

To: MTB Distribution
From: J.R.Ohlin & J.P.Brown
Date: January 2, 1979
Subject: Multics MR6.5A--DN6678 System Test Results

INTRODUCTION

This MTB documents the results of the MR6.5A DN6678 FNP test effort. The overall objectives this test effort were to verify that the performance and functionality of the DN6678 were equivalent to that of the DN6632.

TEST ENVIRONMENT

The initial DN6678 test environment was planned and started on fiscal week 39 on the Multics benchmark service machine (System MB) using the H6680 GCOS system (System Y4) as a Cuesta driver. The environment was then changed to perform testing on the Multics service system (System M) using some deconfigured hardware components (known as System M1) for the Cuesta driver. The testing was concluded on fiscal week 46.

TEST STRATEGY

The testing was performed in five separate phases; some proceeding in parallel with others.

Phase 1:

DN6678 functionality was validated to determine if the systems capability/capacity was identical to that of the DN6632.

- a. A Cuesta script was generated which tests the DN6632 and it was then run on the DN6632 to establish the test vehicle. The Cuesta script was tuned (as close as possible) to a representative "typical" Multics user load.
- b. The Cuesta script was then run on the DN6678; no differences or deficiencies were observed.

Phase 2:

DN6678 software and hardware configuration guidelines and recommendations were determined. Of particular concern in this phase were buffer depletion problems, which were expected, due to

Multics Project working documentation. Not to be reproduced or distributed outside the Multics Project.

the increased buffer requirements for HSLA sub-channels; as compared to LSLA sub-channels.

- a. The Cuesta script was run with a variety of MCS core images which contained combinations of the software modules: trace, autobaud_tables, acu_tables, bsc_tables, g115_tables. These modules might or might not be required at the customer sites.
- b. The number of sub-channels were adjusted (increased and decreased) depending on the amount of available buffer space.

Phase 3:

Various terminal connections were tried to determine if their required channel adapters could be connected and/or operated correctly on the DN6678. Where applicable, terminals were tested in halfduplex, fullduplex, and echoplex modes using a terminal test program especially adapted for this test effort. The following terminals were tested:

- o Several ASCII terminals (up to 9600 baud)
 - dial-up line [DCF6612]
 - direct connect [DCF6624]
 - o TTY33, and TTY38 [DCF6612]
 - o IBM 2741 [DCF6612]
 - o 20 & 60ma current mode interface [DCF6610]
 - o IBM 2780/3780 bisync device [DCF6618]
 - o ARDS 1200 [DCF6612]
 - o Autobaud detection
 - o g115 Remote computer - Sync protocol
 - Direct connect [DCF6625]
 - Dial-up line [DCF6611]
 - o Triformation Braille - LED120 [DCF6612]
- a. Terminal functionality tests not yet completed, but presently in progress are:
- o Autocall (ACU) [DCF6613]

- b. As a result of the terminal tests, only two terminal problems were found; one, the inability to input lower case characters on a 2741 type device, and two the FNP crashed when trying to log a 300 baud device into a 134 baud channel. Both problems necessitated changes to hsla man; once the changes were made, these features were re-tested and verified to work correctly.

Phase 4:

The performance level of MCS running on the DN6678 was established. See Appendix A (Performance Results).

Phase 5:

Various system operations and answering service commands/tools were tested with the DN6678 to verify correct operation(s).

- a. The following MCS tools were tested; no discrepancies were found.
- o load_fnp
 - o dump_fnp
 - o db_fnp
 - o trace
- b. MCS operation was tested only with the console attached; no discrepancies were found. NOTE: A console-less test is planned for a later date.
- c. The following answering service commands were tested. no discrepancies found.
- o remove
 - o detach
 - o attach

MEMORY CONFIGURATOR

The following is a memory configurator which is recommended to be used for configuring DN6678s on MR6.5A.

DN6678 MEMORY CONFIGURATOR

T A B L E 1

MODULE	LENGTH	MUST HAVE	TOTAL
acu_tables	104		
ards_tables	674		
autobaud_tables	292		
breakpoint_man	218		
bsc_tables	2296		
console_man	476	*	476
g115_tables	984		
ibm3270_tables	616		
t202_tables	546		
trace	188		
trace buffer	xxx		
vip_tables	534		
free buff pad	960	*	960
control_tables	1664	*	1664
dia_man	2816	*	2816
hsla_man	3410	*	3410
interpreter	1848	*	1848
scheduler	1172	*	1172
utilities	1630	*	1630
interrupt vect	512	*	512
hsla hwcm	1536	*	1536
init	3278	*	----
SubTotal1			

T A B L E 2

No. of Chans X	Half/ Full duplex	Echo- plex	Half/ Full duplex	Echo- plex	TOTAL
	----- v	or ----- v	= ----- v	or ----- v	
32 tty <9600	152(1)	184(2)	4864	5888	
40 tty <9600	152	184	6080	7360	
48 tty <8832	"	"	7296	8832	
56 tty <9600	"	"	8512	10304	
64 tty <9600	"	"	9728	11776	
72 tty <9600	"	"	10944	13248	
80 tty <9600	"	"	12160	14720	
88 tty <9600	"	"	13376	16192	
96 tty <9600	"	"	14592	17664	
SubTotal2					

