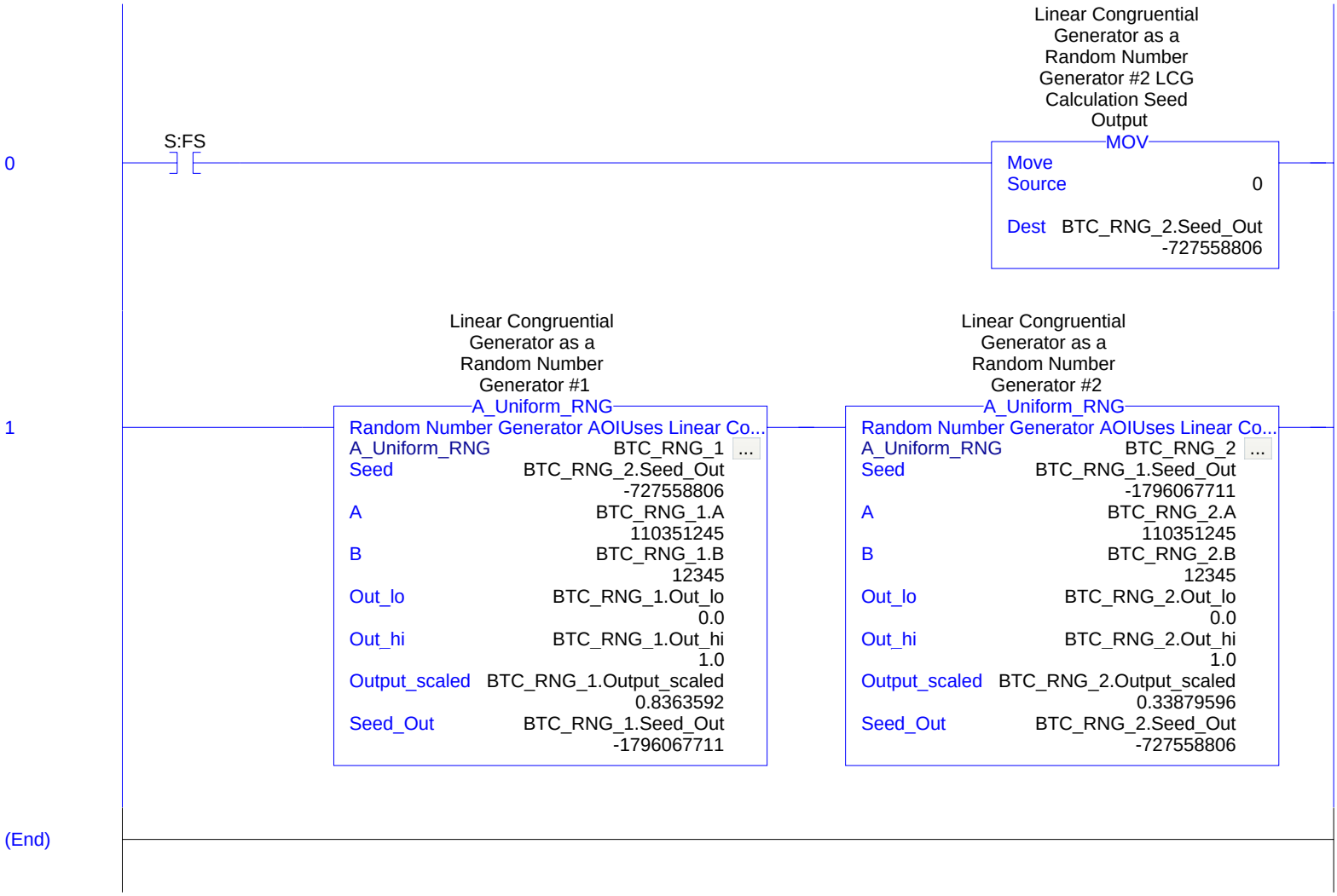
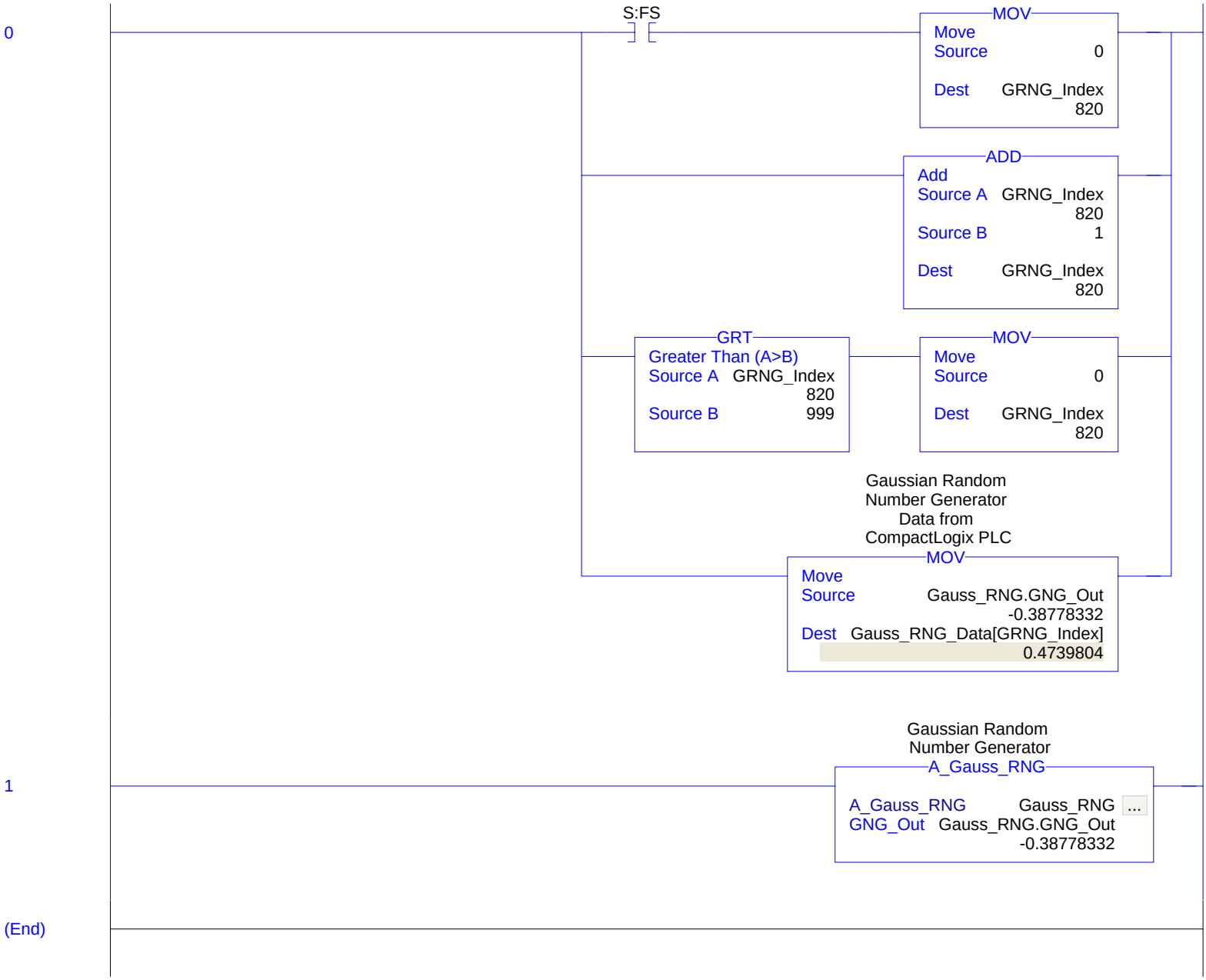

