Agilent Technologies

Agilent 3070 Increasing Throughput-Revised

(Date: 3/3/97: Revision: 20)

WHY IS MY Agilent 3070 BOARD TESTER "SLOW"?

WHY IS HIS Agilent 3070 BOARD TESTER "FAST"?

We've all seen cases where either another test system or another programmer can squeeze more speed out of a board test.

There are decisions one can make that causes an Agilent 3070 test program to be slower or faster than what Test Consultant generates automatically.

The Agilent 3070 is not, however, a "SLOW" system.

HOW MUCH IS ENOUGH?

The Agilent 3070 speed can, in some cases, outstrip the capability of humans or conventional handlers to "move" the boards.

A small board having 82 nodes, over 62 analog parts, one large mixed cluster test, and four boards per panel, was chosen for the experiment. A Series I Agilent 3070 with a 382 computer, Panel Test, and Throughput Multiplier options was used.

We tested these boards at the THROUGHPUT rate of one board every 1.5 seconds, including all overhead. If the handling problem was solved, that would be 288,000 boards every 5 days!

TEST ITEM	SINGLE WELL	DUAL WELL
Panel of Four	7.15	6.00
Single Board	1.79	1.50

Notice that this experiment was done on an older Series I. The throughput would be even higher on a new Series II.

WHY ISN'T THE AGILENT 3070 STANDARD TEST GENERATION AUTOMATICALLY FASTER?

1. Your Personal System Configuration Decisions; One Module vs. Two Modules, Throughput Multiplier,

Panel Test. Number of STC's, POTS channels, and number of DUT supplies.
2. Agilent is known for solid tests. These tests have safety margins built into them to make sure they work on multiple systems, whether that's your own multiple systems, or the test developed and shipped to you by one of Agilent's Channel Partners. This is critical when ramping up a new product line.
3. The ABILITY Factor. Repeatability, Stability, Transportability
4. Universal Testplan Generation. Common programming constructs, subroutine calls, etc. allow test engineers, Channel Partners, Contract Manufacturers, etc. to speak the same language.
5. Deterministic Test Speed. The programmer can trade off speed for other things. Tolerance Multiplier, VCL vs. TestJet vs. ConnectCheck, On Fail Exit, Fetch.
6. Series I versus Series II. While the Series I is comparable to the Series II in test capability, an upgrade to a faster controller, faster testhead hardware (ASRU "C" and new Control cards) should be considered, if throughput is of paramount importance.

Never forget that only Throughput is relevant to production. Because of this OVERHEAD TIME IS IMPORTANT! Do not confuse Test Time with Throughput. Test time will always be the smaller of the two.

If test times get small, it is not uncommon for OVERHEAD and HANDLING time to approach or even exceed test time.

NOTE: THROUGHPUT is a DIRECT FUNCTION of the following:
TEST TIME + OVERHEAD TIME + HANDLING TIME

Test Time: The average time taken to run the desired tests on a board.

Throughput: The average rate that verified boards come off the tester.

NO FIXTURE EXISTS:

If a fixture has not yet been built, some things which can increase throughput should be considered before starting.

A. Use Dual Well

Eliminates handling time AND can decreases Agilent 3070 overhead time by 50% (see 10 below). Throughput can as much as double!

NOTE: You are testing in one well while you are changing the board (or panel) in the second well. Parallel wiring is USUALLY used. Dual Well CAN CERTAINLY be used with the Throughput Multiplier.

B. Throughput Multiplier

Resultant throughput times will be about 1.6 times the single board time (with short test times).

NOTE: You CAN NOT test boards four times as fast! The actual improvement number depends

on the test speed, since overhead is relatively fixed. For this reason, slower throughput gives larger percentage improvement. For example if it took 10 seconds for one board, it would take 16 seconds to test four boards (or three or two), on a 4 module system this is 4 seconds per board.

