

To: Distribution  
From: Richard Fakoury

MULTICS COLTS MTB

03/19/81

---

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

- 1.0 PURPOSE
- 2.0 MULTICS COLTS
- 3.0 ASSUMPTIONS
- 4.0 COLTS implementation on Multics
  - 4.1 COLTS executive in Multics (CCOLT)
  - 4.2 COLTS executive in the FNP (FNP Colt)
  - 4.3 Subchannel Testing
- 5.0 SUMMARY
- 6.0 Basic Theory of Operation
  - 6.1 Summary of Executive ICM Commands
  - 6.2 Test Request
  - 6.3 Initialization and Validation
  - 6.4 Test Page Preprocessing
  - 6.5 Multics Implementation
- 7.0 Summary of FNP COLTS Operation
  - 7.1 List of Basic Functions
  - 7.2 Detailed Sequence of FNP Operations

1.0 PURPOSE

This MTB describes how COLTS works and how it will be implemented on Multics.

2.0 MULTICS COLTS

Very briefly, COLTS under Multics is the Communication On-Line Test System which is responsible for testing the communications peripherals (Datanet 355 and Datanet 66xx) in L68 system. It is a subexecutive of TOLTS, the T&D (test and diagnostic) executive system. TOLTS provides the common executive type services for all the T&D subexecutives, COLTS, POLTS (peripherals), MOLTS (mainframes) and ISOLTS (cpu's).

COLTS allows online testing of all subchannels (that portion of a FNP hardware which is dedicated to a terminal interface). Normal communications activity on other channels can continue while a subchannel test is in progress.

There are two Colts, one in the host, CCOLT, which is in control of the communications system testing, the other, FNP Colts, is resident in the Datanet.

3.0 ASSUMPTIONS

3.1 COLTS user interface will be preserved.

3.2 COLTS test strategy will remain the same:

- \* All tests are user initiated.
- \* Only subchannel tests will be online.

4.0 COLTS IMPLEMENTATION ON MULTICS

4.1 COLTS executive in the host

CCOLT is responsible for:

- \* Resource allocation
- \* Test page loading
- \* Test page dispatching

- \* Test sequencing
- \* I/O set up and issue
- \* Error check
- \* Error message formatting
- \* Test page termination
- \* Memory reallocation
- \* Resource reallocation

The COLTS executive presently running on GCOS is 10K word in size. It is the intent to move the executive over to Multics intact. The GCOS interfaces and the Master Mode calls will be interpreted by Tolts into Multics functionality.

#### 4.2 COLTS executive in the FNP

The FNP Colts executive will communicate with COLTS executive. In addition, it will be responsible for:

- \* Memory allocation
- \* Memory reallocation
- \* Subchannel testing

The FNP COLTS executive will initially be core resident and will occupy approximately 1024 words. FNP Colts and the submodule, which will be approximately 512 words, will interface directly with the Multics Communication System (MCS) and the subchannel that is under test.

All test requests are initiated from a user terminal. The test requests are relayed to TOLTS which then calls CCOLT. The following sections describe the roles of both CCOLT and FNP Colt in the communications system testing in Multics.

- #### 4.3
- Testing is initiated by a Tolts request: test Cnccss which will request that Colts be brought into execution under Tolts. This request specifies which FNP - n, FNP channel - cc and subchannel - ss is required for testing. Tolts will validate the test request by attempting an

dial\_manager\_\$standd\_attach. If the subchannel is available the request is granted, else an error is detected and the test is terminated.

Once the subchannel is dedicated to COLTS, the following are need:

- \* L68 Test Page
- \* FNP Test Page

The L68 test page will be retrieved from the file system. The Test Page is the L68 resident test driver that applies tests to the subchannel, interprets the results and outputs error information to the user terminal.

The FNP Test Page will be sent as data to the FNP Colts, by using iox\_\$put\_chars. The FNP Test Page will support the L68 Test Page.

(In the FNP COLTS, the FNP Test Pages are actually special channel handlers which better accommodate subchannel testing.)

