

# 自动测试系统 BTS

自动测试系统报价分为两部分：自动流程控制系统和测试工具集

自动流程控制系统主要包括：

1. UUT Management
  - a) 支持对 UUT 增删搜索
  - b) 在工具集配合下，支持 UUT 的硬件信息的自动监测
  - c) 支持 UUT 信息的编辑
  - d) 支持 UUT 信息和模板信息的比对
2. Test Case Management
  - a) 支持自动 Test Case 以及手动 Test Case 的增删改
  - b) 预置 Windows2008/2012 自动安装 Case
  - c) 预置 RHEL6.x 自动安装 Case
  - d) 如果购买了工具集，会预置各测试工具对应的自动 Test Case
3. Test Execution Management
  - a) Test Set 结构管理，以及增删改操作
  - b) 支持多种自动执行模式，Loop/Time/Schedule 错误处理机制
  - c) 测试运行管理
4. Test Report
  - a) Excel 测试报告生成和下载

测试工具集合：

基本测试工具集如下表。

Module	Test Items	Win (2008&2012)	Linux (Redhad6.5)	Description
NIC	LOOPBACK Test	✓	✓	程序提供同台主机上的两个 nic port 之间的数据传输，可以无需架设 server 服务，实现单机的 nic 传输检测功能： ※ switch 连接测试模式。待测 port 均连接在 switch 上进行测试 ※ point to point 连接测试模式。待测 port 两两相连进行测试 ※ 可以切换网卡连接速率，测试不同速率下的传输功能。 ※ 提供一定的容错率
	MAC Detect	✓	✓	检测网卡的 MAC 地址： ※ 检测网卡的系统 MAC 地址是否有效 ※ 检测网卡的系统 MAC 地址是否有重复 ※ 检测网卡的系统 MAC 地址是否在预设厂商范围内 ※ 检测网卡的固件中烧录的 MAC 地址是否有效 ※ 检测网卡的固件中烧录的 MAC 地址是否有重复
	Transfer			
	IPTest	✓	✓	输入一个远端主机 ip，然后测试本机网卡到这个远端主机的通讯。达到 ping 测试功能的目的。
	Loss Tolerance	✓	✓	网卡丢包率测试。这项测试使用 c/s 架构，在远程主机上启动 ITCNic_server 服务，然后本地主机的网卡与远端服务器传输数据包，并计算这个测试过程的丢包率。
	Speed	✓	✓	测试网卡当前连接速度是否与期望值相符。（10M/100M/1G half/full）
	TCP_Transfer	✓	✓	测试网卡 TCP 传输。这项测试使用 c/s 架构，在远程主机上启动 ITCNic_server 服务，然后本地主机的网卡与远端服务器使用 TCP 协议进行数据传输，达到测试网卡 TCP 传输功能的目的
	UDP_Transfer	✓	✓	测试网卡 UDP 传输。这项测试使用 c/s 架构，在远程主机上启动 ITCNic_server 服务，然后本地主机的网卡与远端服务器使用 UDP 协议进行传输，达到测试网卡 UDP 传输功能的目的