Throughput Multiplier board size limitations;
Series I allows 72 nodes maximum in any one test and 648 nodes maximum per board
Series II allows 144 nodes maximum in any one test and 1296 nodes maximum per board

C. Test Multiple Boards

Spread overhead over more boards. If ONE fails, have a plan (you probably should fail that one only, and continue testing the other boards). If you wish to fail individual boards, you MUST be able to disconnect power to individual boards! This can be done with fixture electronics or by using multiple power supplies (the easiest way). Decide how you wish to do this BEFORE building the fixture. Testing boards in panels simplifies handling, but you must decide how to "retest" single boards.

D. Minimize Config

Use the minimum number of modules possible. Several operations, such as compiles, look at the board config file. The fewer modules seen, up to three, the faster the final board throughput will be (there's no difference between 3 and 4 modules).

NOTE: That the use of double density hybrid cards may facilitate this by allowing more nodes per module.

E. Tolerance Multiplier

If creating a new board test, consider this technique; Set the tolerance multiplier to .1, and remote sensing to on. Run Test Consultant. This will produce the maximum wiring for the tough tests (re-run at .2 or more if you get fixture overflow). Set the tolerance multiplier to 5, or even 10, and re-run Test Consultant. This will remove most 'en' and 'ed' options for faster, but less accurate, tests. The limits on capacitor tests must be watched when doing this, they can get really loose.

NOTE: This makes the FIXTURE MORE EXPENSIVE.

F. Hardware Changes

Don't forget that some hardware changes ALSO will affect speed. These should also be considered before building the fixture. Hardware changes are covered in items 4 to 9 below.

FIXTURE ALREADY EXISTS

In situations where a fixture already exists, there are still lots of ways to increase throughput. The ideas given below are not listed in any particular order, so start with those that are the easiest.

1. Use msec

Don't belabor insignificant times. Attack the items taking the most times. To find them try the following. Note it may be necessary to loop the test say 100 times if the test times are short.

```
Time = msec
call Digital_Tests
print "Digital time was ";(msec-Time)/1000
```

Amazingly enough this is a step many people, to their own detriment, skip. DON'T SKIP THIS! In the words of a famous general,
IDENTIFY THE ENEMY BEFORE YOU ATTACK.

It is also worth pointing out that 'print msec' yields a meaningless number that increments every millisecond. The value of msec equals zero certainly existed once, but it has not been seen since.

2. Measure the Time

Put the following just after the Wait_For_Start in the testplan to establish throughput times. print "Elapsed time = ";(msec-T)/1000 | T=msec. If you comment "wait for start" so it will loop, put the above line just before "call Test_Sections".

NOTE: This DOES NOT measure TEST time! It measures the inverse of Throughput (seconds/board), assuming the operator does NOT increase handling time.

3. Fewer DUT Supplies

Minimize MODULE supplies (watch power during transitions if dual well is used with a single supply). 10% increases in throughput have been seen.

Series II; Minimize the number used, and comment any unused supplies in the BOARD config file.

Series I; Minimize the number used, and comment any unused supplies in the TESTHEAD config file.

WARNING: If this is done, tests on other board types may fail. The testhead config file MAY need to be changed for each new fixture type.

HARDWARE:

The system hardware can also affect throughput. Whether these changes are worth their costs, only you the customer can decide.

4. Control Plus Cards

17% speed improvement over the Series I Control Card is typical. (See also items 18 and 38 below.) Note that Control and Control Plus cards may NOT be mixed within one testhead. Control Plus cards CAN be used in Series I testheads.

5. Controller

Controller memory and controller speed will affect test time in some cases (3% has been seen in non-telecom, 7% has been seen in POTS applications). An application was run using a 375 controller which took 11.60 seconds per board. Converting to a 382 controller lowered the time to 11.28 seconds. Use of a C110 controller decreases POTS test execution time by about 7%. B2.50 decreases time by another 5%.