The subchannel is now tested by the L68 Test Page in conjunction with the FNP Test Page. The subchannels have a data wrap capability at the point just before the EIA serial interface. The logic up to the EIA interface can be tested by COLTS.

Note that this scheme for online subchannel testing assumes that the FNP Central (CPU and memory), and the DIA are all functioning properly.

## 5.0

### SUMMARY

The basic strategies for implementing COLTS on Multics are:

- \* Maintain the present GCOS COLTS test philosophies and user interfaces.
- \* Preserve as much as possible of the GCOS COLTS executive
- \* Use existing T&D Test Pages.

## 6.0 BASIC THEORY OF COLTS OPERATION

COLTS operation could be summarized by saying that testing is accomplished through an exchange of ICM's across a DAC connection using the GEROUT interface. Since this is probably meaningless to most, lets try to elaborate on each item.

When a COLTS test for an adapter subchannel is running, there are four main modules involved.

The COLTS L68 executive handles test call, allocation, and most overhead functions. It needs to communicate with the COLTS FNP executive, which performs similar functions in the FNP.

The Test Page defines all the channel programs to be executed for testing, and does all error checking. COLTS needs to send these channel programs to the FNP and receive an indication of the response they caused in the FNP - i.e., data, status, residues, etc.

The basic unit that accomplishes these communications functions is the Inter-Computer-Message (ICM). The ICM is nothing more than a block of data whose contents are meaningful only to the T&D system. In general, an ICM will consist of a checksum, a COLTS ICM opcode, and associated data for executive-to-executive communications, or a checksum and channel program or response data for test-to-test communications.

### 6.1 SUMMARY OF EXECUTIVE ICM COMMANDS

#### L68 to FNP

- Start test (unique opcode for each test type)
- link device submodule 1
- link device submodule 2

#### FNP to L68 (responses to above)

- Operation complete
- Invalid ICM
- No test for this resource
- Line Disconnected
- Module not linked
- Buffer denial

Assuming that the ICM's can accomplish the testing function, the task becomes one of providing a means for their exchange. In Multics, FNP Colts will resemble a 9600 baud channel with raw in (rawi) and raw out (rawo) mode attributes, and will be started by Tolts with a call to `dail_manager_$standd_attach` by Tolts, which has the same effect as if the channel had signaled a dialup. If the attachment is without error, FNP Colts will then be ready to process CCOLT requests. Since the ICM is nothing more than blocks of data, it will be transmitted and received by Tolts using `iox_$put_chars` and `iox_$get_chars`. The ICM in the FNP will be accomplished by utilizing the Colts TIB and op-blocks.

On the L68 side, the application program (CCOLTS) now needs a way to process this request, establish the line connection, and send receive ICM's. All of these functions are accomplished by various facets of the GCOS GEROUT interface with Tolts.

The program link with this interface is the MME GEROUT instruction in the applications program. Control information accompanying the MME defines the exact functionality required. A summary of the MME GEROUTS follows with brief description of the COLTS usage and its Multics function.

<u>GEROUT Type</u>	<u>Usage</u>	<u>Multics Cmd.</u>
07 - COLTS Wakeup	L68 COLTS issues it to "wake up" FNP COLTS and tell it to establish a DAC connection.	<code>dail_manager_\$standd_attach</code>
03 - Send Output	Transmit a message to FNP Colts where no immediate response is required.	<code>iox_\$put_chars</code>
04 - Send/Receive	Used for transmitting and receiving ICM's.	<code>iox_\$put_chars</code> followed by an <code>iox_\$get_chars</code>

```

read iom table          db_fnp_evel_
                        followed by
                        db_fnp_memory_$fetch

```

```

17 - Request line      Used to end a      dial_manager_$release_no_listen
disconnect             test page         dial_manager_$release

```

The above subset of GEROUTS are those used by the COLTS package to request certain Tolts actions.

## 6.2 Test Request

COLTS testing is initiated by calling TOLTS from the user terminal and inputting a COLTS test request. A COLTS test request contain FNP number plus sufficient channel/subchannel information to uniquely identify the unit:

test Cncss, where C = COLTS request n = FNP no., cc = FNP channel, ss = subchannel.