******
Call Linear Congruential Generator Routine
******
Random Number
Generator Subroutine
JSR
Jump To Subroutine
Routine Name R010_LCG
0
******
Call Gaussian Random Number Generator Routine
******
Gaussian Random
Number Generator
Routine
JSR
Jump To Subroutine
Routine Name R020_Gauss_RNG
1
(End)


0
S:FS
Linear Congruential Generator as a Random Number Generator #2 LCG Calculation Seed Output
MOV
Move
Source 0
Dest BTC_RNG_2.Seed_Out -727558806
1
Linear Congruential Generator as a Random Number Generator #1
A_Uniform_RNG
Random Number Generator AOIUses Linear Co...
A_Uniform_RNG BTC_RNG_1
Seed BTC_RNG_2.Seed_Out -727558806
A BTC_RNG_1.A 110351245
B BTC_RNG_1.B 12345
Out_lo BTC_RNG_1.Out_lo 0.0
Out_hi BTC_RNG_1.Out_hi 1.0
Output_scaled BTC_RNG_1.Output_scaled 0.8363592
Seed_Out BTC_RNG_1.Seed_Out -1796067711
Linear Congruential Generator as a Random Number Generator #2
A_Uniform_RNG
Random Number Generator AOIUses Linear Co...
A_Uniform_RNG BTC_RNG_2
Seed BTC_RNG_1.Seed_Out -1796067711
A BTC_RNG_2.A 110351245
B BTC_RNG_2.B 12345
Out_lo BTC_RNG_2.Out_lo 0.0
Out_hi BTC_RNG_2.Out_hi 1.0
Output_scaled BTC_RNG_2.Output_scaled 0.33879596
Seed_Out BTC_RNG_2.Seed_Out -727558806
(End)


0
S:FS
MOV
Move
Source 0
Dest GRNG_Index
820
ADD
Add
Source A GRNG_Index
820
Source B 1
Dest GRNG_Index
820
GRT
Greater Than (A>B)
Source A GRNG_Index
820
Source B 999
MOV
Move
Source 0
Dest GRNG_Index
820
Gaussian Random
Number Generator
Data from
CompactLogix PLC
MOV
Move
Source Gauss_RNG.GNG_Out
-0.38778332
Dest Gauss_RNG_Data[GRNG_Index]
0.4739804
Gaussian Random
Number Generator
A_Gauss_RNG
1
A_Gauss_RNG Gauss_RNG ...
GNG_Out Gauss_RNG.GNG_Out
-0.38778332
(End)

