AN-X Technical Note

Measuring Response Time in the AN-X-GENI

This technical note describes how we measured response time and processing time when using the AN-X-GENI with a ControlLogix to control Genius I/O.

Response time is the time from when an output is turned on in the ControlLogix controller to when it appears at the output terminal, or the time from when an input terminal changes until the new value appears in the ControlLogix controller.

Processing time is the time consumed by devices such as the AN-X and the ENBT in transferring data.

Introduction

When a ControlLogix processor uses an AN-X-GENI to scan a Genius I/O network, response time depends on the following factors:

- ControlLogix scan time
- connection RPI
- propagation time over Ethernet (cables, switches, etc.)
- processing time in ENBT and AN-X, depends on the number of connections and traffic
- Genius network scan time
- processing time in the Genius output and input modules

Other factors to consider

- The ControlLogix logic scan and the I/O scan are asynchronous. When the logic turns on an output, there can be a delay of up to one RPI before the ENBT sends the output data on Ethernet.
- Ethernet inputs and outputs are asynchronous. The AN-X sends its data when its RPI timer expires. The ENBT does the same. But the two are not synchronized.

Quest Technical Solutions

4110 Mourning Dove Court
Melbourne FL 32934
321 757-8483

www.qtsusa.com

- Similarly, the update from the ControlLogix to the AN-X and the AN-X scan on the Genius network are asynchronous. There can be a delay of up to one scan time before the AN-X sends the output data on the I/O network.
- When the ControlLogix processor sets an output for an I/O module and the output is wired to an input on the same module, there will be a minimum of one I/O scan before the input is sent to the ControlLogix. In addition, inputs are not sent at the same time that the module receives outputs.
- The Genius network scan time depends on the number of nodes and the other traffic on the network.

Method

To measure the response time, we turned on an output that was wired to an input in the same I/O block and measured the time from when the output was turned on in the ControlLogix to when the input came on in the ControlLogix. We recorded the minimum, maximum and average times for the round trip of the data.

We performed the test on a Genius network with 7 nodes.

To measure the Genius network scan times, we used an AN-X-GENI-CAPT module. We verified the times with a Genius hand-held monitor.

All tests were done with one connection from the ControlLogix to the AN-X, with an RPI of 5 ms and 248 words of outputs and 250 words of inputs.

Results

The table summarizes the results of the measurements. Minimum, maximum and average times are for the round trip of the data, from turning on the output in the ControlLogix to receiving the input in the ControlLogix. All times are in milliseconds.

Quantity	Time, ms
Minimum time	23
Maximum time	45
I/O scan time	13.18 ± 0.02
Input delay	7.94 ± 0.004

The average times were calculated over a minimum of 100 updates.

The I/O scan times were calculated from 100 scans.

Analysis

Processing Time

We can obtain an estimate of the AN-X processing delay from either the maximum or minimum time.

Minimum Time

The minimum time represents the case where the ControlLogix just catches the outgoing RPI from the ENBT to the ANX, and the AN-X just catches the I/O scan. The input is returned on the next I/O scan, after a delay of 7.96 ms, and just catches the Ethernet update from the AN-X to the ENBT. The total delay is therefore one I/O scan time plus one input delay plus 2 processing delay times, one for the output path and one for the input path.

From the table we can estimate the processing delay (ENBT + AN-X + everything else) by subtracting the I/O scan time and input delay from the minimum time and dividing by 2. In each case the result, to the precision possible from the times, is 1 ms.

Maximum Time

The maximum time represents the case where the output just misses the outgoing RPI to the AN-X and then just misses going out on the Genius scan. It waits a full I/O scan before it is sent out on the Genius network. The input comes back on the next scan, and then just misses the RPI from the AN-X to the ENBT. The total delay is therefore two I/O scans plus one input delay plus two plus 2 processing delay times plus 2 RPIs, one for the output path and one for the input path.

From the table we can estimate the processing delay (ENBT + AN-X + everything else) by subtracting (2 * I/O scan time + 2 * RPI + input delay) from the maximum time and dividing by 2. In each case the result, to the precision possible from the times, is less than 1 ms

To test the validity of the analysis of the worst case, we changed the RPI from 5 to 10 ms. We would expect that the minimum time would be unaffected and the maximum time would increase by twice the change in the RPI. The times measured were 23 ms (minimum) and 45 ms (maximum), as predicted.

