**Base SMP Power Management Impacts on Kernel Use of Hardware Timers**

|  |  |
| --- | --- |
| Status: | Released |
| Version: | 1.00 |
|  |  |
| Team/Department : | Kernel/Core Development |

Contents

1 Introduction 3

2 Timer and Core Frequencies 3

3 NKern::Timestamp 4

3.1 Using Global Timer 4

3.2 Using a SOC Timers 5

4 Impacts of Idle Management and Core Retirement 6

5 Dynamic Voltage and Frequency Scaling 7

5.1 Informing the Kernel about Frequency Changes 7

6 Testing 8

6.1 Timestamp Testing 8

6.2 Frequency Change Testing 11

7 References 11

8 Document History 12

9 Appendix – Using two 32 bit timer in combination 12

10 Appendix – Sample runs of tests 12

# Introduction

In ARM based SMP systems, power management can impact the kernel’s ability to timestamp and to provide constant frequency thread time slicing.

The kernel uses ARM local timers to provide thread time slicing. The timers are ideal for this purpose as they are private per CPU and this, amongst other things, enables parallel rescheduling across multiple CPUs to take place.

The kernel also needs to use a timer for the purpose of time stamping, which in turn is used in other areas like load balancing, tracing and power management. In some systems such as CortexA9MP a CPU global timer is available. Where possible the global timer is used otherwise the baseport needs to provide access to a SOC timer. Whichever source is used for time stamping, the underlying timer needs to be fast and high resolution[[1]](#footnote-1).

Power management can have an impact on both types of timer. In some architectures (CortexA9MP for example) the frequency of the core can directly affect the frequency of local and global timers. Consequently dynamic frequency scaling can have a direct effect on time slice durations and time stamping. In heterogeneous systems, changes in core frequency also change the relative processing capacity of a CPU with respect to other cores.

Idle management also has an effect, when a core is idled it will stop clocking, when all cores are idle the system would be put into a low power mode and fast timing sources will generally be disabled. This only affects time slicing if the timer control register can be lost due to the low power mode entry. The timer value is not relevant as the idle thread does not time slice. For time stamping however, saving and restoring the timer control data is not enough. The baseport needs to take care to restart the time stamping timer on wake up, and to set it to a value that reflects the amount of time spent idle.

The following sections provide explanations and sample code on how the kernel represents timer frequencies, how to implement the nanokernel timestamp API, how to tackle the effects of idle management and dynamic frequency scaling, and finally how to test these features are working.

# Timer and Core Frequencies

When the SMP kernel is in the process of initialising, the baseport provides it key timing information using the Variant Interface Block or VIB (see [1]). Of particular relevance to this topic there are the following fields that the baseport must provide:

**TUint64 iMaxCpuClock:** maximum possible CPU clock frequency

**TUint32 iMaxTimerClock:** maximum possible local (or global) timer clock frequency

It is possible for the system to boot running a frequency that is different to the maximum. In fact in a heterogenous system it would be possible to even boot with different cores running at different speeds. To allow for these variations the scheduler and subschedulers control blocks contain ratios that convert from a maximum frequency to a current frequency. The ratios are stored as two numbers, one specifying a mantissa and the other an exponent (or shift). Essentially:

Current frequency = (mantissa\*maximum frequency)>> -exponent

This eases the assembler manipulation of these frequency numbers resulting in faster code. See the SRatio class in nklib.cia and nklib.cpp for more details. On boot the baseport can specify what the relative frequency is for each core and each timer using these ratios. To this end the VIB has the following fields:

**SRatio\* iTimerFreqR[KMaxCpus]:** for each CPU, this array holds a ratio that converts from iMaxTimerClock to the current frequency of the local timer on that CPU

**SRatio\* iCpuFreqR[KMaxCpus]:** for each CPU, this array holds a ratio that converts from iMaxCpuClock to the current frequency of that CPU

**SRatio\* iGTimerFreqR:**  ratio to convert between global timer frequency as a fraction of iMaxTimerClock

If the baseport does not specify a ratio then maximum frequency is assumed. This is typically the case as most boards boot at full speed. Therefore, with regard to timing on an ARM system, a typical VIB initialisation will only specify the addresses of the timers, the maximum frequencies and ratio pointers will all be set to NULL. For example:

SVariantInterfaceBlock VIB; // will be initialised to 0

SVariantInterfaceBlock\* InitVIB()

{

SVariantInterfaceBlock\* v = &VIB;

...

v->iMaxCpuClock = TUint64(KMaxCoreFrequency);

v->iMaxTimerClock = KMaxCoreFrequency/2u;

...

//address of per-CPU timer (must be same for all CPUs)

v->iLocalTimerAddr = KCpuPrivateRegionBase + 0x600u;

v->iGlobalTimerAddr = KCpuPrivateRegionBase + 0x200u;

...

return v;

}

On section 5 we show how to VIB ratios can be used at run time to inform the kernel of changes in core frequency.

# NKern::Timestamp

The SMP nanokernel provides two new functions for use in time stamping :

TUint64 NKern::Timestamp()

TUint32 NKern::TimestampFrequency();

The Timestamp functions returns a 64 bit timestamp. The frequency of the timestamp is returned by TimestampFrequency. The API is used internally by various components such as the load balancer, measuring thread CPU time (as returned by RThread:: GetCpuTime and NKern:: ThreadCpuTime) and kernel stats (which are used for power management). It is used in various other components, notably btrace.

The baseport decides how this API is implemented and the frequency of the timer. Currently there are two approaches.

## Using Global Timer

On CortexA9MP (>=r1p0) there is a global timer available with a generic interface. The SMP nanokernel has a built in implementation that allows the timer to be read, and for appropriate compensations to take place when the frequency changes. From a baseporter’s perspective this is the simplest option.

To use this option the variant.mmh should have the following lines:

macro \_\_CPU\_ARM\_HAS\_GLOBAL\_TIMER\_BLOCK

macro \_\_NKERN\_TIMESTAMP\_USE\_SCU\_GLOBAL\_TIMER\_\_

The baseport also needs to populate the VIB (VIB see [1]) fields to provide the nanokernel with the address of the global timer. If the global timer was running at a frequency that is lower than iMaxTimerClock, then the VIB iGTimerFreqR needs to be initialised e.g:

// Assume platform is booting af half speed for some obscure reason

// somewhere in variant

SVariantInterfaceBlock VIB; // will be initialised to zero

// we are booting at half speed

static SRatio gBootRatio;

SVariantInterfaceBlock\* InitVIB()

{

SVariantInterfaceBlock\* v = &VIB;

v->iVer = 0;

v->iSize = sizeof(VIB);

v->iMaxCpuClock = TUint64(KMaxCoreFrequency);

v->iMaxTimerClock = KMaxCoreFrequency/2u;

...

//address of per-CPU timer (must be same for all CPUs)

v->iLocalTimerAddr = KCpuPrivateRegionBase + 0x600u;

v->iGlobalTimerAddr = KCpuPrivateRegionBase + 0x200u;

...

gBootRatio.Set(1,1) // set ratio to 1/2

for (TInt i=0; i< KNumberOfCpus; i++)

{

v->iTimerFreqR[i] = &gBootRatio;

v->iCpuFreqR[i] = &gBootRatio;

}

v->iGTimerFreqR = &gBootRatio;

return v;

}

The kernel start the global timer on boot. It also sets the prescaler to 1 on global and local timers. Unless the VIB ratios are initialised by the baseport, they are all assumed to be 1. So the kernel assumes that normally timers and cores would boot at their maximum frequencies, which is typically the case.

## Using a SOC Timers

If the global timer is not available then the baseport can provide an implementation for NKern::Timestamp using a SOC timer. In thi case variant.mmh needs to include the following:

macro \_\_NKERN\_TIMESTAMP\_USE\_INLINE\_BSP\_CODE\_\_

#define AsspNKernIncludePath SYMBIAN\_OS\_LAYER\_PLATFORM\_EXPORT\_PATH(assp/<assp name>/nkern)

The baseport also needs to supply a header containing the implementation. It is possible to supply an assembler or a C++ implementation. The header has the following format:

// Define a timestamp frequency in the following constant

const TUint32 KTimestampFrequency = ... ;

#ifdef \_\_DEFINE\_NKERN\_TIMESTAMP\_ASM\_\_

/\* If NKern::Timestamp() is written in assembler define it here

\*/

#endif

#ifdef \_\_DEFINE\_NKERN\_TIMESTAMP\_CPP\_\_

/\* If NKern::Timestamp() is written in C++ define it here

\*/

#endif

The baseport first needs to supply a value for KTimestampFrequency, then provide an implementation for NKern::Timestamp either in assembler or in C++ in the appropriate section. The header then needs to be exported by the bld.inf to the AsspNKernIncludePath directory. The kernel will then pick up this file in its compilation.

In some boards there are no 64 bit timers available. One scheme to get around this problem is to use two 32 bit timers in combination (see appendix 9), or to combine a single timer with the nanokernel tick count.

# Impacts of Idle Management and Core Retirement

To save power, fast time sources can be switched off when a platform enters a retention mode. Consequently the baseport power management adaptation code needs to look after this eventuality. No kernel APIs are required, as the baseport implementation decides which timer (global or SOC) is used for the timestamp API. The baseport’s power adaptation also provides the code which enters a low power mode, and which runs after wakeup.

Before entry into a low mode the baseport should do as follows:

1. If this low power mode does not affect the timer move onto low power mode entry
2. Otherwise save timer context. This might involve all of the timer control registers as well as its present count and frequency.