Signature Listing

A_Gauss_RNG v1.0

Available Languages

Relay Ladder

Function Block

Structured Text

```
A_Gauss_RNG(GNG_Out);
```

Parameters

Required	Name	Data Type	Usage	Description
X	A_Gauss_RNG	A_Gauss_RNG	InOut	
	EnableIn	BOOL	Input	
	EnableOut	BOOL	Output	
X	GNG_Out	REAL	Output	Gaussian-Distributed Random Number

Extended Description

Execution

Condition	Description
EnableIn is true	

Revision v1.0 Notes

Name	Default	Data Type	Scope
GNG_Out	0.0	REAL	A_Gauss_RNG
Gaussian-Distributed Random Number			
Usage:	Output Parameter		
Required:	Yes		
Visible:	Yes		
External Access:	Read Only		
*GNG_Out - A_Gauss_RNG/Logic - *5(MUL)*			

Name	Default	Data Type	Scope
A_URNG		A_Uniform_RNG	A_Gauss_RNG
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi)			
Usage:	Local Tag		
External Access:	None		
*A_URNG - A_Gauss_RNG/Logic - *1(A_Uniform_RNG)*			
A_URNG.EnableIn	1	BOOL	
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) Enable Input - System Defined Parameter			
A_URNG.EnableOut	0	BOOL	
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) Enable Output - System Defined Parameter			
A_URNG.Seed	0	DINT	
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) LCG input seed; updated after each execution			
A_URNG.A	110351245	DINT	
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) LCG multiplicative parameter A in (Seed(n+1) = Seed(n) * A + C) MOD Modulus; default is from glibc implementation			
A_URNG.A - A_Gauss_RNG/Logic - 1(A_Uniform_RNG)			
A_URNG.B	12345	DINT	
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) LCG additive parameter A in Seed(n+1) = (Seed(n) * A + C) MOD Modulus; default PJC/BTC/WA value			
A_URNG.B - A_Gauss_RNG/Logic - 1(A_Uniform_RNG)			
A_URNG.Out_lo	0.0	REAL	
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) Output value when updated seed is 0			
A_URNG.Out_lo - A_Gauss_RNG/Logic - 1(A_Uniform_RNG)			
A_URNG.Out_hi	1.0	REAL	
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) Output value for an updated seed of 2**31 (theoretical)			
A_URNG.Out_hi - A_Gauss_RNG/Logic - 1(A_Uniform_RNG)			
A_URNG.Output_scaled	0.0	REAL	
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) Actual output value at each execution			
*A_URNG.Output_scaled - A_Gauss_RNG/Logic - *1(A_Uniform_RNG), 2(LN)*			
A_URNG.Seed_Out	0	DINT	
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) LCG Calculation Seed Output			
A_URNG.Seed_Out - A_Gauss_RNG/Logic - 1(A_Uniform_RNG)			
AOI_FS	0	BOOL	A_Gauss_RNG
Gaussian Random Number Generator First Execution Done			
Usage:	Local Tag		
External Access:	None		
*AOI_FS - A_Gauss_RNG/Logic - *6(OTE), 0(XIO), 1(XIO), 2(XIO), 3(XIO), 4(XIC), 6(XIO)*			
B_URNG		A_Uniform_RNG	A_Gauss_RNG
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi)			

B_URNG (Continued)
Usage: Local Tag
External Access: None
*B_URNG - A_Gauss_RNG/Logic - *1(A_Uniform_RNG)*

B_URNG.EnableIn 1 BOOL
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) Enable Input - System Defined Parameter

B_URNG.EnableOut 0 BOOL
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) Enable Output - System Defined Parameter

B_URNG.Seed 0 DINT
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) LCG input seed; updated after each execution

B_URNG.A 110351245 DINT
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) LCG multiplicative parameter A in (Seed(n+1) = Seed(n) * A + C) MOD Modulus; default is from glibc implementation

B_URNG.A - A_Gauss_RNG/Logic - 1(A_Uniform_RNG)

B_URNG.B 12345 DINT
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) LCG additive parameter A in Seed(n+1) = (Seed(n) * A + C) MOD Modulus; default PJC/BTC/WA value

B_URNG.B - A_Gauss_RNG/Logic - 1(A_Uniform_RNG)

B_URNG.Out_lo 0.0 REAL
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) Output value when updated seed is 0

B_URNG.Out_lo - A_Gauss_RNG/Logic - 1(A_Uniform_RNG)

B_URNG.Out_hi 1.0 REAL
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) Output value for an updated seed of 2**31 (theoretical)

B_URNG.Out_hi - A_Gauss_RNG/Logic - 1(A_Uniform_RNG)

B_URNG.Output_scaled 0.0 REAL
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) Actual output value at each execution

*B_URNG.Output_scaled - A_Gauss_RNG/Logic - *1(A_Uniform_RNG), 3(MUL), 4(MUL)*

B_URNG.Seed_Out 0 DINT
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi) LCG Calculation Seed Output

*B_URNG.Seed_Out - A_Gauss_RNG/Logic - *0(MOV), 1(A_Uniform_RNG)*

Odd_Even 0 BOOL A_Gauss_RNG
Odd Even Scan Bit
Usage: Local Tag
External Access: None
*Odd_Even - A_Gauss_RNG/Logic - *6(OTE), 1(XIO), 2(XIO), 3(XIO), 4(XIC), 6(XIO)*