Conclusions

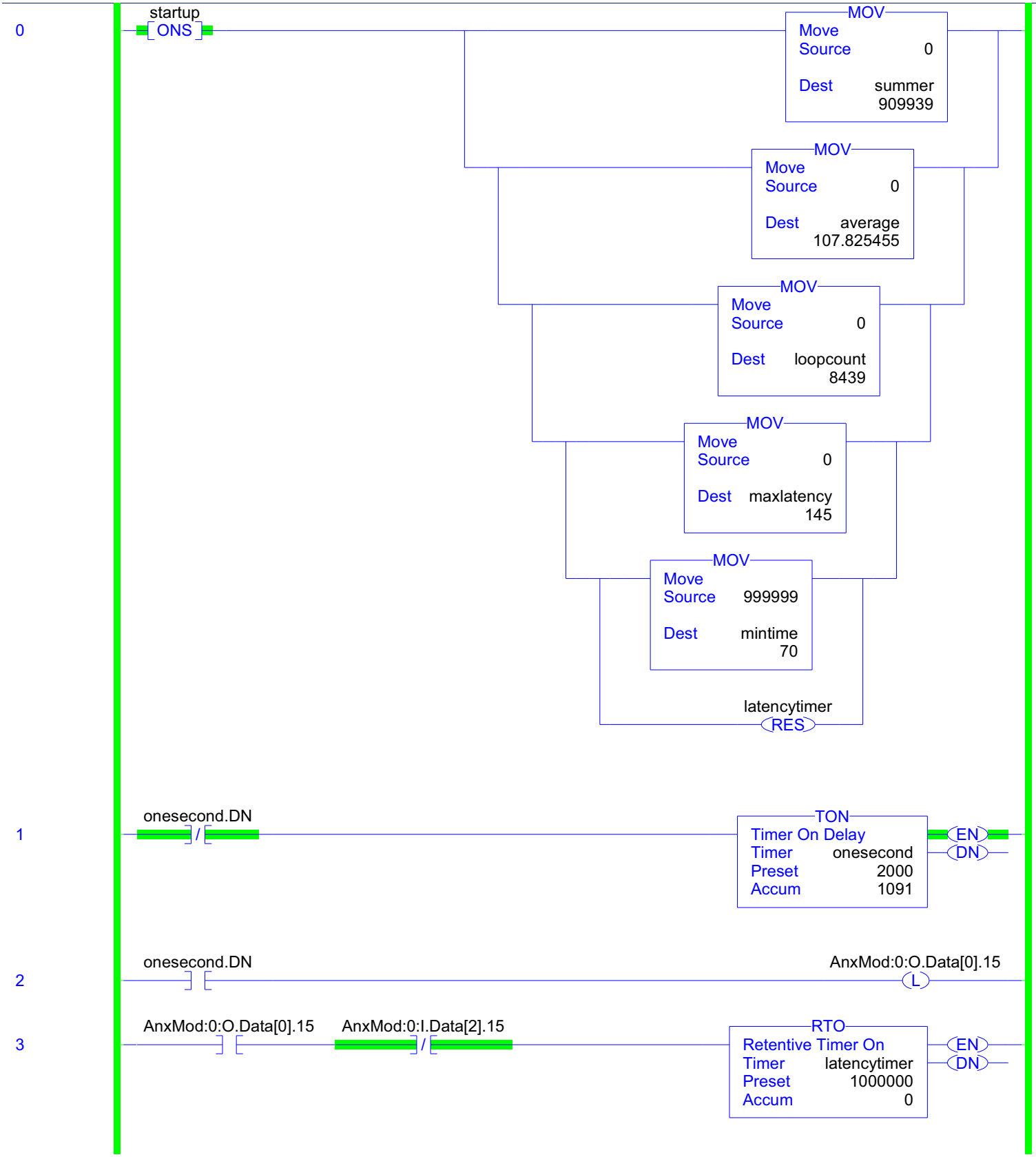
The estimated total data processing time in the AN-X, ENBT and so on is 1 ms or less.

All other variations in the delivery time of data are the result of the asynchronous nature of communication between the ENBT and the AN-X and the communication on the Genius network.

Quest Technical Solutions

4110 Mourning Dove Court
Melbourne FL 32934
321 757-8483

www.qtsusa.com


0
startup
ONS
MOV
Move
Source 0
Dest summer
909939
MOV
Move
Source 0
Dest average
107.825455
MOV
Move
Source 0
Dest loopcount
8439
MOV
Move
Source 0
Dest maxlatency
145
MOV
Move
Source 999999
Dest mintime
70
latencytimer
RES
1
onesecond.DN
TON
Timer On Delay
Timer onesecond
Preset 2000
Accum 1091
EN
DN
2
onesecond.DN
AnxMod:0:O.Data[0].15
L
3
AnxMod:0:O.Data[0].15
AnxMod:0:I.Data[2].15
RTO
Retentive Timer On
Timer latencytimer
Preset 1000000
Accum 0
EN
DN

4

latencytimer.TT

GRT
Greater Than (A>B)
Source A latencytimer.ACC 0
Source B maxlatency 145

MOV
Move
Source latencytimer.ACC 0
Dest maxlatency 145

5

AnxMod:0:O.Data[0].15 AnxMod:0:I.Data[2].15 one ONS

MOV
Move
Source latencytimer.ACC 0
Dest last 83

ADD
Add
Source A loopcount 8439
Source B 1
Dest loopcount 8439

ADD
Add
Source A summer 909939
Source B last 83
Dest summer 909939

LES
Less Than (A<B)
Source A last 83
Source B mintime 70

MOV
Move
Source last 83
Dest mintime 70

6

AnxMod:0:O.Data[0].15 AnxMod:0:I.Data[2].15

AnxMod:0:O.Data[0].15 U

latencytimer RES

7

DIV
Divide
Source A summer 909939
Source B loopcount 8439
Dest average 107.825455