In an SMP system step 2 would usually be carried out by the last CPU to go idle. Upon wakeup from a low power mode that requires timer restoration the baseport would do as follows:

1. Ascertain how much time has been spent sleeping. This is already required for the purpose of advancing kernel time. Though fast time sources are not generally available in low power modes, slow time sources (most often at 32.768Khz) normally are.
2. Restore the timer, but reset its current value to include the amount of time spent sleeping.

Note that in this effective change of timer frequency, from a high to low and back again, will introduce minor errors in the NKern::Timestamp API. These can be minimised but never completely eliminated. For the purpose of load balancing these should be small enough to not present any discernable impact. However the timestamp is also used for other purposes such as btrace for example. The fact that there is an accompanying NKern::TimestampFrequency could lead to programmers assuming an exact time source. However with current technology, in a mobile device requiring good power management, such as thing is impossible. Expectations therefore should be clearly set in the relevant documentation.

A similar level of support would be expected for CPU local timers used in timeslicing. In this case it is not necessary to take into account the amount of time slept. That is the timer count does not need restoration. However any timer control registers, dictating settings such whether the timer is running, pre/post scalars and so on need to be restored if the sleep mode will result in these settings being lost. This is particularly likely to occur when a CPU gets retired. The retiring code should backup the control register, so that it can be restored when the CPU is reengaged.

# Dynamic Voltage and Frequency Scaling

DVFS can have a number of effects on the kernel. In some devices, changes in core frequency can translate into changes in global and local timer frequency. As the kernel uses the local timers for time slicing, this change, unless it is countered, can result in changes in time slice periods. In these systems if the global timer is used for time stamping, the kernel needs to be informed of frequency changes so that that timestamp frequency remains invariant to DVFS. Finally CPU speed can also have an impact on load balancing.

## Informing the Kernel about Frequency Changes

The VIB exposes an API for this purpose. It is expected that the baseport would use this API when the baseport changes speed. To call the function, the following function pointer from the VIB is used:

**TFrequencyChangeFn iFrqChgFn**: function to notify frequency changes

The API takes no parameters, instead the ratios for the new frequencies are provided using the ratio members in the VIB, just as it can be done at boot time (see section 2).

Let’s consider for example a homogeneous system which uses a global timer and which supports three frequencies 600Mhz, 300Mhz and 192Mhz. The baseport adaptation code would create three ratios:

SRatio iFullSpeed; // used when running at 600Mhz

SRatio iHalfSpeed; // used when running at 300Mhz

SRatio iLowSpeed; // used when running at 192Mhz

. . .

iFullSpeed.Set(1);

iHalfSpeed.Set(1,1);

iLowSpeed.Set(2748779069,33); // 192/600 == 8/25 == (8\*(2^33)/25)/(2^33)

On frequency change:

// imaginary class name

void DXXXDVFSManager::FrequencyChagedDfcFn(TAny\* aParam)

{

...

TInt nc = NKern::NumberOfCpus();

// assume class has a member pointing to VIB

// notify nanokernel of frequency changes

SRatio\* ratio = (newfreq == KFullSpeed) ? &iFullSpeed) :

((newfreq == KHalfSpeed) ? &iHalfSpeed : & iLowSpeed);

for (i=0; i<nc; ++i)

{

iVIB->iTimerFreqR[i] = ratio;

// don’t really need following line in a homogeneous system

// iVIB->iCpuFreqR[i] = ratio;

}

iVIB->iGTimerFreqR = ratio;

(\*iVIB->iFrqChgFn)();

...

}

To prevent corruptions in the VIB data, the baseport should ensure that that changes to the VIB ratios and the following call to the frequency change function always happen together, and are not pre-empted by a similar code elsewhere. In the example above the call and the changes to the VIB ratios are done inside a DFC for this purpose. It is assumed this would be the most common pattern. The interface actually allows different ratios to be changed from different cores or threads, ie thread 1 only ever modifies ratios for CPU1 and thread 2 for CPU2 and so on. But changes of the same ratios in different threads would race.

# Testing

## Timestamp Testing

The e32test t\_timestamp allows baseports to check that the timestamp is working properly when power management is enabled. To be able to use the baseport needs to supply a pdd. If the pdd is not present the test is simply skipped.

The test basically proceeds as follows:

1. Take a timestamp. T0, using NKern::Timestamp
2. Wait for an interval (usually 5s)
3. Take another timestamp, T1, and check that the following two conditions are true:

* (wait interval)\* timestamp frequency - %error < (T1-T0) < (wait interval)\* timestamp frequency + %error
* If the above condition fails the test will fail and exit
* Power management saved and restored the timestamp timer. If this check fails, the test will loop and retry. After a configurable number of retries the test will fail and exit

1. If the above success for a configurable number of times the test will pass

The PDD interface that needs to be implemented looks like this:

/\*

\* Hold time stamp information, for now just freq

\*/

struct STimestampTestConfig

{

TUint32 iFreq;

TUint iIterations;

TUint iRetries;

TUint iTimerDurationS;

TUint iErrorPercent;

};

/\*

\* DTimestampTestPddChannel

\* Interface Pdd

\*/

class DTimestampTestPddChannel : public DBase

{

public:

/\*

\* reset any variables requried to check entry into a low power mode

\* between time of call and time at which iTimer will next expire

\*/

virtual void StartLPMEntryCheck() = 0;

/\*

\* @return ETrue if a low power modes was entered

\*/

virtual TBool EndLPMEntryCheck() = 0;

/\*

\* Provide test parameters

\*/

virtual void TestConfig(STimestampTestConfig& aInfo) = 0;

};

The PDD can use the TestConfig function to override the default parameters used in the test:

1. %Error which is deemed acceptable (defaults to 1% iErrorPercent)
2. Number of iterations the test for it to pass (defaults to 5 iIterations)
3. Number of retries to attempt if idle is not entered (defaults to 4 iRetries)
4. Duration of wait interval (defaults to 5s iTimerDurationS)

The PDD then only needs to provide two functions

**TBool EndLPMEntryCheck:** returns true if test entered a lower power mode which caused save and restore of the timestamp implementing timer.

**void StartLPMEntryCheck:** Utility function called before the test enters the wait period. This can be used if any variables need to be reset before EndLPMEntryCheck is called again.

For example let’s assume that hardware uses the global timer and supports three low power modes, WFI, SLEEP and DEEP SLEEP. And let’s suppose that only the latter two can cause a reset of the global timer. And finally, let’s assume that the power controller implementation counts how many times each low power mode is entered and offers APIs to collect these counts. The PDD implementation would take the following form:

...

#include "d\_timestamp.h"

#include "d\_timestamp\_dev.h"

#include --- power controller include file ---

...

class DXXXTimestampTestPddChannel : public DTimestampTestPddChannel

{

public:

// Inherited from DTimestampTestPddChanel. These called by the LDD.

virtual void StartLPMEntryCheck();

virtual TBool EndLPMEntryCheck();

virtual void TestConfig(STimestampTestConfig& aInfo);

private:

TUint iSleeCount;

TUint iDeepSleepCount;

};

/\*\*

Logical Device (factory class) for DXXXTimestampTestPddChannel

\*/

class DXXXTimestampTestPddFactory : public DPhysicalDevice

{

...

};

...

////

// Channel implementation

/\*\*

Called before each cycle in the test. Takes a copy of current idle count in power controller

\*/

void DXXXTimestampTestPddChannel::StartLPMEntryCheck()

{

iSleepCount = TXXXTBPowerController::SleepCount();

iDeepSleepCount = TXXXTBPowerController::DeepSleepCount();

}

/\*\*

Called at the end of each cycle. Should return true if we have entered idle since call to

StartLPMEntryCheck. This will be the case if the idle count has changed

\*/

TBool DXXXTimestampTestPddChannel::EndLPMEntryCheck()

{

return (iSleepCount!=TXXXTBPowerController::SleepCount() ||

iDeepSleepCount!= TXXXTBPowerController::DeepSleepCount());

}

/\*\*

Called to allow baseport to override test parameters.

\*/

void DXXXTimestampTestPddChannel::TestConfig(STimestampTestConfig& aInfo)

{

};

For a sample of a run of this test see Appendix 10.

## Frequency Change Testing

The t\_freqchng test is available for this purpose. The test checks the correctness of the SRatio implementation, and on ARM SMP systems the integrity of time slicing and NKern::Timestamp[[2]](#footnote-2) against DVFS. The test does this by disabling DVFS in the platform if it is present and then using the timer prescalers to simulate frequency changes. In a similar vein to the t\_timestamp test, the test uses a driver which is extended by the baseport. In particular the baseport needs to provide some additional functions if DVFS is supported. If the driver is not present the test is skipped.

The baseport needs to provide an mmp file which includes the d\_frqchg mmh file and any other sources it may need to provide the missing functions:

// Description:

// d\_frqchg.mmp

#include <variant\_test.mmh>

#include "../../../../../kernelhwsrv/kerneltest/e32test/group/d\_frqchg.mmh"

macro \_\_PLATFORM\_SUPPORTS\_DVFS\_\_

sourcepath ./.

source d\_frqchg\_dvfs.cpp

library <whichever library has functions to control dvfs>

The variant\_test.mmh would include the baseports variant.mmh, from this the test driver can pick up whether the global timer is supported and whether the platform is ARM SMP. The lines in red are only necessary if the platform supports DVFS. In that case the baseport needs to supply functions to disable and reenable the DVFS. These are used so that DVFS does not interfere with the test, which simulated frequency changes by using the timer’s prescalers. An example follows:

#include <kernel/kernel.h>

#include <header for library which controls dvfs>

TInt DisableDvfs()

{

return DXXXDVFSManager::DisableDVFS( ... );

}

void RestoreDvfs()

{

DXXXDVFSManager::EnableDVFS( ...);

}

#endif

Note that the SHAI PM services API could be used for this purpose.