Seed_A 0 DINT A_Gauss_RNG
Usage: Local Tag
External Access: None
*Seed_A - A_Gauss_RNG/Logic - *1(A_Uniform_RNG)*

Seed_B 0 DINT A_Gauss_RNG
Usage: Local Tag
External Access: None

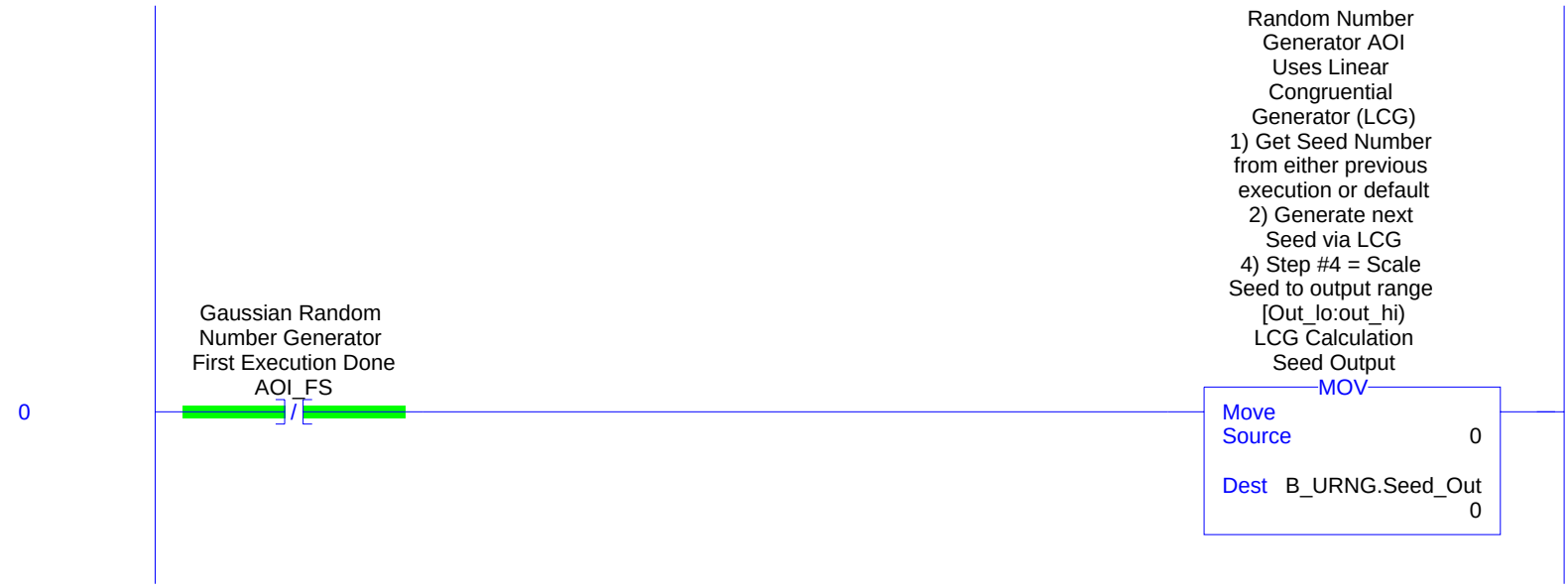
Seed_B (Continued)
*Seed_B - A_Gauss_RNG/Logic - *1(A_Uniform_RNG)*