6. DD Hybrid + ASRU C

Systems that contain BOTH Double Density hybrid cards, AND revision C ASRU cards will perform TestJet measurements USING THOSE CARDS about 10% faster than those using other cards.

NOTE: ASRU, ASRU B, and ASRU C cards may be mixed within one testhead with no problems. Although Single Density and Double Density Hybrid cards may be mixed and matched at will, all hybrid cards in a testhead must have the same vector application rate.

7. 8 Channel POTS Bay

If you are testing multichannel telecom cards having at least one analog side, you should be using an Agilent 3079CT with a POTS bay. Increasing the number of POTS hardware channels allows more parallel testing, increasing throughput. POTS bays are available in 2, 4, or 8 channel versions. Obviously an 8 channel upgrade could provide up to a 100% improvement over 4 channel, and up to 200% over a 2 channel, POTS bay. Typical improvements are in the 30% and 60% range.

8. Additional Modules

If adding additional modules allows the use of the Agilent Throughput Multiplier, then throughput can be increased by up to almost 300% (see item B above)

9. Additional STC Cards

Serial Test Cards were developed specifically to handle long streams of data such as found on many telecom boards. Each STC card can handle up to 4 bit streams. It is easy to make TWO

BER measurements on each stream AT THE SAME TIME. Since measuring BER is inherently slow, this can be a real time saver.

NOTE: It is possible to configure a testhead with up to 12 cards (no more than 3 per module), allowing 96 simultaneous BER measurements. We actually have had excellent results already with 7 STC cards transmitting 18 parallel streams into a board at one frequency, and making 18 BER measurements on the boards different frequency outputs, all in parallel.

TESTPLAN:

There are several changes that can be made within the testplan to increase throughput. Again, start with those that are the easiest. Remember it is usually not necessary to make every change to reach your throughput objectives.

10. Modify faon, faoff

Both Series II and Series I have a default wait (wait time IF no time is specified) on the faon commands of 1.5 seconds (faoff defaults to 0). Decrease these waits as much as possible. Beware of Pre-Shorts or Shorts failures on the first test however. In dual well operations, you can often eliminate all the faoff (fboff, fcoff, fdoff) commands.

NOTE: Executing an FAON command automatically sends F?OFF to all ports that should not be on. The ports that 'should be on' are defined with the 'vacuum well' statement.

11. Pins Test

This is set in the testplan in the subroutine 'sub Set_Custom_Options' as Chek_Point_Mode. If 'Pretest' is used, pins will be run on every board. Using 'Failures' is faster, but can still take 2.4 seconds in some cases. Consider doing this test on demand, or perhaps once per run; day; etc. Remember Pins test verifies the Fixture, not the Board.

12. Pretty PRINTs

Normally these collectively cost at least half a second. Are they needed? In your application can the screen PASS message can be eliminated?

NOTE: DO NOT eliminate these prints unless you need to; they make debug easier.

13. Turn Off Data Logging

On a four panel dual well test with a two panel throughput time of 14.5 seconds, and two panel test time of 9.9 seconds, a 1% sample rate with Histo increased the throughput times by 7.7%, while a 10% sample rate increased it 14.3%!

NOTE: Even when the sample rate was 0, Histo still slowed throughput by 5.4% due to added

overhead. The proper way to set the sample rate depends on your failure rate, run size, and volume per day. The rate sets the number of GOOD boards logged, ALL failures are logged if HISTO is selected. If you test 10,000 boards per day with a .1% failure rate, you might decide you wanted a sample of 100 good boards to compare the 10 bad boards against. That would be 100/10000 = .01= 1 %.

14. BSCAN Power Cycle

The standard Series I testplan contains a power cycle just after the 'call BScan_Incircuit_Tests'. Don't do this unless you have to reset the board. With the Series II it's automatically optional. Give it thought, do you need it?