For a sample of a run of this test see Appendix 10.

# References

| No. | Document Reference | Version | Description |
| --- | --- | --- | --- |
| [1] | BSP Development for SMP (\sf\os\kernelhwsrv\userlibandfileserver\basedocs) |  |  |

# Document History

| Date | Version | Status | Description |
| --- | --- | --- | --- |
| 28-06-2010 | 1.00 | Released | Minor edits mode to initial draft. |
| 27-06-2010 | 0.01 | Draft | Initial draft |

# Appendix – Using two 32 bit timer in combination

If a 64 bit timer is not available, it is possible to use two 32 bits timers in combination.

Suppose that two timers T1 and T2 are available and that they count down and wrap. T1 is initialised at 0xffffffff and T2 is instead initialised to 0xffffefff with a reload value of 0xffffefff. The timers are started very close to each other, T2 is started first. So at any point in time, the difference between the two timers will be:

T1 – T2 = 0x1000+startError+number of times T2 has wrapped\*0x1000

Where startError is the difference in starting time between T1 and T2. Thus before T2 wraps you’d expect to see a difference of 0x1000+startError after the first wrap this would grow to 0x2000+startError and so on. We can count up to 0x100000 wraps of T2 giving a total resolution of 0x100000\*0xfffff000, not quite 64 bit but much better than 32bit. In theory a much smaller gap could be chosen between the timers. We could instead initialise T2 to 0xffffffef. This would give far greater resolution (0x10000000\*0xfffffff0). However there needs to be enough gap left between the timers to accommodate for the difference in time it takes to start the timers and to read them in the NKern::Timestamp function itself. There is another reason for having wide gap between the timers. At any point in time NKern::Timestamp or code starting the timers could be interrupted by secure mode code. Therefore the gap has been made wide enough to be detect an interruption. The timestamp code would works as follows:

1. Disable interrupts
2. Read T1
3. Read T2. This is read last so error is in same direction as the start error.
4. work out the difference ANDED with 0xfff
5. If the difference is > (longest possible secure mode incursion) then assume that secure mode has interrupted the read and start again.
6. Return (0xffffefff-T2) + ((T1-T2)>>16) \* 0xfffff000;

With a gap of 0x1000 between the timers, and at 10Mhz this will work so long as secure mode incursions do not exceed 409.6uS. If it is possible for this period to be exceeded, then the code could be changed so that it works as follows:

1. Read T1
2. Read T2
3. Read T1 again

If the difference in T1 value between read at step 1 and read at step 3 exceeds the maximum time a secure mode incursion can take, then goto 1. This approach does give the full resolution of the timer to measure a secure mode incursion but has the disadvantage of requiring an extra read from the timer value register.

# Appendix – Sample runs of tests

This section shows sample results for t\_timestamp and t\_frqchg on a platform which supports global timer.

**t\_timestamp**

RTEST TITLE: T\_TIMESTAMP 2.00(3090)

Epoc/32 2.00(3090)

RTEST: Level 001

Next test - Timestamp accuracy test

RTEST: Level 002

Next test - Get timestamp frequency

RTEST: Level 003

Next test - Get nanotick frequency

tick period in uS== 1000

running at 300000000Hz for 5 interations, each lasting 5 seconds and with 4 retries

expecting 1500000000 with up to 30000000 error

RTEST: Level 004

Next test - test timer interval

Got 1503999255 expected 1500000000, LPM Entered:1, error 3999255 is OK

Got 1504000272 expected 1500000000, LPM Entered:1, error 4000272 is OK

Got 1503999789 expected 1500000000, LPM Entered:1, error 3999789 is OK

Got 1504000705 expected 1500000000, LPM Entered:1, error 4000705 is OK

Got 1503999325 expected 1500000000, LPM Entered:1, error 3999325 is OK

Unloading LDD

Unloading PDD: TIMESTAMPTEST.XXXX

RTEST: SUCCESS : T\_TIMESTAMP test completed O.K.

**t\_frqchg**

RTEST TITLE: T\_FRQCHG 2.00(3090)

Epoc/32 2.00(3090)

RTEST: Level 001

Next test - Testing

Test1 00000001 0

R0X,R0: 80000000 ffe1 80000000 ffe1

Testing 103 integers

R1X,R1: 80000000 ffe1 80000000 ffe1

Testing 103 integers

Test1 00000003 0

R0X,R0: c0000000 ffe2 c0000000 ffe2

Testing 103 integers

R1X,R1: aaaaaaab ffdf aaaaaaab ffdf

Testing 103 integers

Test1 b504f334 32

R0X,R0: b504f334 ffe0 b504f334 ffe0

Testing 103 integers

R1X,R1: b504f334 ffe1 b504f334 ffe1

Testing 103 integers

Test1 c90fdaa2 30

R0X,R0: c90fdaa2 ffe2 c90fdaa2 ffe2

Testing 103 integers

R1X,R1: a2f9836e ffdf a2f9836e ffdf

Testing 103 integers

Test1 0000000a 0

R0X,R0: a0000000 ffe4 a0000000 ffe4

Testing 103 integers

R1X,R1: cccccccd ffdd cccccccd ffdd

Testing 103 integers

Test1 cccccccd 35

R0X,R0: cccccccd ffdd cccccccd ffdd

Testing 103 integers

R1X,R1: a0000000 ffe4 a0000000 ffe4

Testing 103 integers

Test1 00000064 0

R0X,R0: c8000000 ffe7 c8000000 ffe7

Testing 103 integers

R1X,R1: a3d70a3d ffda a3d70a3d ffda

Testing 103 integers

Test1 a3d70a3d 38

R0X,R0: a3d70a3d ffda a3d70a3d ffda

Testing 103 integers

R1X,R1: c8000001 ffe7 c8000001 ffe7

Testing 103 integers

Timeslicing test ...