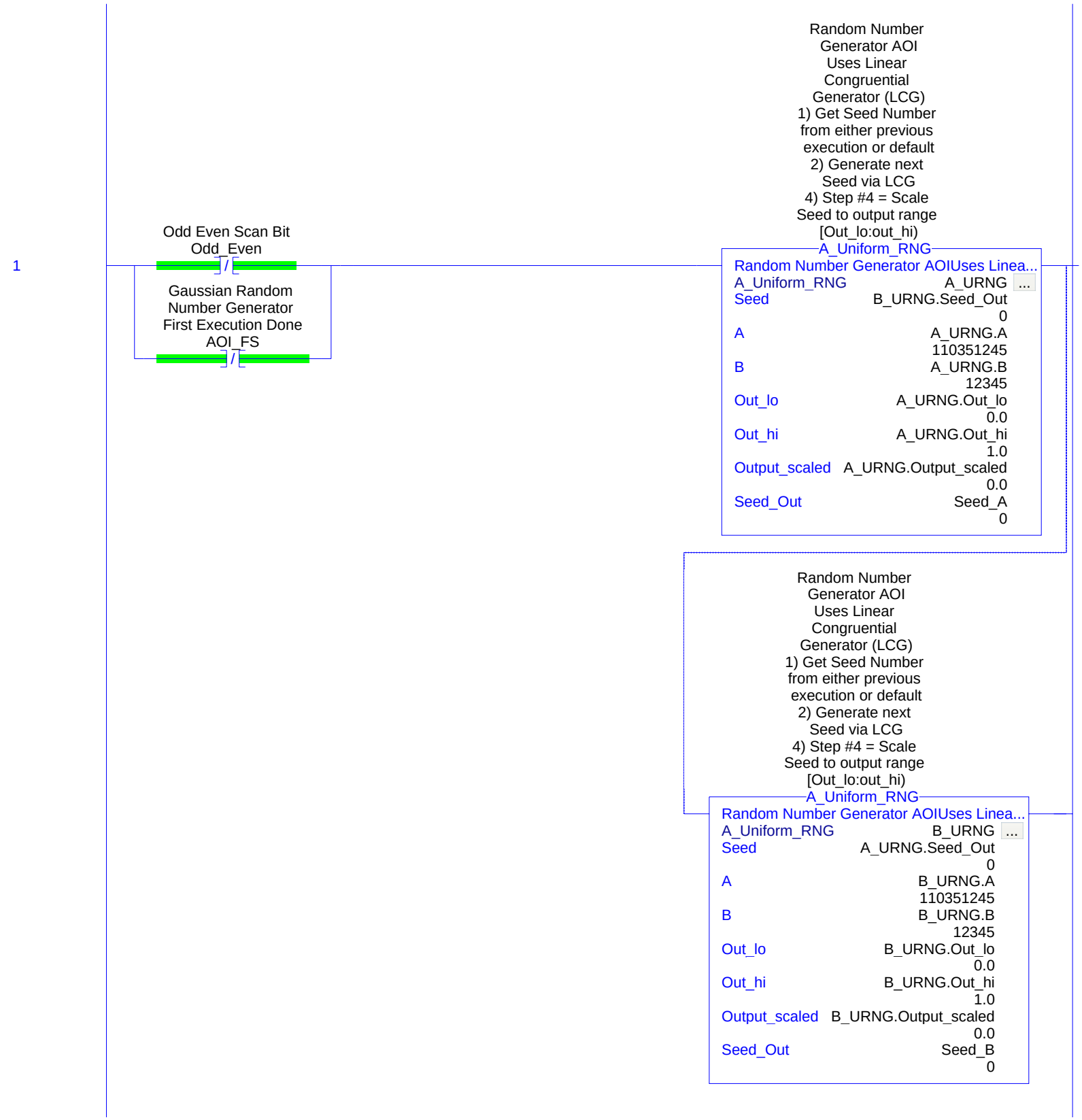
Z1	0.0	REAL	A_Gauss_RNG

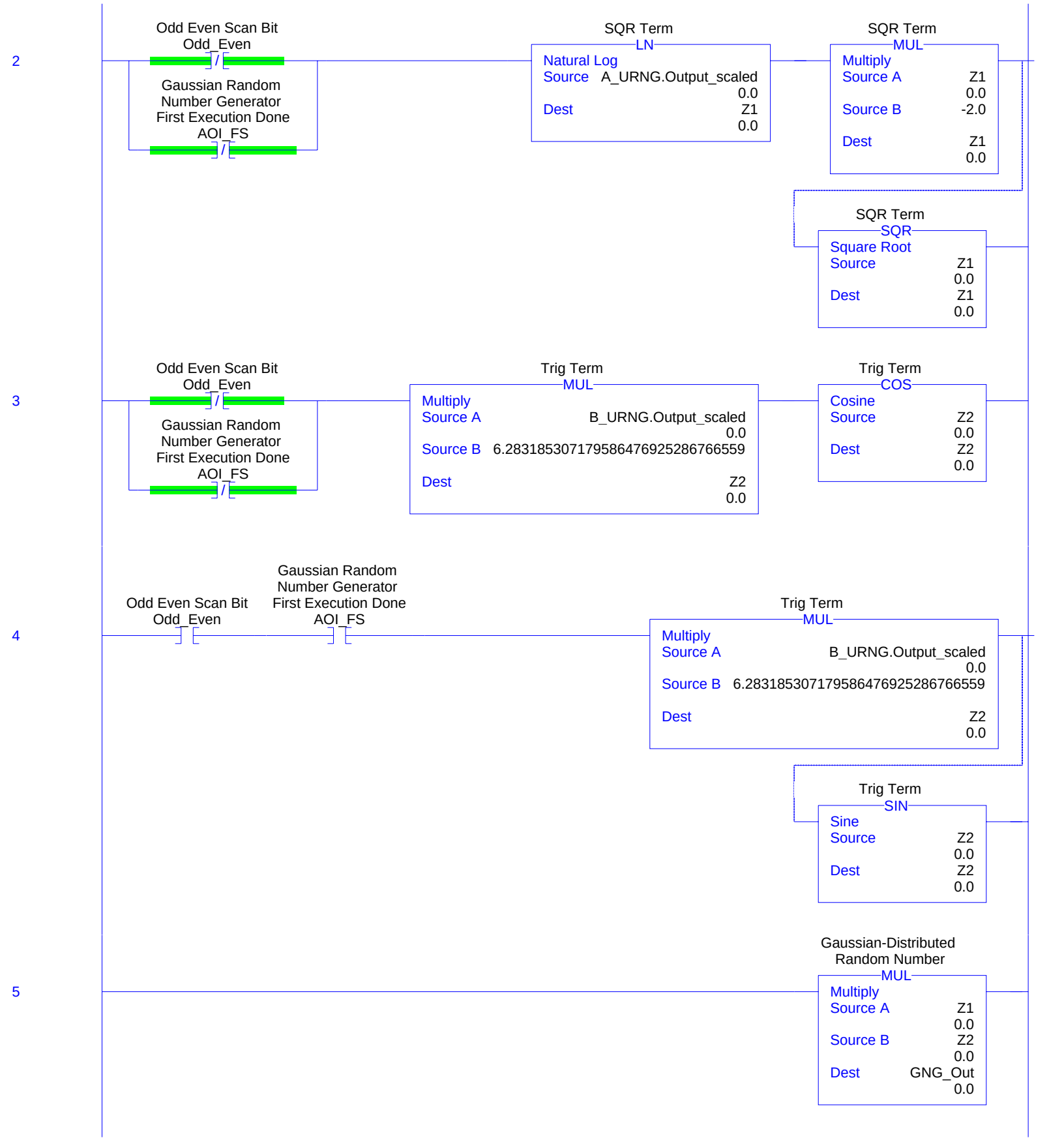
SQR Term
Usage: Local Tag
External Access: None
*Z1 - A_Gauss_RNG/Logic - *2(LN), *2(MUL), *2(SQR), 2(MUL), 2(SQR), 5(MUL)*