This is an adapter subchannel test request, where testing is requested for an FNP subchannel. The subchannel must be available either in the accept or ignore dialup state. Once the subchannel is assigned to the Tolts process all activity will be isolated from normal system activity.

## 6.3 Initialization & Validation

COLTS does some initial checking on the test request to ensure that syntax is correct and that parameters are consistent. It then checks the CDT to see if the FNP number "n" exists, and if so what it is - L6, 355, 18X - and what software it's running - MCS, NPS, GRTS, GRTS2 - and sets up some internal tables.

As part of initialization, COLTS issues a GEROUT remote inquiry which will be answered when testing lines are established.

COLTS maintains one executive line to the FNP COLTS and one additional line for each active test.

To establish its executive line, COLTS executes a GEROUT 07 to wake up FNP COLTS, which generates an I.D. for each executive line. In doing the GEROUT 07, COLTS passes a terminal type code unique to T&D. FNP Colts then does a

DAC connect to "COLTS" with the new I.D. In the L68, L68 COLTS sees a response to its outstanding remote inquiry (05) with this new I.D. It then executes a GEROUT 06 terminal type request, sees the type code that it passed with the (07) above, and the executive line is now established complete with I.D.

#### 6.4 Test Page Preprocessing

With the executive line established, COLTS can now proceed to get the test going. A start test ICM is issued (GEROUT 04) instructing FNP COLTS to perform the necessary allocation (subchannel) and establish a test page line I.D. If the return ICM indicates that a user is active on the subchannel, L68 COLTS informs the test caller and goes into a wait loop. If testing can proceed, the FNP test module is retrieved from system mass storage (different modules for different FNP software) and sent to the FNP via an ICM using the imbedded COLTS "link submodule" commands.

Meanwhile, an I.D. for the test page line has been established (when the "start test" ICM was sent) via the DAC procedure previously described. (COLTS always maintains an "active" remote inquiry). The L68 test page is now called into core and initialized, being passed line I.D., config info, etc., and testing begins.

#### 6.5 MULTICS Implementation

For Multics, L68 COLTS will be patterned after POLTS - a slave mode COLTS has been written for use on any O.S. This is a GMAP program. Interfacing to Multics will be a COLTS Interface Module written in pl/1 and will be a part of Tolts. Test pages use a GELBAR approach and are written in GMAP.

#### 7.0 FNP COLTS

##### Summary of FNP COLTS Operation

To illustrate a typical sequence of operations for FNP COLTS when a subchannel test has been initiated by the L68 the function performed and the sequence in which they occur are listed below.

#### 7.1 List of Basic Functions

1. Wake up FNP COLTS
2. Obtain channel number of channel to be tested
3. Establish subchannel interrupt processor
4. Take control of subchannel
5. Establish test line
6. Test subchannel
7. Wrapup test driver
8. Release resources and cleanup interface

## 7.2 Detailed Sequence OF FNP COLTS OPERATION

### 7.2.1 WAKE UP FNP COLTS

Activated by a call to dial\_mannager\_\$standd\_attach from Tolts. One of three returns will be expected: a dialup interrupt from the channel, channel is already attached to requesting process, or channel busy. If the channel is busy, ie. another user is testing a subchannel on the same FNP, the test request will be terminated. If a dialup interrupt is received or the colts channel is already attached to this process the test request will continue. As a result of the call to dial\_manager\_\$standd\_attach the channel (FNP Colt) is awakened and an ICM is sent to the L68 by executing the op\_block "signal sndout".

### 7.2.2 OBTAIN CHANNEL NUMBER of CHANNEL TO BE TESTED

The channel number of the channel to be tested will be transmitted to FNP Colt.

### 7.2.3 ESTABLISH SUBCHANNEL INTERRUPT PROCESSOR

The submodule will then be transmitted. There is a unique submodule for each type subchannel and it is the responsibility of the submodule to drive the subchannel much in the same fashion as hlsa\_man or lsla\_man does. When the submodule is received, FNP Colt will then execute the op\_block "holdot" which will prevent the output buffer in which the submodule resides from being destroyed.