RTEST: Level 002

Next test - Baseline - expecting normal

ID: 0 IV: 14998555 (= 49995us) TIV 0us

ID: 1 IV: 14998866 (= 49996us) TIV 49996us

ID: 0 IV: 14999450 (= 49998us) TIV 49998us

ID: 1 IV: 14999533 (= 49998us) TIV 49998us

ID: 0 IV: 14999573 (= 49999us) TIV 49999us

ID: 1 IV: 14999606 (= 49999us) TIV 49999us

ID: 0 IV: 14999601 (= 49999us) TIV 49999us

ID: 1 IV: 14999580 (= 49999us) TIV 49999us

ID: 0 IV: 14999566 (= 49999us) TIV 49999us

ID: 1 IV: 14999584 (= 49999us) TIV 49999us

ID: 0 IV: 14999583 (= 49999us) TIV 49999us

ID: 1 IV: 14999585 (= 49999us) TIV 49999us

ID: 0 IV: 14999698 (= 49999us) TIV 49999us

ID: 1 IV: 14999685 (= 49999us) TIV 49999us

ID: 0 IV: 14999587 (= 49999us) TIV 49999us

ID: 1 IV: 14999577 (= 49999us) TIV 49999us

ID: 0 IV: 14999576 (= 49999us) TIV 49999us

ID: 1 IV: 14999667 (= 49999us) TIV 49999us

ID: 0 IV: 14999602 (= 49999us) TIV 49999us

ID: 1 IV: 14999573 (= 49999us) TIV 49999us

ID: 0 IV: 14999476 (= 49998us) TIV 49998us

ID: 1 IV: 14999663 (= 49999us) TIV 49999us

ID: 0 IV: 14999606 (= 49999us) TIV 49999us

ID: 1 IV: 14999681 (= 49999us) TIV 49999us

ID: 0 IV: 14999488 (= 49998us) TIV 49998us

ID: 1 IV: 14999576 (= 49999us) TIV 49999us

ID: 0 IV: 14999586 (= 49999us) TIV 49999us

ID: 1 IV: 14999563 (= 49999us) TIV 49999us

ID: 0 IV: 14999485 (= 49998us) TIV 49998us

ID: 1 IV: 14999585 (= 49999us) TIV 49999us

ID: 0 IV: 14999570 (= 49999us) TIV 49999us

ID: 1 IV: 14999563 (= 49999us) TIV 49999us

ID: 0 IV: 14999695 (= 49999us) TIV 49999us

ID: 1 IV: 14999702 (= 49999us) TIV 49999us

ID: 0 IV: 14999678 (= 49999us) TIV 49999us

ID: 1 IV: 14999695 (= 49999us) TIV 49999us

ID: 0 IV: 14999675 (= 49999us) TIV 49999us

ID: 1 IV: 14999607 (= 49999us) TIV 49999us

ID: 0 IV: 14999579 (= 49999us) TIV 49999us

ID: 1 IV: 14999692 (= 49999us) TIV 49999us

ID: 0 IV: 14999564 (= 49999us) TIV 49999us

ID: 1 IV: 14999681 (= 49999us) TIV 49999us

ID: 0 IV: 14999601 (= 49999us) TIV 49999us

ID: 1 IV: 14999681 (= 49999us) TIV 49999us

ID: 0 IV: 14999577 (= 49999us) TIV 49999us

ID: 1 IV: 14999569 (= 49999us) TIV 49999us

ID: 0 IV: 14999572 (= 49999us) TIV 49999us

ID: 1 IV: 14999583 (= 49999us) TIV 49999us

ID: 0 IV: 14999580 (= 49999us) TIV 49999us

ID: 1 IV: 14999659 (= 49999us) TIV 49999us

ID: 0 IV: 14999686 (= 49999us) TIV 49999us

ID: 1 IV: 14999563 (= 49999us) TIV 49999us

ID: 0 IV: 14999500 (= 49998us) TIV 49998us

ID: 1 IV: 14999692 (= 49999us) TIV 49999us

ID: 0 IV: 14999491 (= 49998us) TIV 49998us

ID: 1 IV: 14999681 (= 49999us) TIV 49999us

ID: 0 IV: 14999579 (= 49999us) TIV 49999us

ID: 1 IV: 14999488 (= 49998us) TIV 49998us

ID: 0 IV: 14999675 (= 49999us) TIV 49999us

ID: 1 IV: 14999586 (= 49999us) TIV 49999us

ID: 0 IV: 14999624 (= 49999us) TIV 49999us

ID: 1 IV: 14999687 (= 49999us) TIV 49999us

ID: 0 IV: 14999497 (= 49998us) TIV 49998us

ID: 1 IV: 14999581 (= 49999us) TIV 49999us

ID: 0 IV: 14999570 (= 49999us) TIV 49999us

ID: 1 IV: 14999589 (= 49999us) TIV 49999us

ID: 0 IV: 14999562 (= 49999us) TIV 49999us

ID: 1 IV: 14999582 (= 49999us) TIV 49999us

ID: 0 IV: 14999667 (= 49999us) TIV 49999us

ID: 1 IV: 14999683 (= 49999us) TIV 49999us

ID: 0 IV: 14999681 (= 49999us) TIV 49999us

ID: 1 IV: 14999567 (= 49999us) TIV 49999us

ID: 0 IV: 14999598 (= 49999us) TIV 49999us

ID: 1 IV: 14999599 (= 49999us) TIV 49999us

ID: 0 IV: 14999590 (= 49999us) TIV 49999us

ID: 1 IV: 14999588 (= 49999us) TIV 49999us

ID: 0 IV: 14999581 (= 49999us) TIV 49999us

ID: 1 IV: 14999567 (= 49999us) TIV 49999us

ID: 0 IV: 14999581 (= 49999us) TIV 49999us

ID: 1 IV: 14999671 (= 49999us) TIV 49999us

ID: 0 IV: 14999582 (= 49999us) TIV 49999us

ID: 1 IV: 14999674 (= 49999us) TIV 49999us

ID: 0 IV: 14999605 (= 49999us) TIV 49999us

ID: 1 IV: 14999581 (= 49999us) TIV 49999us

ID: 0 IV: 14999481 (= 49998us) TIV 49998us

ID: 1 IV: 14999581 (= 49999us) TIV 49999us

ID: 0 IV: 14999566 (= 49999us) TIV 49999us

ID: 1 IV: 14999673 (= 49999us) TIV 49999us

ID: 0 IV: 14999568 (= 49999us) TIV 49999us

ID: 1 IV: 14999570 (= 49999us) TIV 49999us

ID: 0 IV: 14999566 (= 49999us) TIV 49999us

ID: 1 IV: 14999476 (= 49998us) TIV 49998us

ID: 0 IV: 14999609 (= 49999us) TIV 49999us

ID: 1 IV: 14999676 (= 49999us) TIV 49999us

ID: 0 IV: 14999681 (= 49999us) TIV 49999us

ID: 1 IV: 14999598 (= 49999us) TIV 49999us

ID: 0 IV: 14999571 (= 49999us) TIV 49999us

ID: 1 IV: 14999577 (= 49999us) TIV 49999us

ID: 0 IV: 14999679 (= 49999us) TIV 49999us

ID: 1 IV: 14999575 (= 49999us) TIV 49999us

RANGE 49996-49999us (3us)