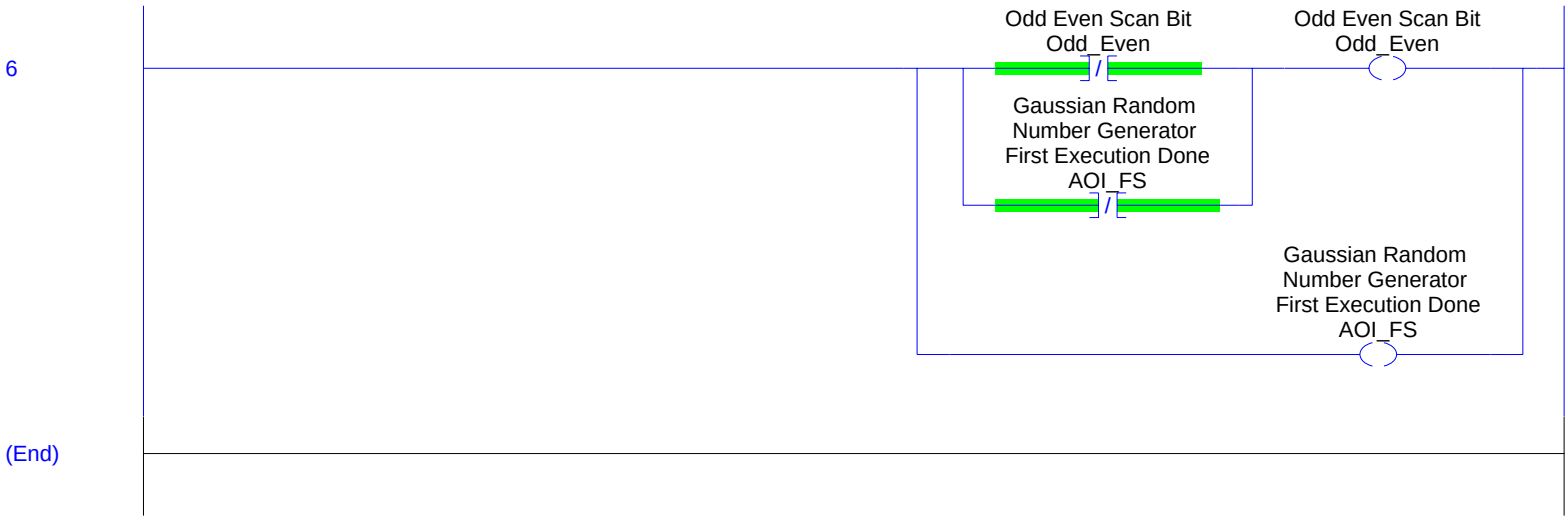
Z2	0.0	REAL	A_Gauss_RNG

Trig Term
Usage: Local Tag
External Access: None
*Z2 - A_Gauss_RNG/Logic - *3(COS), *3(MUL), *4(MUL), *4(SIN), 3(COS), 4(SIN), 5(MUL)*


Random Number Generator AOI Uses Linear Congruential Generator (LCG)
1) Get Seed Number from either previous execution or default
2) Generate next Seed via LCG
4) Step #4 = Scale Seed to output range [Out_lo:out_hi)
LCG Calculation Seed Output
Gaussian Random Number Generator First Execution Done AOI_FS
0
MOV
Move
Source 0
Dest B_URNG.Seed_Out
0


Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi)
Odd Even Scan Bit
Odd_Even
1
Gaussian Random Number Generator First Execution Done AOI_FS
A_Uniform_RNG
Random Number Generator AOIUses Linea...
A_Uniform_RNG A_URNG ...
Seed B_URNG.Seed_Out 0
A A_URNG.A 110351245
B A_URNG.B 12345
Out_lo A_URNG.Out_lo 0.0
Out_hi A_URNG.Out_hi 1.0
Output_scaled A_URNG.Output_scaled 0.0
Seed_Out Seed_A 0
Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi)
A_Uniform_RNG
Random Number Generator AOIUses Linea...
A_Uniform_RNG B_URNG ...
Seed A_URNG.Seed_Out 0
A B_URNG.A 110351245
B B_URNG.B 12345
Out_lo B_URNG.Out_lo 0.0
Out_hi B_URNG.Out_hi 1.0
Output_scaled B_URNG.Output_scaled 0.0
Seed_Out Seed_B 0