T A B L E 3

No. of Chans X (IO buf + Misc.) = $\frac{\text{-----}}{\text{-----}}$ = TOTAL
 |
 v

g115 protocol	192	108(3)		300		
2780 protocol	224	118(4)		342		
3780 protocol	256	118		374		
				Subtotal3		

Legend: (1) =	SCFM	52	Legend: (2) =	SCFM	52
	TIB	34		TIB	34
	TB tbl	2		TB tbl	2
	IO buf	64		IO buf	96
		<u>152</u>			<u>184</u>

Legend: (3) =	SCFM	52	Legend: (4) =	SCFM	52
	TIB	34		TIB	34
	TB tbl	2		TB tbl	2
	TB ext	20		TB ext	30
		<u>108</u>			<u>118</u>

HOW TO USE THE CONFIGURATOR

To compute the available free buffer space (as a result of configuring required software modules) use the following algorithm.

1. Sum the length of all required modules in the TOTAL column (Table 1) to obtain SubTotal1.
2. Enter the appropriate length of Half/full duplex, or echoplex tty channels in the TOTAL column (Table 2) to obtain SubTotal2.
3. Enter the appropriate length of the g115 and 2780/3780 channels in the TOTAL column (Table 3) to obtain SubTotal3.
4. Add SubTotal1, SubTotal2 SubTotal3. Subtract that sum from the value 32,768; the difference is the amount of available free buffer space.

NOTE: With many sync and bisync channels configured (Table 3) there may be some contention for free buffer space for output

buffers, if so, the length of "free buff pad" (Table 1) should be increased by increments of 32 words until contention ceases. NOTE: If the remainder is negative, you have too many channels configured.

APPENDIX A.

(Performance Results)

OBJECTIVES

The objective of DN6678 performance testing was to determine the throughput limits of the device, and the user environment at which these limits occurred.

ENVIRONMENT

Cuesta was chosen as the driver system because of the ability to place a predictable and controllable load on the host system (DNMR6MR678/Multics). System M was used as the system under test using MR6.5A Multics. The available communications network consisted of:

- o 60 1200 baud echoplex tty lines
- o 2 4800 baud bisync lines
- o 2 9600 baud bisync lines

The bisync lines were exercised using the "test_bisync" module obtained from CISL. This module is a user process that employs physically wrapped lines to send and receive messages. It was used with the multi-record and non-transparent modes selected, and physical record size was fixed at 300 characters. Several Cuesta scripts were developed, however only one was used.

This script was intended to represent a typical interactive user session. It was designed so as not to drive the central system into saturation. At no time did we want central system response time to be a limiting factor. Comparing the script to the loads normally seen on System M service indicated that the loads on the FNP were similar. The difference being that System M service, during peak loads, runs at or near saturation while the Cuesta scripts kept system utilization near or below 50%.

TEST FNP SOFTWARE CORE IMAGE

```
scheduler
interpreter
control_tables
dia_man
hsla_man
console_man
bsc_tables
utilities
trace
init
```

PERFORMANCE CHART

	Number of Users			
	40	60	60 and 2 bisyncs	60 and 4 bisyncs
Average DIA throughput Characters/second	180	300	2100	2650
Interactions/second	1.4	2.3	5.8	6.8
Response Time	<1 sec	<1 sec	<1 sec	<1 sec
Average Free Buffers	121	46	35	26
FNP Busy	8%	11%	39%	50%

PERFORMANCE DISCUSSION

The DN6678 proved to be a highly reliable device. After approximately 200 hours of testing, no instance of DN6678 or MCS failures were encountered. However, occasional FNP crashes did occur when the number of available free buffers fell below 25. In general, though, this is recognized as the expected threshold for failure, regardless of specific hardware configurations.

It is interesting that a significant performance degradation occurs at approximately 30-35 average free buffers. This degradation manifested itself as spurious "bell quits" (10-100/per hour) and often resulted in halving of total throughput.

While there is no definition of a tolerable level of "bell quits" it was observed that 2-10/per hour had little effect on throughput. This level occurred at approximately 30-35 average free buffers. Because of the larger buffer requirements, buffer depletion problems more severely affected the bisync lines. During periods of high "bell quit" activity the bisync lines would effectively be unusable.

DIA throughput was measured between 300 and 3000 chars/sec depending on the lines configured. This throughput rate seems to be well within the requirements of current FNP usage. As expected, the FNP (because of its finite queue) response time

proved to be quite adequate. Given that it is operating free of the "bell quit" syndrome, it consistently produced response times less than 1 second.

Interactions per second gives us insight into the DIA related activity. The achievement of 7 interactions per second with excellent response time is an adequate test. This appears to be within current FNP requirements.

At no time did it appear that we were taxing the DN6678 processor. Even our max throughput was achieved using 50% of available power. These figures were derived from the fnp_idle module. Unfortunately during the DN6632 tests, this package was not available for comparison.

Finally, from a system performance standpoint, testing has shown that the DN6678 is a reasonable FNP replacement for the DN6632.