RTEST: Level 003

Next test - expecting double

ID: 1 IV: 29998603 (= 99995us) TIV 0us

ID: 0 IV: 29998963 (= 99997us) TIV 99997us

ID: 1 IV: 29999434 (= 99998us) TIV 99998us

ID: 0 IV: 29999533 (= 99998us) TIV 99998us

ID: 1 IV: 29999557 (= 99999us) TIV 99999us

ID: 0 IV: 29999587 (= 99999us) TIV 99999us

ID: 1 IV: 29999638 (= 99999us) TIV 99999us

ID: 0 IV: 29999634 (= 99999us) TIV 99999us

ID: 1 IV: 29999620 (= 99999us) TIV 99999us

ID: 0 IV: 29999615 (= 99999us) TIV 99999us

ID: 1 IV: 29999662 (= 99999us) TIV 99999us

ID: 0 IV: 29999554 (= 99999us) TIV 99999us

ID: 1 IV: 29999540 (= 99998us) TIV 99998us

ID: 0 IV: 29999626 (= 99999us) TIV 99999us

ID: 1 IV: 29999617 (= 99999us) TIV 99999us

ID: 0 IV: 29999551 (= 99999us) TIV 99999us

ID: 1 IV: 29999646 (= 99999us) TIV 99999us

ID: 0 IV: 29999642 (= 99999us) TIV 99999us

ID: 1 IV: 29999633 (= 99999us) TIV 99999us

ID: 0 IV: 29999657 (= 99999us) TIV 99999us

ID: 1 IV: 29999564 (= 99999us) TIV 99999us

ID: 0 IV: 29999553 (= 99999us) TIV 99999us

ID: 1 IV: 29999633 (= 99999us) TIV 99999us

ID: 0 IV: 29999642 (= 99999us) TIV 99999us

ID: 1 IV: 29999628 (= 99999us) TIV 99999us

ID: 0 IV: 29999668 (= 99999us) TIV 99999us

ID: 1 IV: 29999626 (= 99999us) TIV 99999us

ID: 0 IV: 29999625 (= 99999us) TIV 99999us

ID: 1 IV: 29999611 (= 99999us) TIV 99999us

ID: 0 IV: 29999633 (= 99999us) TIV 99999us

ID: 1 IV: 29999554 (= 99999us) TIV 99999us

ID: 0 IV: 29999638 (= 99999us) TIV 99999us

ID: 1 IV: 29999541 (= 99998us) TIV 99998us

ID: 0 IV: 29999618 (= 99999us) TIV 99999us

ID: 1 IV: 29999617 (= 99999us) TIV 99999us

ID: 0 IV: 29999658 (= 99999us) TIV 99999us

ID: 1 IV: 29999634 (= 99999us) TIV 99999us

ID: 0 IV: 29999623 (= 99999us) TIV 99999us

ID: 1 IV: 29999641 (= 99999us) TIV 99999us

ID: 0 IV: 29999606 (= 99999us) TIV 99999us

ID: 1 IV: 29999572 (= 99999us) TIV 99999us

ID: 0 IV: 29999632 (= 99999us) TIV 99999us

ID: 1 IV: 29999518 (= 99998us) TIV 99998us

ID: 0 IV: 29999614 (= 99999us) TIV 99999us

ID: 1 IV: 29999628 (= 99999us) TIV 99999us

ID: 0 IV: 29999666 (= 99999us) TIV 99999us

ID: 1 IV: 29999648 (= 99999us) TIV 99999us

ID: 0 IV: 29999625 (= 99999us) TIV 99999us

ID: 1 IV: 29999631 (= 99999us) TIV 99999us

ID: 0 IV: 29999588 (= 99999us) TIV 99999us

ID: 1 IV: 29999644 (= 99999us) TIV 99999us

ID: 0 IV: 29999635 (= 99999us) TIV 99999us

ID: 1 IV: 29999630 (= 99999us) TIV 99999us

ID: 0 IV: 29999643 (= 99999us) TIV 99999us

ID: 1 IV: 29999627 (= 99999us) TIV 99999us

ID: 0 IV: 29999668 (= 99999us) TIV 99999us

ID: 1 IV: 29999634 (= 99999us) TIV 99999us

ID: 0 IV: 29999524 (= 99998us) TIV 99998us

ID: 1 IV: 29999625 (= 99999us) TIV 99999us

ID: 0 IV: 29999538 (= 99998us) TIV 99998us

ID: 1 IV: 29999657 (= 99999us) TIV 99999us

ID: 0 IV: 29999654 (= 99999us) TIV 99999us

ID: 1 IV: 29999618 (= 99999us) TIV 99999us

ID: 0 IV: 29999633 (= 99999us) TIV 99999us

ID: 1 IV: 29999613 (= 99999us) TIV 99999us

ID: 0 IV: 29999591 (= 99999us) TIV 99999us

ID: 1 IV: 29999635 (= 99999us) TIV 99999us

ID: 0 IV: 29999636 (= 99999us) TIV 99999us

ID: 1 IV: 29999639 (= 99999us) TIV 99999us

ID: 0 IV: 29999617 (= 99999us) TIV 99999us

ID: 1 IV: 29999580 (= 99999us) TIV 99999us

ID: 0 IV: 29999637 (= 99999us) TIV 99999us

ID: 1 IV: 29999522 (= 99998us) TIV 99998us

ID: 0 IV: 29999624 (= 99999us) TIV 99999us

ID: 1 IV: 29999624 (= 99999us) TIV 99999us

ID: 0 IV: 29999671 (= 99999us) TIV 99999us

ID: 1 IV: 29999640 (= 99999us) TIV 99999us

ID: 0 IV: 29999506 (= 99998us) TIV 99998us

ID: 1 IV: 29999631 (= 99999us) TIV 99999us

ID: 0 IV: 29999674 (= 99999us) TIV 99999us

ID: 1 IV: 29999546 (= 99998us) TIV 99998us

ID: 0 IV: 29999626 (= 99999us) TIV 99999us

ID: 1 IV: 29999623 (= 99999us) TIV 99999us

ID: 0 IV: 29999624 (= 99999us) TIV 99999us

ID: 1 IV: 29999716 (= 99999us) TIV 99999us

ID: 0 IV: 29999655 (= 99999us) TIV 99999us

ID: 1 IV: 29999524 (= 99998us) TIV 99998us

ID: 0 IV: 29999529 (= 99998us) TIV 99998us

ID: 1 IV: 29999639 (= 99999us) TIV 99999us

ID: 0 IV: 29999617 (= 99999us) TIV 99999us

ID: 1 IV: 29999580 (= 99999us) TIV 99999us

ID: 0 IV: 29999638 (= 99999us) TIV 99999us

ID: 1 IV: 29999534 (= 99998us) TIV 99998us

ID: 0 IV: 29999630 (= 99999us) TIV 99999us

ID: 1 IV: 29999636 (= 99999us) TIV 99999us

ID: 0 IV: 29999561 (= 99999us) TIV 99999us

ID: 1 IV: 29999557 (= 99999us) TIV 99999us

ID: 0 IV: 29999627 (= 99999us) TIV 99999us

ID: 1 IV: 29999626 (= 99999us) TIV 99999us

ID: 0 IV: 29999628 (= 99999us) TIV 99999us

RANGE 99997-99999us (2us)

RTEST: Level 004

Next test - expecting normal again

ID: 1 IV: 14997823 (= 49993us) TIV 0us

ID: 0 IV: 14998947 (= 49996us) TIV 49996us

ID: 1 IV: 14999500 (= 49998us) TIV 49998us

ID: 0 IV: 14999556 (= 49999us) TIV 49999us

ID: 1 IV: 14999573 (= 49999us) TIV 49999us

ID: 0 IV: 14999601 (= 49999us) TIV 49999us

ID: 1 IV: 14999565 (= 49999us) TIV 49999us

ID: 0 IV: 14999582 (= 49999us) TIV 49999us

ID: 1 IV: 14999467 (= 49998us) TIV 49998us

ID: 0 IV: 14999494 (= 49998us) TIV 49998us

ID: 1 IV: 14999388 (= 49998us) TIV 49998us

ID: 0 IV: 14999573 (= 49999us) TIV 49999us

ID: 1 IV: 14999613 (= 49999us) TIV 49999us

ID: 0 IV: 14999683 (= 49999us) TIV 49999us

ID: 1 IV: 14999586 (= 49999us) TIV 49999us

ID: 0 IV: 14999678 (= 49999us) TIV 49999us

ID: 1 IV: 14999573 (= 49999us) TIV 49999us

ID: 0 IV: 14999671 (= 49999us) TIV 49999us

ID: 1 IV: 14999570 (= 49999us) TIV 49999us

ID: 0 IV: 14999588 (= 49999us) TIV 49999us

ID: 1 IV: 14999585 (= 49999us) TIV 49999us

ID: 0 IV: 14999673 (= 49999us) TIV 49999us

ID: 1 IV: 14999577 (= 49999us) TIV 49999us

ID: 0 IV: 14999688 (= 49999us) TIV 49999us

ID: 1 IV: 14999587 (= 49999us) TIV 49999us

ID: 0 IV: 14999481 (= 49998us) TIV 49998us

ID: 1 IV: 14999587 (= 49999us) TIV 49999us

ID: 0 IV: 14999638 (= 49999us) TIV 49999us

ID: 1 IV: 14999567 (= 49999us) TIV 49999us

ID: 0 IV: 14999615 (= 49999us) TIV 49999us

ID: 1 IV: 14999681 (= 49999us) TIV 49999us

ID: 0 IV: 14999675 (= 49999us) TIV 49999us

ID: 1 IV: 14999667 (= 49999us) TIV 49999us

ID: 0 IV: 14999601 (= 49999us) TIV 49999us

ID: 1 IV: 14999557 (= 49999us) TIV 49999us

ID: 0 IV: 14999577 (= 49999us) TIV 49999us

ID: 1 IV: 14999568 (= 49999us) TIV 49999us

ID: 0 IV: 14999486 (= 49998us) TIV 49998us

ID: 1 IV: 14999585 (= 49999us) TIV 49999us

ID: 0 IV: 14999604 (= 49999us) TIV 49999us

ID: 1 IV: 14999595 (= 49999us) TIV 49999us

ID: 0 IV: 14999682 (= 49999us) TIV 49999us

ID: 1 IV: 14999675 (= 49999us) TIV 49999us

ID: 0 IV: 14999590 (= 49999us) TIV 49999us

ID: 1 IV: 14999582 (= 49999us) TIV 49999us

ID: 0 IV: 14999563 (= 49999us) TIV 49999us

ID: 1 IV: 14999577 (= 49999us) TIV 49999us

ID: 0 IV: 14999678 (= 49999us) TIV 49999us

ID: 1 IV: 14999568 (= 49999us) TIV 49999us

ID: 0 IV: 14999588 (= 49999us) TIV 49999us

ID: 1 IV: 14999594 (= 49999us) TIV 49999us

ID: 0 IV: 14999568 (= 49999us) TIV 49999us

ID: 1 IV: 14999584 (= 49999us) TIV 49999us

ID: 0 IV: 14999598 (= 49999us) TIV 49999us

ID: 1 IV: 14999563 (= 49999us) TIV 49999us

ID: 0 IV: 14999680 (= 49999us) TIV 49999us

ID: 1 IV: 14999674 (= 49999us) TIV 49999us

ID: 0 IV: 14999477 (= 49998us) TIV 49998us

ID: 1 IV: 14999570 (= 49999us) TIV 49999us

ID: 0 IV: 14999494 (= 49998us) TIV 49998us

ID: 1 IV: 14999673 (= 49999us) TIV 49999us

ID: 0 IV: 14999593 (= 49999us) TIV 49999us

ID: 1 IV: 14999572 (= 49999us) TIV 49999us

ID: 0 IV: 14999499 (= 49998us) TIV 49998us

ID: 1 IV: 14999568 (= 49999us) TIV 49999us

ID: 0 IV: 14999587 (= 49999us) TIV 49999us

ID: 1 IV: 14999671 (= 49999us) TIV 49999us

ID: 0 IV: 14999569 (= 49999us) TIV 49999us

ID: 1 IV: 14999574 (= 49999us) TIV 49999us

ID: 0 IV: 14999603 (= 49999us) TIV 49999us

ID: 1 IV: 14999497 (= 49998us) TIV 49998us

ID: 0 IV: 14999674 (= 49999us) TIV 49999us

ID: 1 IV: 14999576 (= 49999us) TIV 49999us

ID: 0 IV: 14999689 (= 49999us) TIV 49999us

ID: 1 IV: 14999573 (= 49999us) TIV 49999us

ID: 0 IV: 14999674 (= 49999us) TIV 49999us

ID: 1 IV: 14999584 (= 49999us) TIV 49999us

ID: 0 IV: 14999575 (= 49999us) TIV 49999us

ID: 1 IV: 14999679 (= 49999us) TIV 49999us

ID: 0 IV: 14999652 (= 49999us) TIV 49999us

ID: 1 IV: 14999599 (= 49999us) TIV 49999us

ID: 0 IV: 14999675 (= 49999us) TIV 49999us

ID: 1 IV: 14999573 (= 49999us) TIV 49999us

ID: 0 IV: 14999589 (= 49999us) TIV 49999us

ID: 1 IV: 14999571 (= 49999us) TIV 49999us

ID: 0 IV: 14999573 (= 49999us) TIV 49999us

ID: 1 IV: 14999560 (= 49999us) TIV 49999us

ID: 0 IV: 14999540 (= 49998us) TIV 49998us

ID: 1 IV: 14999661 (= 49999us) TIV 49999us

ID: 0 IV: 14999594 (= 49999us) TIV 49999us

ID: 1 IV: 14999686 (= 49999us) TIV 49999us

ID: 0 IV: 14999586 (= 49999us) TIV 49999us

ID: 1 IV: 14999569 (= 49999us) TIV 49999us

ID: 0 IV: 14999604 (= 49999us) TIV 49999us

ID: 1 IV: 14999664 (= 49999us) TIV 49999us

ID: 0 IV: 14999481 (= 49998us) TIV 49998us

ID: 1 IV: 14999677 (= 49999us) TIV 49999us

ID: 0 IV: 14999576 (= 49999us) TIV 49999us

ID: 1 IV: 14999574 (= 49999us) TIV 49999us

ID: 0 IV: 14999617 (= 49999us) TIV 49999us

RANGE 49996-49999us (3us)