Module	Test Items	Win (2008&2012)	Linux (Redhad6.5)	Description
Memory				获取内存产生的可修复错误 (Correctable error) 和不可修复错误(Uncorrectable error)的数量, 可以定位到错误发生的具体位置 (CPU number -> Channel number -> DIMM number)。可以支持目前最新的Intel Romley, Grantly和Brickland平台。
	ECC	✓	✓	
	Address Line	✓	✓	對內存地址線進行測試,在內存中依次寫入一個遞增的值,然後讀出校驗
	ATSplus	✓	✓	本算法提供了兩個獨立的過程write (stepA) 和check (stepB); write過程用來根據參數進行內存的寫操作, check則是用來校驗的
	Bit Stuck	✓	✓	通過對比向內存中寫入和從內存中讀出的測試Pattern是否相同來判斷內存是否存在物理問題。(測試Pattern是兩個值,一個是各位均為0,另一各是各位均為1).
	Bus Noise	✓	✓	通過對比向內存中寫入和從內存中讀出的測試Pattern是否相同來判斷內存是否存在物理問題,以一段大小的內存裏的數據作為測試pattern,把這段數據填充到待測的內存區域,然後再讀出並檢查,如果對應地址的數據一致則測試通過,否則失敗。
	Checker Board	✓	✓	該項測試通過對比向內存中寫入和從內存中讀出的測試Pattern是否相同來判斷內存是否存在物理問題。(測試Pattern是一組數據,而且每個數據的各個位的值是0,1交錯的),將多種測試Pattern依次寫入全部內存中,然後讀出校驗
	Constant	✓	✓	通過寫入常數的方法測試內存,將一個常數連續的寫入到測試內存中,讀出并校驗結果是否正確。該常數由用戶在界面定義
	Copy	✓	✓	通過內存拷貝的方法測試內存,將某隨機數連續寫入內存的前一半,同時將該數取反寫入後一半,拷貝前一半到後一半,校驗,後一半的值,用同樣的方法校驗前一半
	Data Bus	✓	✓	對內存數據線進行測試,將測試內存依次按照BYTE/WORD/DWORD置1,然後讀出校驗
	Increment	✓	✓	通過將遞增的序列連續寫入內存的方法測試內存,將一個遞增的BYTE(WORD,DWORD)型序列連續寫入內存中;然後讀出并校驗
	Inverse	✓	✓	將數據寫入全部內存中再取反重新寫入的方法測試內存,將某常數寫入內存,取反後重新寫入,讀出并校驗;該常數由客戶在界面設定
	Jump	✓	✓	按照一定的規則跳躍性的選擇內存測試地址進行讀寫測試,根據任一值被另一個值異或兩次該值的內容保持不變的基本規律實現
	March			March系列算法使用相同的語言符號來描述,元素如下: ↑: 從低地址向高地址作後續的讀寫操作 ↓: 從高地址向低地址作後續的讀寫操作 l: 讀寫操作的順序是任意的,可以是↑,也可以是↓ r: 讀操作 r0: 讀取一個內存單元,期望的讀出數值是0 r1: 讀取一個內存單元,期望的讀出數值是1 w: 寫操作 w0: 將數值0寫入指定內存單元 w1: 將數值1寫入指定內存單元 上述r和w操作的基本單位是memory device width (word)。下述算法均用上文的語言符號來描述
	March A	✓	✓	{ ll(w0); ↑(r0,w1,w0,w1); ↑(r1,w0,w1); ↓(r1,w0,w1,w0); ↓(r0,w1,w0) }
	March B	✓	✓	{ ↑(w0); ↑(r0,w1,r1,w0,r0,w1); ↑(r1,w0,w1); ↓(r1,w0,w1,w0); ↓(r0,w1,w0) }
	March C	✓	✓	{ ll(w0); ↑(r0,w1); ↑(r1,w0); ll(r0); ↓(r0,w1); ↓(r1,w0); ll(r0) }
	March LR	✓	✓	{ ↓(w0); ↓(r0,w1); ↑(r1,w0,r0,w1); ↑(r1,w0); ↑(r0,w1,r1,w0); ↑(r0) }
	March R	✓	✓	{ ↑(w0); ↑(r0,w1); ↑(r1,w0); ↑(r0,w1); ↑(r1) }
	March IntraCFs		✓	基於march算法框架針對DRAM chips intra couple faults進行檢測 針對inversion coupling fault, state coupling fault, Disturb coupling fault和idempotent coupling fault這四種基本的CF錯誤模型我們總結出了相應的最優校驗pattern序列。應用march算法基本原則對針對以上四種錯誤的pattern序列進行讀寫校驗以確保規避這四種錯誤的出現。
	Moving Inversion	✓	✓	測試步驟: (1) 將待測內存塊寫入預設的pattern數值 (2) 從最低地址開始作如下a)和b)操作步驟,直至到達最高地址結束 a) 檢測步驟(1)寫入的pattern是否正確 b) 將當前pattern的反碼寫入當前內存位置 (3) 從最高地址開始作如下a)和b)操作步驟,直至到達最低地址結束 a) 檢測步驟(2,b)寫入的pattern是否正確 b) 將當前pattern的反碼寫入當前內存位置
	Memory Address	✓	✓	將每個內存單元的內容寫為這個單元的物理地址,然後讀取查看是否寫入正確,寫入的基本單元是一個memory bus width。
	memorycpuLoading	✓	✓	同時對memory和CPU進行壓力測試,使Server的運行功耗達到最大值,從而達到測試Server在高压、高功耗狀態下穩定性的目的。為了達到這個目標,MemoryCPUTesting在每顆CPU上開啟兩個線程進行壓力測試,線程1進行memory壓力測試,線程2進行CPU壓力測試。其中,memory壓力測試,通過持續的內存拷貝指令來增加內存的使用率;CPU壓力測試,通過持續的Streaming SIMD Extensions 4 (SSE4) transfer指令來增加CPU的佔用率,SSE4指令主要用來完成多媒體處理和高性能計算。
	Memory Speed	✓	✓	計算CPU訪問memory、L1 cache、L2 cache的速度。 其中,對於memory訪問速度的計算,MemorySpeed通過MMX data transfer指令來進行CPU寄存器和memory間的数据傳輸,通過指令執行時間和数据傳輸量來計算memory訪問速度。MMX指令集主要用來完成doubleword(4字节)和quadword(8字节)為單位的CPU寄存器和memory間的数据傳輸功能。 對於L1和L2 cache通過SSE transfer指令集來完成單精度浮點數在CPU寄存器和memory間的傳輸,傳輸数据总量分別設置為L1和L2 cache的大小,從而保證真正的数据transfer是在CPU寄存器和L1/L2 cache中完成,同樣也通過傳輸時間和傳輸数据总量來計算L1/L2 cache的訪問速度。
	NUMA	✓	✓	根據系統具體的numa架構進行內存讀寫測試。 根據系統numa node個數生成相應測試線程,每個測試線程分別綁定一個與之對應的numa node,有這些線程在自己綁定的numa node的端內存中分配待測內存進行讀寫操作,讀寫校驗正確testitem pass,其中任何一個線程出現讀寫錯誤則fail。
	Power Two Inverse	✓	✓	先初始化所有內存為1,再全部寫為0,然後對每一位做寫1再恢復為0的操作,最後檢查內存是否全部為零
	Power Two	✓	✓	先初始化所有內存為零,再全部寫為1;然後對每一位做寫0再恢復為1的操作,最後檢查內存是否全部為1
Pseudo Random	✓	✓	通過對比向內存中寫入和從內存中讀出的測試Pattern是否相同來判斷內存是否存在物理問題。(測試項的測試Pattern是一個隨機數),將一個隨機數連續的寫入到內存中,然後讀出校驗。	
Random Address	✓	✓	通過隨機取地址寫入隨機數的方法測試內存,基於內存的首地址,隨機產生偏移量,並在該位置寫入隨機數,重新產生相同的地址偏移量校驗	
Random Cycle	✓	✓	通過先寫入隨機圖樣再隨機取地址校驗的方法測試內存。在要測試的buffer中寫入隨機序列,然後隨機產生地址偏移量進行校驗,並在該地之中寫入新的隨機數,繼續進行下一次的測試	
			通過在內存中寫入隨機圖樣再校驗的方法測試內存。把隨機數填入內存中,然後讀出并且校	