2
Odd Even Scan Bit
Odd_Even
Gaussian Random
Number Generator
First Execution Done
AOI_FS
SQR Term
LN
Natural Log
Source A_URNG.Output_scaled
0.0
Dest Z1
0.0
SQR Term
MUL
Multiply
Source A Z1
0.0
Source B -2.0
Dest Z1
0.0
SQR Term
SQR
Square Root
Source Z1
0.0
Dest Z1
0.0
3
Odd Even Scan Bit
Odd_Even
Gaussian Random
Number Generator
First Execution Done
AOI_FS
Trig Term
MUL
Multiply
Source A B_URNG.Output_scaled
0.0
Source B 6.28318530717958647692528676655 9
Dest Z2
0.0
Trig Term
COS
Cosine
Source Z2
0.0
Dest Z2
0.0
4
Odd Even Scan Bit
Odd_Even
Gaussian Random
Number Generator
First Execution Done
AOI_FS
Trig Term
MUL
Multiply
Source A B_URNG.Output_scaled
0.0
Source B 6.28318530717958647692528676655 9
Dest Z2
0.0
Trig Term
SIN
Sine
Source Z2
0.0
Dest Z2
0.0
5
Gaussian-Distributed
Random Number
MUL
Multiply
Source A Z1
0.0
Source B Z2
0.0
Dest GNG_Out
0.0

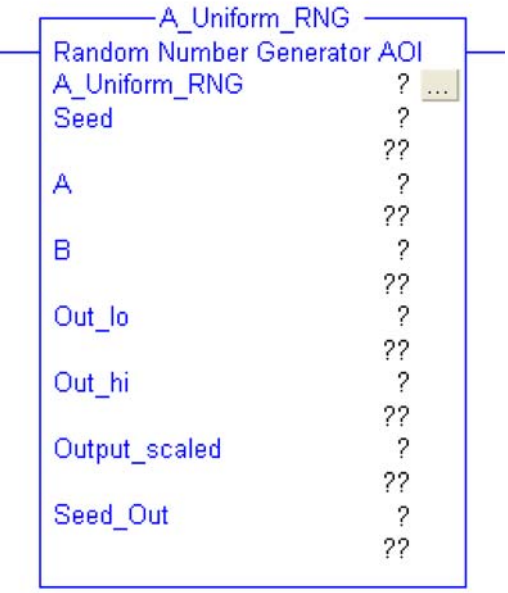

6
Odd Even Scan Bit
Odd_Even
Odd Even Scan Bit
Odd_Even
Gaussian Random
Number Generator
First Execution Done
AOI_FS
Gaussian Random
Number Generator
First Execution Done
AOI_FS
(End)

A_Uniform_RNG v1.0

Random Number Generator AOI
Uses Linear Congruential Generator (LCG)
1) Get Seed Number from either previous execution or default
2) Generate next Seed via LCG
4) Step #4 = Scale Seed to output range [Out_lo:out_hi)

Available Languages

Relay Ladder



Function Block

Structured Text

```
A_Uniform_RNG(Seed, A, B, Out_lo, Out_hi, Output_scaled, Seed_Out);
```

Parameters

Required	Name	Data Type	Usage	Description
X	A_Uniform_RNG	A_Uniform_RNG	InOut	Random Number Generator AOI Uses Linear Congruential Generator (LCG) 1) Get Seed Number from either previous execution or default 2) Generate next Seed via LCG 4) Step #4 = Scale Seed to output range [Out_lo:out_hi)
	EnableIn	BOOL	Input	
	EnableOut	BOOL	Output	
X	Seed	DINT	Input	LCG input seed; updated after each execution
X	A	DINT	Input	LCG multiplicative parameter A in (Seed(n+1) = Seed(n) * A + C) MOD Modulus; default is from glibc implementation
X	B	DINT	Input	LCG additive parameter A in Seed(n+1) = (Seed(n) * A + C) MOD Modulus; default PJC/BTC/WA value

X	Out_lo	REAL	Input	Output value when updated seed is 0
X	Out_hi	REAL	Input	Output value for an updated seed of 2**31 (theoretical)
X	Output_scaled	REAL	Output	Actual output value at each execution
X	Seed_Out	DINT	Output	LCG Calculation Seed Output