RTEST: Level 005

Next test - expecting half

ID: 0 IV: 7498010 (= 24993us) TIV 0us

ID: 1 IV: 7498904 (= 24996us) TIV 24996us

ID: 0 IV: 7499432 (= 24998us) TIV 24998us

ID: 1 IV: 7499221 (= 24997us) TIV 24997us

ID: 0 IV: 7499512 (= 24998us) TIV 24998us

ID: 1 IV: 7499494 (= 24998us) TIV 24998us

ID: 0 IV: 7499698 (= 24999us) TIV 24999us

ID: 1 IV: 7499604 (= 24999us) TIV 24999us

ID: 0 IV: 7499499 (= 24998us) TIV 24998us

ID: 1 IV: 7499492 (= 24998us) TIV 24998us

ID: 0 IV: 7499614 (= 24999us) TIV 24999us

ID: 1 IV: 7499605 (= 24999us) TIV 24999us

ID: 0 IV: 7499514 (= 24998us) TIV 24998us

ID: 1 IV: 7499600 (= 24999us) TIV 24999us

ID: 0 IV: 7499603 (= 24999us) TIV 24999us

ID: 1 IV: 7499594 (= 24999us) TIV 24999us

ID: 0 IV: 7499600 (= 24999us) TIV 24999us

ID: 1 IV: 7499597 (= 24999us) TIV 24999us

ID: 0 IV: 7499593 (= 24999us) TIV 24999us

ID: 1 IV: 7499632 (= 24999us) TIV 24999us

ID: 0 IV: 7499663 (= 24999us) TIV 24999us

ID: 1 IV: 7499603 (= 24999us) TIV 24999us

ID: 0 IV: 7499501 (= 24998us) TIV 24998us

ID: 1 IV: 7499608 (= 24999us) TIV 24999us

ID: 0 IV: 7499613 (= 24999us) TIV 24999us

ID: 1 IV: 7499410 (= 24998us) TIV 24998us

ID: 0 IV: 7499596 (= 24999us) TIV 24999us

ID: 1 IV: 7499602 (= 24999us) TIV 24999us

ID: 0 IV: 7499501 (= 24998us) TIV 24998us

ID: 1 IV: 7499489 (= 24998us) TIV 24998us

ID: 0 IV: 7499615 (= 24999us) TIV 24999us

ID: 1 IV: 7499490 (= 24998us) TIV 24998us

ID: 0 IV: 7499506 (= 24998us) TIV 24998us

ID: 1 IV: 7499608 (= 24999us) TIV 24999us

ID: 0 IV: 7499596 (= 24999us) TIV 24999us

ID: 1 IV: 7499602 (= 24999us) TIV 24999us

ID: 0 IV: 7499602 (= 24999us) TIV 24999us

ID: 1 IV: 7499589 (= 24999us) TIV 24999us

ID: 0 IV: 7499407 (= 24998us) TIV 24998us

ID: 1 IV: 7499617 (= 24999us) TIV 24999us

ID: 0 IV: 7499599 (= 24999us) TIV 24999us

ID: 1 IV: 7499608 (= 24999us) TIV 24999us

ID: 0 IV: 7499606 (= 24999us) TIV 24999us

ID: 1 IV: 7499602 (= 24999us) TIV 24999us

ID: 0 IV: 7499595 (= 24999us) TIV 24999us

ID: 1 IV: 7499395 (= 24998us) TIV 24998us

ID: 0 IV: 7499610 (= 24999us) TIV 24999us

ID: 1 IV: 7499604 (= 24999us) TIV 24999us

ID: 0 IV: 7499602 (= 24999us) TIV 24999us

ID: 1 IV: 7499592 (= 24999us) TIV 24999us

ID: 0 IV: 7499513 (= 24998us) TIV 24998us

ID: 1 IV: 7499498 (= 24998us) TIV 24998us

ID: 0 IV: 7499592 (= 24999us) TIV 24999us

ID: 1 IV: 7499597 (= 24999us) TIV 24999us

ID: 0 IV: 7499605 (= 24999us) TIV 24999us

ID: 1 IV: 7499498 (= 24998us) TIV 24998us

ID: 0 IV: 7499704 (= 24999us) TIV 24999us

ID: 1 IV: 7499591 (= 24999us) TIV 24999us

ID: 0 IV: 7499602 (= 24999us) TIV 24999us

ID: 1 IV: 7499607 (= 24999us) TIV 24999us

ID: 0 IV: 7499607 (= 24999us) TIV 24999us

ID: 1 IV: 7499598 (= 24999us) TIV 24999us

ID: 0 IV: 7499396 (= 24998us) TIV 24998us

ID: 1 IV: 7499608 (= 24999us) TIV 24999us

ID: 0 IV: 7499707 (= 24999us) TIV 24999us

ID: 1 IV: 7499504 (= 24998us) TIV 24998us

ID: 0 IV: 7499395 (= 24998us) TIV 24998us

ID: 1 IV: 7499599 (= 24999us) TIV 24999us

ID: 0 IV: 7499405 (= 24998us) TIV 24998us

ID: 1 IV: 7499606 (= 24999us) TIV 24999us

ID: 0 IV: 7499613 (= 24999us) TIV 24999us

ID: 1 IV: 7499601 (= 24999us) TIV 24999us

ID: 0 IV: 7499408 (= 24998us) TIV 24998us

ID: 1 IV: 7499595 (= 24999us) TIV 24999us

ID: 0 IV: 7499595 (= 24999us) TIV 24999us

ID: 1 IV: 7499601 (= 24999us) TIV 24999us

ID: 0 IV: 7499600 (= 24999us) TIV 24999us

ID: 1 IV: 7499705 (= 24999us) TIV 24999us

ID: 0 IV: 7499710 (= 24999us) TIV 24999us

ID: 1 IV: 7499606 (= 24999us) TIV 24999us

ID: 0 IV: 7499542 (= 24998us) TIV 24998us

ID: 1 IV: 7499612 (= 24999us) TIV 24999us

ID: 0 IV: 7499600 (= 24999us) TIV 24999us

ID: 1 IV: 7499591 (= 24999us) TIV 24999us

ID: 0 IV: 7499599 (= 24999us) TIV 24999us

ID: 1 IV: 7499509 (= 24998us) TIV 24998us

ID: 0 IV: 7499600 (= 24999us) TIV 24999us

ID: 1 IV: 7499596 (= 24999us) TIV 24999us

ID: 0 IV: 7499702 (= 24999us) TIV 24999us

ID: 1 IV: 7499614 (= 24999us) TIV 24999us

ID: 0 IV: 7499611 (= 24999us) TIV 24999us

ID: 1 IV: 7499492 (= 24998us) TIV 24998us

ID: 0 IV: 7499609 (= 24999us) TIV 24999us

ID: 1 IV: 7499598 (= 24999us) TIV 24999us

ID: 0 IV: 7499609 (= 24999us) TIV 24999us

ID: 1 IV: 7499604 (= 24999us) TIV 24999us

ID: 0 IV: 7499395 (= 24998us) TIV 24998us

ID: 1 IV: 7499389 (= 24998us) TIV 24998us

ID: 0 IV: 7499503 (= 24998us) TIV 24998us

ID: 1 IV: 7499702 (= 24999us) TIV 24999us

RANGE 24996-24999us (3us)