Module	Test Items	Win (2008&2012)	Linux (Redhad6.5)	Description
HardDisk				
	IO Pattern Test	✓		※此项测试采用伪随机算法对硬盘扇区进行读写测试。先将特定扇区的数据读出，然后写入读出的数据。或者当用户选择Use Pattern Test时，则将特定扇区的数据读出备份，然后写入指定pattern data，并读取比较。最后将备份的数据恢复到原位置。 ※用户可以通过调整参数来决定测试次数，测试中的传输数据大小和测试的pattern data ※此项测试为自动测试方式
	Bus Speed	✓		※测试方法：通过多次读取同一硬盘，计算其传输速度，并与期望值比较 ※用户可以通过调整参数来决定期望值的上限与下限 ※此项测试为自动测试方式
	DST Test	✓	✓	※DST也称为Long Self Test，主要用于检查硬盘电气和机械性能以及磁盘的读取性能。通过ATA or SCSI命令开启硬盘自带自检测试，分为电气、机械性能扫描和全盘读取两部分，消耗时间长（全盘读取耗时较长，且硬盘容量越大，时间越长），由程序跟踪收集测试中是否有error生成。发生错误时程序会收集错误信息及sense key。本测试至少需要1小时以上。 ※用户可以通过调整参数来限定测试时间，到达一定时间自动停止测试 ※此项测试为自动测试方式
	Quick DST Test	✓	✓	※Quick DST也称为Short Self Test，主要用于检查硬盘电气和机械性能以及磁盘的读取性能。通过ATA or SCSI命令开启硬盘自带自检测试，后台检测硬盘，分为电气、机械性能扫描和随机读取两部分，消耗时间短（约2-4分钟），由程序跟踪收集测试中是否有error生成。发生错误时程序会收集错误信息及sense key。一般测试时间大约需要2分钟（有些硬盘多于2分钟）。 ※用户可以通过调整参数来决定fail时是否重新测试和判定fail的sense key ※此项测试为自动测试方式
	Quick Test	✓	✓	※对硬盘上的扇区选择一部分（1%）进行测试，先将扇区的数据读出，然后写入读出的数据 ※此项测试为自动测试方式
	Random Read	✓	✓	※此项测试采用伪随机算法对硬盘扇区进行读写测试，检测硬盘是否存在坏扇区。 ※用户可以通过调整参数来决定测试模式和测试中的传输数据大小a ※此项测试为自动测试方式
	Random Write	✓	✓	※此项测试采用伪随机算法对硬盘扇区进行读写测试。先将特定扇区的数据读出，然后写入读出的数据。或者当用户选择Use Pattern Test时，则将特定扇区的数据读出备份，然后写入指定pattern data，并读取比较。最后将备份的数据恢复到原位置。 ※用户可以通过调整参数来决定测试模式，测试中的传输数据大小和测试的pattern data ※此项测试为自动测试方式
	Sequential Read	✓	✓	※此项测试采用顺序递增算法对硬盘扇区进行读写测试，检测硬盘是否存在坏扇区。 ※用户可以通过调整参数来决定测试模式，测试起始扇区位置，测试终止扇区位置和测试中的传输数据大小 ※此项测试为自动测试方式
	Sequential Write	✓	✓	※此项测试采用顺序递增算法对硬盘扇区进行读写测试。先将特定扇区的数据读出，然后写入读出的数据。或者当用户选择Use Pattern Test时，则将特定扇区的数据读出备份，然后写入指定pattern data，并读取比较。最后将备份的数据恢复到原位置。 ※用户可以通过调整参数来决定测试模式，测试起始扇区位置，测试终止扇区位置，测试中的传输数据大小和测试的pattern data ※此项测试为自动测试方式
	S-M-A-R-T	✓	✓	※本测试为S. M. A. R. T属性检查，通过ATA or SCSI的命令读取S. M. A. R. T. 信息。该项测试一旦发生读取属性值失败、Error Log中出现错误记录和某些项值超出预期，测试会FAIL。 ※用户可以通过调整参数来决定是否测试SMART error log和属性测试的模式 ※此项测试为自动测试方式
	Temperature	✓	✓	※测试方法：通过硬盘SMART功能读取硬盘温度，并与期望值比较 ※用户可以通过调整参数来决定期望值的上限与下限 ※此项测试为自动测试方式