15. Group Test Statements

A sync occurs at the end of each 'block' of test statements. Minimizing the number of syncs will speed up test times. Group 'test' statements with NO intervening commands (including 'if boardfailed then subexit') for best speed. Blank lines and comments are no problem.

16. Empty Calls

Eliminate all calls to empty subroutines.

17. Disc Access Time

To determine if disc access time is a problem, first measure the average time to make one testplan run. Then turn OBJECT CHECKING off (in the testplan) and recheck times. If they are faster, access is slowing down testing. Object checking insures that only the latest .o's are used, a good idea during debug. Unfortunately this can add delays of .5 seconds or more.

NOTE: The 'load board' command automatically turns Object Checking on. See also item 43 below. Accessing board directories over a net will really highlight this difference. Remember not to include any nrun = 1 times when doing timed comparisons.

18. Testhead Memory

You can put the command given below several places in the testplan to see how much memory each section uses by viewing the CHANGE in the number. If the number doesn't change, you've run out of memory.

NOTE: Each test that doesn't fit in memory is run like a 'first run time test'; and this is usually slow. The command is a debug tool only, and may leave the testhead in an unknown state. Doing a clear after EACH use is recommended. Memory is cleared by 'testhead is *', or by exiting basic. A fairly complete clear can also be done by executing the command 'testhead power on'.

NOTE: Standard Series I Control Cards have about 570k of user available memory, and Series II Control Plus Cards have about 3,133k (the E3986A upgrade has almost 16,000k). Sometimes reordering tests will improve speed by having the 'littlest' test be the one run using 'first run times'. As an alternative perhaps eliminate some tests.

--- testhead status A,B,C,D,E,F \ print F ---

19. Debug Safeguard

Time test execution with "safeguard cool". Then time test execution with "safeguard none". If the time is different, try to debug safeguard. Disable upstream devices to eliminate warnings. Remove any duplicate safeguard statements in testplan. Be very careful with the use of "safeguard none". Using Delay For Cooling in tests, rather than Safeguard Cool in the subroutine, saves about 30 msec per test.

NOTE: Segmented tests use the assigned safeguard level for EACH SEGMENT. If Safeguard Cool is used, tests can be really slow.

20. sps Waits

The default on the Series II is to turn supplies on in parallel (optimize). If this is not done, the Series I and II both default to a 30 msec wait. If the optimize option isn't used, and there are multiple supplies, put wa0 on all but final sps if possible.

TESTS THEMSELVES:

21. GP Relays

Minimize the usage of GP relays where possible. Connect times are about 50msec, but vary a lot. Assume between 1msec and 100msec each to be safe. Also note that gpconnects and disconnects take about 35m sec LESS TIME (each) when done in the analog section of a functional test, rather than the testplan. Note however that it takes about 38m sec to call an isolated powered analog test from basic.

22. Modify Pre_Shorts

Remove tests for open jumpers. These should be tested for automatically in the Shorts file anyway. If you use "report common devices", the message should identify any failing jumper.

23. Modify Analog Tests

BEFORE DOING THIS, make sure that the 'sub analog' routines time really is significant. Most people start here, and THEY ARE USUALLY WRONG to do so.

If "analog/" tests take too long, try the following. Be sure the resulting tests are stable after

modification. It is strongly recommended that you 'rerun' Board Test Grader (see Test Development Tools manual) when you have 'finished' modifying the tests.

Try removing the 'ed' option.
--Tests run about three times faster.
--Note 'ed' is required if test frequency is 128 Hz.

Try removing the 'en' option.
--Tests run about two and a half times faster.
-- Note 'en' should always be used with remote sensing.

NOTE: Removing both en and ed, if present, results in tests about 17 times faster (per test)

Remove any possible waits (for example wa.2. Saves twice the wait time.

Remove any 'dwa' uses you can. Saves specified time.

Remove the 'fi' option if possible. Note this is NOT really a filter, it averages the specified number of samples. fi 10 therefore takes ten readings!