RTEST: Level 006

Next test - expecting normal again

ID: 0 IV: 14998134 (= 49994us) TIV 0us

ID: 1 IV: 14998988 (= 49997us) TIV 49997us

ID: 0 IV: 14999501 (= 49998us) TIV 49998us

ID: 1 IV: 14999409 (= 49998us) TIV 49998us

ID: 0 IV: 14999573 (= 49999us) TIV 49999us

ID: 1 IV: 14999608 (= 49999us) TIV 49999us

ID: 0 IV: 14999573 (= 49999us) TIV 49999us

ID: 1 IV: 14999591 (= 49999us) TIV 49999us

ID: 0 IV: 14999575 (= 49999us) TIV 49999us

ID: 1 IV: 14999590 (= 49999us) TIV 49999us

ID: 0 IV: 14999601 (= 49999us) TIV 49999us

ID: 1 IV: 14999618 (= 49999us) TIV 49999us

ID: 0 IV: 14999676 (= 49999us) TIV 49999us

ID: 1 IV: 14999581 (= 49999us) TIV 49999us

ID: 0 IV: 14999584 (= 49999us) TIV 49999us

ID: 1 IV: 14999669 (= 49999us) TIV 49999us

ID: 0 IV: 14999589 (= 49999us) TIV 49999us

ID: 1 IV: 14999579 (= 49999us) TIV 49999us

ID: 0 IV: 14999643 (= 49999us) TIV 49999us

ID: 1 IV: 14999678 (= 49999us) TIV 49999us

ID: 0 IV: 14999593 (= 49999us) TIV 49999us

ID: 1 IV: 14999474 (= 49998us) TIV 49998us

ID: 0 IV: 14999592 (= 49999us) TIV 49999us

ID: 1 IV: 14999502 (= 49998us) TIV 49998us

ID: 0 IV: 14999580 (= 49999us) TIV 49999us

ID: 1 IV: 14999575 (= 49999us) TIV 49999us

ID: 0 IV: 14999594 (= 49999us) TIV 49999us

ID: 1 IV: 14999567 (= 49999us) TIV 49999us

ID: 0 IV: 14999586 (= 49999us) TIV 49999us

ID: 1 IV: 14999591 (= 49999us) TIV 49999us

ID: 0 IV: 14999632 (= 49999us) TIV 49999us

ID: 1 IV: 14999697 (= 49999us) TIV 49999us

ID: 0 IV: 14999579 (= 49999us) TIV 49999us

ID: 1 IV: 14999684 (= 49999us) TIV 49999us

ID: 0 IV: 14999680 (= 49999us) TIV 49999us

ID: 1 IV: 14999680 (= 49999us) TIV 49999us

ID: 0 IV: 14999580 (= 49999us) TIV 49999us

ID: 1 IV: 14999580 (= 49999us) TIV 49999us

ID: 0 IV: 14999668 (= 49999us) TIV 49999us

ID: 1 IV: 14999676 (= 49999us) TIV 49999us

ID: 0 IV: 14999608 (= 49999us) TIV 49999us

ID: 1 IV: 14999605 (= 49999us) TIV 49999us

ID: 0 IV: 14999573 (= 49999us) TIV 49999us

ID: 1 IV: 14999682 (= 49999us) TIV 49999us

ID: 0 IV: 14999589 (= 49999us) TIV 49999us

ID: 1 IV: 14999691 (= 49999us) TIV 49999us

ID: 0 IV: 14999586 (= 49999us) TIV 49999us

ID: 1 IV: 14999680 (= 49999us) TIV 49999us

ID: 0 IV: 14999592 (= 49999us) TIV 49999us

ID: 1 IV: 14999565 (= 49999us) TIV 49999us

ID: 0 IV: 14999678 (= 49999us) TIV 49999us

ID: 1 IV: 14999508 (= 49998us) TIV 49998us

ID: 0 IV: 14999671 (= 49999us) TIV 49999us

ID: 1 IV: 14999566 (= 49999us) TIV 49999us

ID: 0 IV: 14999590 (= 49999us) TIV 49999us

ID: 1 IV: 14999595 (= 49999us) TIV 49999us

ID: 0 IV: 14999594 (= 49999us) TIV 49999us

ID: 1 IV: 14999564 (= 49999us) TIV 49999us

ID: 0 IV: 14999676 (= 49999us) TIV 49999us

ID: 1 IV: 14999603 (= 49999us) TIV 49999us

ID: 0 IV: 14999676 (= 49999us) TIV 49999us

ID: 1 IV: 14999618 (= 49999us) TIV 49999us

ID: 0 IV: 14999562 (= 49999us) TIV 49999us

ID: 1 IV: 14999590 (= 49999us) TIV 49999us

ID: 0 IV: 14999693 (= 49999us) TIV 49999us

ID: 1 IV: 14999594 (= 49999us) TIV 49999us

ID: 0 IV: 14999661 (= 49999us) TIV 49999us

ID: 1 IV: 14999683 (= 49999us) TIV 49999us

ID: 0 IV: 14999569 (= 49999us) TIV 49999us

ID: 1 IV: 14999592 (= 49999us) TIV 49999us

ID: 0 IV: 14999678 (= 49999us) TIV 49999us

ID: 1 IV: 14999612 (= 49999us) TIV 49999us

ID: 0 IV: 14999572 (= 49999us) TIV 49999us

ID: 1 IV: 14999670 (= 49999us) TIV 49999us

ID: 0 IV: 14999591 (= 49999us) TIV 49999us

ID: 1 IV: 14999694 (= 49999us) TIV 49999us

ID: 0 IV: 14999479 (= 49998us) TIV 49998us

ID: 1 IV: 14999673 (= 49999us) TIV 49999us

ID: 0 IV: 14999575 (= 49999us) TIV 49999us

ID: 1 IV: 14999616 (= 49999us) TIV 49999us

ID: 0 IV: 14999680 (= 49999us) TIV 49999us

ID: 1 IV: 14999599 (= 49999us) TIV 49999us

ID: 0 IV: 14999591 (= 49999us) TIV 49999us

ID: 1 IV: 14999571 (= 49999us) TIV 49999us

ID: 0 IV: 14999669 (= 49999us) TIV 49999us

ID: 1 IV: 14999491 (= 49998us) TIV 49998us

ID: 0 IV: 14999576 (= 49999us) TIV 49999us

ID: 1 IV: 14999572 (= 49999us) TIV 49999us

ID: 0 IV: 14999581 (= 49999us) TIV 49999us

ID: 1 IV: 14999576 (= 49999us) TIV 49999us

ID: 0 IV: 14999569 (= 49999us) TIV 49999us

ID: 1 IV: 14999544 (= 49998us) TIV 49998us

ID: 0 IV: 14999580 (= 49999us) TIV 49999us

ID: 1 IV: 14999575 (= 49999us) TIV 49999us

ID: 0 IV: 14999575 (= 49999us) TIV 49999us

ID: 1 IV: 14999585 (= 49999us) TIV 49999us

ID: 0 IV: 14999572 (= 49999us) TIV 49999us

ID: 1 IV: 14999585 (= 49999us) TIV 49999us

ID: 0 IV: 14999681 (= 49999us) TIV 49999us

ID: 1 IV: 14999596 (= 49999us) TIV 49999us

RANGE 49997-49999us (2us)