#### 7.2.4 TAKE CONTROL OF SUCHANNEL

The submodule will then change the third word of the jump table (3WJT) to take over subchannel interrupts. FNP Colt will use the subchannels's TIB as the test TIB, and will save the operating system values. It will shut off the active status ICW in Hardware Communication Region (hwcm), and increment test driver users count in T&D table. A flag in the ICM to show test driver present is set and the op\_block "signal dialup" is executed. Then FNP Colts returns to the scheduler. FNP Colt then sets an "op complete" in the executive ICM and sets "accept direct input" in executive TIB. The checksum is computed and inserted in the ICM. The op\_block "signal sendout" is executed and control is returned to the scheduler.

#### 7.2.5 ESTABLISH TEST LINE

CColt will return through Tolts an ICM with a "terminal accepted" which will complete the establishment of the test line. FNP Colt will then create an ICM with a "connect to slave" and the op\_block "sndout" is executed, and control is returned to the scheduler.

#### 7.2.6 TEST SUBCHANNEL

Now that an environment has been created for CColt to have a dual communication path with FNP Colt, an executive and a test, testing can continue as it does in the GCOS and CP6 environment. Testing will begin with the reception of the "start subchannel test" ICM. As always the checksum is checked. Next the particular test driver is requested and received from CColt. Upon reception of the test driver an op\_block "holdot" is executed to prevent the buffer from being discarded. The HCR data is saved and a send and receive indirect control words (ICWS) are built. The CCT data received in the test ICM will be moved to the CCT buffer and build a new base address word (BAW) using the CCT buffer pointer from the test TIB. The active status ICW in the HCR is initialized and the config status words in the HCR are cleared. The new BAW, the primary and secondary, send and receive, ICWs are stored in the HCR. All counters and flags are cleared. Then the first peripheral control word (pcw) is gotten from test ICM and a connect is issued. The test TIB timer words are set up, and the timeout return pointer is set to "send ICM" routine. The op\_block "set\_timer" is executed and control is returned to the scheduler.

When an interrupt occurs the interrupt processor for the subchannel (part of the submodule) is called by an address computed by the third word of the jump table (3wjt). The ICM shutoff flag is checked for the presence of an ICM. If not present, exit to the master scheduler. If "wait for interrupt flag" is set in the ICM and desired number of interrupts have been received, clear the flag, and get the second and third PCWs from the test ICM and issue connects. If the number of statuses received is not equal to the number expected (ICM), exit to the scheduler. If test TIB timer has hung, exit to the scheduler. Clear timer value in test TIB and exit to the scheduler.

#### SEND ICM

Move a copy of final state of ICWs from the HCR and store the status store count and the interrupts count from SCR into a test ICM. Put a shutdown ICW in the HCR and set "ICM shutoff flag" in SCR. Set the reply "op complete" in the test ICM and the "accept direct input" op code in the test TIB and generate and store the checksum in the ICM. Execute the "sndin" op\_block to send the ICM to CColt and return to the scheduler. Repeat section 6 (above) until all test ICMs have been sent and the subchannel test is complete.

#### 7.2.9 WRAPUP TEST DRIVER

If the op code in the test ICM is "disconnect" (subchannel test is completed). The disconnect flag in test TIB will be set and the timer word cleared. Get test type number from test TIB and use as offset to get the test driver.

The test driver will restore original BAW and the config PCW to the HCR and issue a connect to restore the system config. Then get the reset mask PCW and issue a connect. The software Communication Region (SCR) counters, ICM shutoff flag, dummy status buffer in SCR are all cleared. Restore the 3wjt, which returns the subchannel interrupts to the FNP operating system. The submodule will then execute the op\_block "dmpout" to release the assigned buffers used for this subchannel's test. Execute the op\_block "signal hangup" to notify CColts that the subchannel can be returned to the system.

#### 7.2.11 RELEASE RESOURCES AND CLEANUP INTERFACE

Decrement the number of test driver users in the T&D table. If no test driver users now remain, FNP Colt can now begin to wrapup. It will execute the op\_block "dmpout" to release buffers, and return to event dispatcher. All COLTS testing is done.