Specify the range (for example ar.2) to prevent autorange times. Saves up to 3m seconds per range.

Any tests which use the adjust option will wait for the operator. Eliminate them if possible.

See also item 38.

24. Expect Option

Use the expect option to select the proper detector range in the analog part of functional tests to eliminate autoranging time.

25. On Failure Exit

Use of this command in analog functional tests will slow tests down, even with NO FAILURES. Use it sparingly. On Failure / Off Failure, and On Failure Report in analog tests are obviously also slow.

26. Single Step ARB

Normally the single step mode uses lots of triggers, and switching between analog and digital in a mixed test. Each switch takes time.

27. Digitizer

If large numbers of variables are passed, tests are slowed. Use the Digitizer only when justified. It may be worth timing several methods.

NOTE: There are some things which are very hard to do unless the ARB or Digitizer is used. Don't avoid using them, just consider all the possible options.

28. Modify TestJet

Insure that "default throughput adjustment 1" is at the start of each TestJet test. Increases speed from about 200 pins/sec to about 500 pins/sec.

29. Use VCL Defaults

Any changes to the "family" information will slow the test time because the system can not use its precalculated values (slew, levels, or load).

30. Vector Cycle 50ns*X

Use increments of 50 nanoseconds for both the vector cycle and receive delay for best speed.

31. Don't Pass Variables

Passing variables in STL, VCL, or ATL takes time. Use static data where possible.

32. Analog "Fetch"

Tests using the initiate, fetch, report analog, statements are SLOWER than those using measure.

33. Avoid Test Pauses

While the pause command is often critical in order to get certain tests working, any tests using pause will be slower than those without it.

34. Modify Mixed Tests

In mixed tests, minimize the number of times you change from VCL to ATL and visa versa. This saves only about 3.5msec per pair, so weigh the speed benefits against the debugging trouble. If possible, change mixed tests to simple analog or digital tests. These are normally faster.

35. Modify Serial Sections

In the serial section of mixed tests, group tests such that you minimize turning loop holds on and

off. Also changing connectivity takes time, so do as much in each individual test as possible.

36. Modify "Series" Tests

IF Throughput Multiplier is used, modify any tests that are running in "series". Series tests generate UNIX sync commands and affect throughput when the Throughput Multiplier option is used. See the manual section on Test Strategy for the Agilent Throughput Multiplier.

digital tests compiled with debug
digital tests containing sync pulses
analog potentiometer adjustments
tests with pause
tests with initiate, fetch, report analog
tests with on failure exit
tests using a variable
TestJet tests containing over 1000 pins (use 2 tests)

37. Discharge Routines

WARNING: BE CAREFUL; erroneous changes can damage both the tester and the boards under test. Check each test; any with entry voltages HIGHER than the maximum possible on that node might be eliminated. Also it is usually quickest to discharge really large caps with an external resistor and a GP relay.

38. Board Config Wrong

If the Testhead Config file lists a Control Plus card, and the Board Config file list only a Control Card, then the system automatically adds an additional 3msec delay in EVERY unpowered analog test. To eliminate the added delay, either; change the Board Config file, or add "minimum wait 0" to the testplan. Doing either will probably mean you will need to debug a few tests because of the additional Control Plus speed.

39. Unneeded Sections

OP Amp and Comparator tests often contain multiple sections, only one of which is applicable to a particular board application. They also may contain tests to insure that the input voltage will not damage the device. Spend the time to figure out which sections are needed, and comment the rest.

PROCEDURAL:

There are several common procedures which should be considered that can affect throughput.

40. Recompile All .d's

Recompile all tests having a .d. Tests compiled with debug option run slower than normal compiles, EVEN WHEN .d's ARE REMOVED.