Extended Description

Execution

Condition	Description
EnableIn is true	

Revision v1.0 Notes

Name	Default	Data Type	Scope
A	110351245	DINT	A_Uniform_RNG
LCG multiplicative parameter A in (Seed(n+1) = Seed(n) * A + C) MOD Modulus; default is from glibc implementation			
Usage:	Input Parameter		
Required:	Yes		
Visible:	Yes		
External Access:	Read/Write		
A - A_Uniform_RNG/Logic - 1(MUL)			
B	12345	DINT	A_Uniform_RNG
LCG additive parameter A in Seed(n+1) = (Seed(n) * A + C) MOD Modulus; default PJC/BTC/WA value			
Usage:	Input Parameter		
Required:	Yes		
Visible:	Yes		
External Access:	Read/Write		
B - A_Uniform_RNG/Logic - 1(ADD)			
Output_scaled	0.0	REAL	A_Uniform_RNG
Actual output value at each execution			
Usage:	Output Parameter		
Required:	Yes		
Visible:	Yes		
External Access:	Read Only		
*Output_scaled - A_Uniform_RNG/Logic - *3(CPT)*			
Out_hi	1.0	REAL	A_Uniform_RNG
Output value for an updated seed of 2**31 (theoretical)			
Usage:	Input Parameter		
Required:	Yes		
Visible:	Yes		
External Access:	Read/Write		
Out_hi - A_Uniform_RNG/Logic - 3(CPT)			
Out_lo	0.0	REAL	A_Uniform_RNG
Output value when updated seed is 0			
Usage:	Input Parameter		
Required:	Yes		
Visible:	Yes		
External Access:	Read/Write		
Out_lo - A_Uniform_RNG/Logic - 3(CPT)			
Seed	0	DINT	A_Uniform_RNG
LCG input seed; updated after each execution			
Usage:	Input Parameter		
Required:	Yes		
Visible:	Yes		
External Access:	Read/Write		
Seed - A_Uniform_RNG/Logic - 1(MUL)			
Seed_Out	0	DINT	A_Uniform_RNG
LCG Calculation Seed Output			
Usage:	Output Parameter		
Required:	Yes		
Visible:	Yes		
External Access:	Read Only		
*Seed_Out - A_Uniform_RNG/Logic - *1(OR), 3(DIV)*			

Name	Default	Data Type	Scope
float_internal	0.0	REAL	A_Uniform_RNG
Intermediate calculations for output scaling			
Usage:	Local Tag		
External Access:	None		
*float_internal - A_Uniform_RNG/Logic - *3(DIV), 3(CPT)*			
Se	0	DINT	A_Uniform_RNG
Internal copy, and calculation, of seed			
Usage:	Local Tag		
External Access:	None		
*Se - A_Uniform_RNG/Logic - *1(ADD), *1(MUL), 1(ADD), 1(OR)*			

Random Number Generator using Linear Congruential Generator

0

NOP

This rung updates Seed: Seed(n+1) = (Seed(n) * A + B) MOD (2**31)
N.B. MOD is implemented via DINT overflow plus bit-wise AND with (2**31 - 1) i.e. 0x7FFFFFFF.,
so local tag [Seed_internal] may not be the same as the final seed after the AND-cum-MOD

1

Internal copy, and calculation, of seed

MUL
Multiply
Source A Seed 0
Source B A 110351245
Dest Se 0

Internal copy, and calculation, of seed

ADD
Add
Source A Se 0
Source B B 12345
Dest Se 0

LCG Calculation Seed Output

OR
Bitwise Inclusive OR
Source A Se 0
Source B -2147483648
Dest Seed_Out 0

2

S:V
U

This rung scales Seed to Output_scaled: Output_scaled(n+1) = Func(Seed(n+1),Out_lo,Out_hi)

3

Intermediate calculations for output scaling

DIV
Divide
Source A Seed_Out 0
Source B -2147483648.0
Dest float_internal 0.0

Actual output value at each execution

CPT
Compute
Dest Output_scaled 0.0
Expression ((1.0 - float_internal) * Out_lo) + (float_internal * Out_hi)

(End)

Data type Name: A_Gauss_RNG

Description:

Size: 104 byte(s)

Name	Data Type	Style	Description
EnableIn	BOOL	Decimal	Enable Input - System Defined Parameter
EnableOut	BOOL	Decimal	Enable Output - System Defined Parameter
GNG_Out	REAL	Float	Gaussian-Distributed Random Number

External Access
Read Only
Read Only
Read Only

Data type Name: A_Uniform_RNG

Description:
Random Number Generator AOI
Uses Linear Congruential Generator (LCG)
1) Get Seed Number from either previous execution or default
2) Generate next Seed via LCG
4) Step #4 = Scale Seed to output range [Out_lo:out_hi)

Size: 40 byte(s)

Name	Data Type	Style	Description	External Access
EnableIn	BOOL	Decimal	Enable Input - System Defined Parameter	Read Only
EnableOut	BOOL	Decimal	Enable Output - System Defined Parameter	Read Only
Seed	DINT	Decimal	LCG input seed; updated after each execution	Read/Write
A	DINT	Decimal	LCG multiplicative parameter A in (Seed(n+1) = Seed(n) * A + C) MOD Modulus; default is from glibc implementation	Read/Write
B	DINT	Decimal	LCG additive parameter A in Seed(n+1) = (Seed(n) * A + C) MOD Modulus; default PJC/BTC/WA value	Read/Write
Out_lo	REAL	Float	Output value when updated seed is 0	Read/Write
Out_hi	REAL	Float	Output value for an updated seed of 2**31 (theoretical)	Read/Write
Output_scaled	REAL	Float	Actual output value at each execution	Read Only
Seed_Out	DINT	Decimal	LCG Calculation Seed Output	Read Only