Module	Test Items	Win (2008&2012)	Linux (Redhad6.5)	Description
CPU	64BitOperation	✓	✓	進行64位整數運算以確定運算器的功能是否達到要求 彙編執行整數的加減乘除運算,並校驗.
	BasicFunction	✓	✓	
	NPUFunction	✓	✓	執行浮點數的加減乘除,並校驗.
	SpeedStep		✓	實現方法 Test CPU speedstep function by setting processor p-states. 彙編調用CPU各指令集中"mov"類相關指令,完成數據傳輸校驗.實現對CPU各指令集的測試.因調用多為SIMD指令,幫測試對顯著提升CPU POWER.
	TransferFunction	✓	✓	
	Floating	✓	✓	
	QPI(HyperTransp ort) Frequency			獲取目標系統的所有的socket的所有的qpi通道的current frequency value。
	QPILaneWidth			獲取目標系統的所有的socket的所有的qpi通道的Tx data lane和Rx data lane的width值并与标准值进行比较。
Mainboard	ACPI		✓	通过对RSDT ( XSDT ) , FADT , FACS和DSDT的查找以及对各个表格checksum的验证来测试ACPI是否正常
	Clock&Calendar (RTC)	✓	✓	通过比较系统运行时间和现实运行时间差来判断RTC是否正常
	CMOS RAM	✓	✓	通过将相应模式的数据写入CMOS,再读出验证一致性,测试CMOS RAM是否正常
	DMA Transfer	✓	✓	通过对Hard disk进行读写测试DMA Channel
	Interval Time	✓	✓	测试计数器的实际工作频率是否在理论计数频率内,实际频率会因软件产生延时,延时随主机频率不同而产生变化
Video Port	Video Port	✓	✓	※测试Server的Video Port工作是否正常 ※支持单Video Port & 双Video Port测试 ※采用交互测试的方式,测试时屏幕随机显示RGB三色及3个随机数字,用户通过计算3个数字和的方式选择测试结果。 ※自动留档保存,方便日后查看
Video Memory	Native Video Memory	✓	✓	向显存中写入各种测试Pattern,然后再从相应位置读出并与写入Pattern比对。 ※可指定地址范围进行测试 ※可随机选取显存大小进行测试 ※可选择不同的分辨率及色深进行测试 ※可选择不同的测试Pattern进行测试。可选Pattern: 0x55, 0x00, 0x5A, 0xAA, 0xFF, 0xA5 (用户可任意选定)
PCIE			✓	监测 PCIE 设备 AER Correctable 和 UnCorrectable Error Demon形式提供, 监测频率可配置
system error check			✓	check 系统SEL, demessage,edac, pcei error,mce log