41. Ignore Code Size

Many people equate lines of code with speed. This is NOT usually the case. Use all the lines of code you need. For example with multiple revs you can put multiple IF THEN statements in the analog subroutine. This creates many syncs. A slightly better way is to duplicate, and modify, the entire analog subroutine; then just call whichever routine applies. For those with B2:50 or above software, enable and use the Agilent Multiple Board Versions software. This will actually allow testing at the same speed as would having separate directories for each revision!

42. Test Consultant

Rerun Test Consultant. This may find changes you thought you made, but were not implemented yet (for example Control Plus Card recompiles). ALWAYS run it when you think your finished, to avoid a time bomb waiting for the next ECO to come.

43. Other Users

Make sure any speed problems aren't due to other users. Experiment with temporarily disconnecting the Agilent 3070 from the LAN. A customer once paid for a factory engineer to fly to his facility, only to prove that "other users" were drastically affecting his production throughput.

44. Using String Names

Using string concatenation in test names runs SLOW.
If A$= 'analog/u3', and B$ = 'u3', then;
test 'analog/u3', and test A$, run at the same speed.
test 'analog/'&B$ runs about 300msec slower,
and it ALWAYS runs with an nrun = 1 time

45. Initializations

Some processors need time to awaken. Are there other tests you can do first? Sometimes test ordering can speed things up quite a bit. Sometimes tricky things can be done with dual wells.

46. Action Upon Failures

This affects real production throughput, often dramatically. Several things are worth considering. See also item C above. Non-deterministic tests (safeguard can not estimate their length, for example those with homing loops). Add a TEST TIME statement in the VCL section. The time is easily determined by timing a non-failing test. Add say 10%, and use that for a test time entry. The TIMEOUT statement in the testplan becomes really important when external instruments

are used. You should use a considered value, the default of 5 seconds is often longer than really needed.

47. Sampling Theory

Do you need to test every part every time?

Perhaps do a "full" test on one board per run/day/etc, and just perform certain tests needed due to the fault spectrum on all other boards?

If there are lots of parts on a board of one part number, could you just test the first and last placed of those parts on each board? This would highlight bad reels.

Maybe you can test the first and last placed of each part number with an accurate test, and use a really loose limit fast test for all the other same part number board parts.

HANDLING:

Board handling is often the final stumbling block on high throughput lines. As mentioned at the start, we have tested four board panels at 1.5 seconds per board. Even with dual wells this leaves just 6 seconds to remove a tested board and replace it with an untested one; quite a feat.

48. Handling

A model 44990A (Agilent handler) will move one panel in and out in 10 seconds. It is capable of handling a TWO module Throughput Multiplier test. It's strength is high mix, low volume, production lines.

TSL makes a model AFS400 handler that will move one panel in and out in 8 seconds. It is capable of handling a FOUR module Throughput Multiplier test.

The new Philips handler, model PA5067 will move two panels in and out in 8 seconds (or four seconds per panel). It is capable of handling a FOUR module Throughput Multiplier test.

A comparison is difficult. A few assumptions, which certainly DO NOT universally apply, will enable some comparison at least. Let us assume that we want the maximum throughput, testing only one kind of board, for long periods of time. The board is available in panels of ANY size (a really bad assumption), we will use the pass through mode, and the individual board test times given below include Agilent 3070 overhead time. The theoretical Agilent 3070 throughput limit is; ('single board test time' x 1.6) / 4 modules

THROUGHPUT:

Assuming a board test time of 5 seconds yields;

Theoretical limit	= 2 sec/board
Agilent Handler	(10 + (5*1.6))/2 = 9 sec/board
TSL Handler	(8 + (5*1.6))/4 = 4 sec/board
Philips Handler	(4 + (5*1.6))/4 = 3 sec/board

Assuming a board test time of 2 seconds yields;

Theoretical limit	= 0.8 sec/board
Agilent Handler	(10 + (2*1.6))/2 = 6.6 sec/board
TSL Handler	(8 + (2*1.6))/4 = 2.8 sec/board

Return to top of page