RTEST: Level 007

Next test - expecting random

ID: 1 IV: 74998262 (= 249994us) TIV 0us

ID: 0 IV: 74999070 (= 249997us) TIV 249997us

ID: 1 IV: 74998893 (= 249996us) TIV 249996us

ID: 0 IV: 74999389 (= 249998us) TIV 249998us

ID: 1 IV: 14998832 (= 49996us) TIV 49996us

ID: 0 IV: 14999569 (= 49999us) TIV 49999us

ID: 1 IV: 14998967 (= 49997us) TIV 49997us

ID: 0 IV: 14999634 (= 49999us) TIV 49999us

ID: 1 IV: 29999085 (= 99997us) TIV 99997us

ID: 0 IV: 29999634 (= 99999us) TIV 99999us

ID: 1 IV: 29998974 (= 99997us) TIV 99997us

ID: 0 IV: 29999559 (= 99999us) TIV 99999us

ID: 1 IV: 44998954 (= 149997us) TIV 149997us

ID: 0 IV: 44999488 (= 149998us) TIV 149998us

ID: 1 IV: 44998887 (= 149996us) TIV 149996us

ID: 0 IV: 44999526 (= 149998us) TIV 149998us

ID: 1 IV: 59998967 (= 199997us) TIV 199997us

ID: 0 IV: 59999566 (= 199999us) TIV 199999us

ID: 1 IV: 59999002 (= 199997us) TIV 199997us

ID: 0 IV: 59999557 (= 199999us) TIV 199999us

ID: 1 IV: 74998936 (= 249996us) TIV 249996us

ID: 0 IV: 74999494 (= 249998us) TIV 249998us

ID: 1 IV: 74998998 (= 249997us) TIV 249997us

ID: 0 IV: 74999515 (= 249998us) TIV 249998us

ID: 1 IV: 14998901 (= 49996us) TIV 49996us

ID: 0 IV: 14999518 (= 49998us) TIV 49998us

ID: 1 IV: 14999014 (= 49997us) TIV 49997us

ID: 0 IV: 14999587 (= 49999us) TIV 49999us

ID: 1 IV: 29998801 (= 99996us) TIV 99996us

ID: 0 IV: 29999540 (= 99998us) TIV 99998us

ID: 1 IV: 29998912 (= 99996us) TIV 99996us

ID: 0 IV: 29999656 (= 99999us) TIV 99999us

ID: 1 IV: 44998886 (= 149996us) TIV 149996us

ID: 0 IV: 44999600 (= 149999us) TIV 149999us

ID: 1 IV: 44998920 (= 149996us) TIV 149996us

ID: 0 IV: 44999644 (= 149999us) TIV 149999us

ID: 1 IV: 59999152 (= 199997us) TIV 199997us

ID: 0 IV: 59999536 (= 199998us) TIV 199998us

ID: 1 IV: 59999076 (= 199997us) TIV 199997us

ID: 0 IV: 59999565 (= 199999us) TIV 199999us

ID: 1 IV: 74998976 (= 249997us) TIV 249997us

ID: 0 IV: 74999606 (= 249999us) TIV 249999us

ID: 1 IV: 74998954 (= 249997us) TIV 249997us

ID: 0 IV: 74999617 (= 249999us) TIV 249999us

ID: 1 IV: 14999029 (= 49997us) TIV 49997us

ID: 0 IV: 14999616 (= 49999us) TIV 49999us

ID: 1 IV: 14998914 (= 49996us) TIV 49996us

ID: 0 IV: 14999508 (= 49998us) TIV 49998us

ID: 1 IV: 29998866 (= 99996us) TIV 99996us

ID: 0 IV: 29999642 (= 99999us) TIV 99999us

ID: 1 IV: 29999008 (= 99997us) TIV 99997us

ID: 0 IV: 29999566 (= 99999us) TIV 99999us

ID: 1 IV: 44999008 (= 149997us) TIV 149997us

ID: 0 IV: 44999590 (= 149999us) TIV 149999us

ID: 1 IV: 44998981 (= 149997us) TIV 149997us

ID: 0 IV: 44999611 (= 149999us) TIV 149999us

ID: 1 IV: 59998979 (= 199997us) TIV 199997us

ID: 0 IV: 59999568 (= 199999us) TIV 199999us

ID: 1 IV: 59998909 (= 199996us) TIV 199996us

ID: 0 IV: 59999541 (= 199998us) TIV 199998us

ID: 1 IV: 74998843 (= 249996us) TIV 249996us

ID: 0 IV: 74999507 (= 249998us) TIV 249998us

ID: 1 IV: 74998953 (= 249997us) TIV 249997us

ID: 0 IV: 74999577 (= 249999us) TIV 249999us

ID: 1 IV: 14999129 (= 49997us) TIV 49997us

ID: 0 IV: 14999535 (= 49998us) TIV 49998us

ID: 1 IV: 14998930 (= 49996us) TIV 49996us

ID: 0 IV: 14999624 (= 49999us) TIV 49999us

ID: 1 IV: 29998957 (= 99997us) TIV 99997us

ID: 0 IV: 29999594 (= 99999us) TIV 99999us

ID: 1 IV: 29998978 (= 99997us) TIV 99997us

ID: 0 IV: 29999544 (= 99998us) TIV 99998us

ID: 1 IV: 44998890 (= 149996us) TIV 149996us

ID: 0 IV: 44999516 (= 149998us) TIV 149998us

ID: 1 IV: 44999005 (= 149997us) TIV 149997us

ID: 0 IV: 44999447 (= 149998us) TIV 149998us

ID: 1 IV: 59999004 (= 199997us) TIV 199997us

ID: 0 IV: 59999607 (= 199999us) TIV 199999us

ID: 1 IV: 59998920 (= 199996us) TIV 199996us

ID: 0 IV: 59999542 (= 199998us) TIV 199998us

ID: 1 IV: 74999042 (= 249997us) TIV 249997us

ID: 0 IV: 74999602 (= 249999us) TIV 249999us

ID: 1 IV: 74999083 (= 249997us) TIV 249997us

ID: 0 IV: 74999537 (= 249998us) TIV 249998us

ID: 1 IV: 14999040 (= 49997us) TIV 49997us

ID: 0 IV: 14999501 (= 49998us) TIV 49998us

ID: 1 IV: 14999080 (= 49997us) TIV 49997us

ID: 0 IV: 14999530 (= 49998us) TIV 49998us

ID: 1 IV: 29999106 (= 99997us) TIV 99997us

ID: 0 IV: 29999545 (= 99998us) TIV 99998us

ID: 1 IV: 29999183 (= 99997us) TIV 99997us

ID: 0 IV: 29999637 (= 99999us) TIV 99999us

ID: 1 IV: 44999008 (= 149997us) TIV 149997us

ID: 0 IV: 44999613 (= 149999us) TIV 149999us

ID: 1 IV: 44999081 (= 149997us) TIV 149997us

ID: 0 IV: 44999638 (= 149999us) TIV 149999us

ID: 1 IV: 59998934 (= 199996us) TIV 199996us

ID: 0 IV: 59999579 (= 199999us) TIV 199999us

ID: 1 IV: 59998973 (= 199997us) TIV 199997us

ID: 0 IV: 59999616 (= 199999us) TIV 199999us

ID: 1 IV: 74999045 (= 249997us) TIV 249997us

ID: 0 IV: 74999529 (= 249998us) TIV 249998us

ID: 1 IV: 74999017 (= 249997us) TIV 249997us

ID: 0 IV: 74999611 (= 249999us) TIV 249999us

ID: 1 IV: 14999029 (= 49997us) TIV 49997us

ID: 0 IV: 14999601 (= 49999us) TIV 49999us

ID: 1 IV: 14998920 (= 49996us) TIV 49996us

ID: 0 IV: 14999527 (= 49998us) TIV 49998us

ID: 1 IV: 29998903 (= 99996us) TIV 99996us

ID: 0 IV: 29999546 (= 99998us) TIV 99998us

ID: 1 IV: 29998873 (= 99996us) TIV 99996us

ID: 0 IV: 29999554 (= 99999us) TIV 99999us

ID: 1 IV: 44998869 (= 149996us) TIV 149996us

ID: 0 IV: 44999616 (= 149999us) TIV 149999us

ID: 1 IV: 44998947 (= 149996us) TIV 149996us

ID: 0 IV: 44999570 (= 149999us) TIV 149999us

ID: 1 IV: 59999081 (= 199997us) TIV 199997us

ID: 0 IV: 59999593 (= 199999us) TIV 199999us

ID: 1 IV: 59999029 (= 199997us) TIV 199997us

ID: 0 IV: 59999480 (= 199998us) TIV 199998us

ID: 1 IV: 74998971 (= 249997us) TIV 249997us

ID: 0 IV: 74999477 (= 249998us) TIV 249998us

ID: 1 IV: 74999095 (= 249997us) TIV 249997us

ID: 0 IV: 74999612 (= 249999us) TIV 249999us

ID: 1 IV: 14999008 (= 49997us) TIV 49997us

ID: 0 IV: 14999573 (= 49999us) TIV 49999us

ID: 1 IV: 14998898 (= 49996us) TIV 49996us

ID: 0 IV: 14999528 (= 49998us) TIV 49998us

ID: 1 IV: 29998901 (= 99996us) TIV 99996us

ID: 0 IV: 29999558 (= 99999us) TIV 99999us

ID: 1 IV: 29998953 (= 99997us) TIV 99997us

ID: 0 IV: 29999521 (= 99998us) TIV 99998us

ID: 1 IV: 44998986 (= 149997us) TIV 149997us

ID: 0 IV: 44999636 (= 149999us) TIV 149999us

ID: 1 IV: 44999149 (= 149997us) TIV 149997us

ID: 0 IV: 44999614 (= 149999us) TIV 149999us

ID: 1 IV: 59999134 (= 199997us) TIV 199997us

ID: 0 IV: 59999604 (= 199999us) TIV 199999us

ID: 1 IV: 59998997 (= 199997us) TIV 199997us

ID: 0 IV: 59999577 (= 199999us) TIV 199999us

ID: 1 IV: 74998924 (= 249996us) TIV 249996us

ID: 0 IV: 74999522 (= 249998us) TIV 249998us

ID: 1 IV: 74998965 (= 249997us) TIV 249997us

ID: 0 IV: 74999589 (= 249999us) TIV 249999us

ID: 1 IV: 14999053 (= 49997us) TIV 49997us

ID: 0 IV: 14999500 (= 49998us) TIV 49998us

ID: 1 IV: 14998909 (= 49996us) TIV 49996us

ID: 0 IV: 14999629 (= 49999us) TIV 49999us

ID: 1 IV: 29998939 (= 99996us) TIV 99996us

ID: 0 IV: 29999544 (= 99998us) TIV 99998us

ID: 1 IV: 29998954 (= 99997us) TIV 99997us

ID: 0 IV: 29999651 (= 99999us) TIV 99999us

ID: 1 IV: 44998960 (= 149997us) TIV 149997us

ID: 0 IV: 44999609 (= 149999us) TIV 149999us

ID: 1 IV: 44999081 (= 149997us) TIV 149997us

ID: 0 IV: 44999609 (= 149999us) TIV 149999us

ID: 1 IV: 59998928 (= 199996us) TIV 199996us

ID: 0 IV: 59999500 (= 199998us) TIV 199998us

ID: 1 IV: 59998878 (= 199996us) TIV 199996us

ID: 0 IV: 59999556 (= 199999us) TIV 199999us

ID: 1 IV: 74999047 (= 249997us) TIV 249997us

ID: 0 IV: 74999620 (= 249999us) TIV 249999us

ID: 1 IV: 74999068 (= 249997us) TIV 249997us

ID: 0 IV: 74999495 (= 249998us) TIV 249998us

ID: 1 IV: 14998930 (= 49996us) TIV 49996us

ID: 0 IV: 14999575 (= 49999us) TIV 49999us

ID: 1 IV: 14998907 (= 49996us) TIV 49996us

ID: 0 IV: 14999538 (= 49998us) TIV 49998us

ID: 1 IV: 29999013 (= 99997us) TIV 99997us

ID: 0 IV: 29999586 (= 99999us) TIV 99999us

ID: 1 IV: 29998951 (= 99997us) TIV 99997us

ID: 0 IV: 29999542 (= 99998us) TIV 99998us

ID: 1 IV: 44998864 (= 149996us) TIV 149996us

ID: 0 IV: 44999589 (= 149999us) TIV 149999us

ID: 1 IV: 44999034 (= 149997us) TIV 149997us

ID: 0 IV: 44999538 (= 149998us) TIV 149998us

ID: 1 IV: 59998936 (= 199996us) TIV 199996us

ID: 0 IV: 59999511 (= 199998us) TIV 199998us

ID: 1 IV: 59998871 (= 199996us) TIV 199996us

ID: 0 IV: 59999598 (= 199999us) TIV 199999us

ID: 1 IV: 74998937 (= 249996us) TIV 249996us

ID: 0 IV: 74999611 (= 249999us) TIV 249999us

ID: 1 IV: 74998926 (= 249996us) TIV 249996us

ID: 0 IV: 74999501 (= 249998us) TIV 249998us

ID: 1 IV: 14999102 (= 49997us) TIV 49997us

ID: 0 IV: 14999526 (= 49998us) TIV 49998us

ID: 1 IV: 14998899 (= 49996us) TIV 49996us

ID: 0 IV: 14999505 (= 49998us) TIV 49998us

ID: 1 IV: 29999019 (= 99997us) TIV 99997us

ID: 0 IV: 29999569 (= 99999us) TIV 99999us

ID: 1 IV: 29998865 (= 99996us) TIV 99996us

ID: 0 IV: 29999543 (= 99998us) TIV 99998us

ID: 1 IV: 44998888 (= 149996us) TIV 149996us

ID: 0 IV: 44999565 (= 149999us) TIV 149999us

ID: 1 IV: 44999035 (= 149997us) TIV 149997us

ID: 0 IV: 44999535 (= 149998us) TIV 149998us

ID: 1 IV: 59999091 (= 199997us) TIV 199997us

ID: 0 IV: 59999516 (= 199998us) TIV 199998us

ID: 1 IV: 59999093 (= 199997us) TIV 199997us

ID: 0 IV: 59999583 (= 199999us) TIV 199999us

RANGE 49996-249999us (200003us)

RTEST: Level 008

Next test - expecting normal again

ID: 1 IV: 14998294 (= 49994us) TIV 0us

ID: 0 IV: 14998836 (= 49996us) TIV 49996us

ID: 1 IV: 14998889 (= 49996us) TIV 49996us

ID: 0 IV: 14999494 (= 49998us) TIV 49998us

ID: 1 IV: 14966315 (= 49888us) TIV 49888us

ID: 0 IV: 14999550 (= 49999us) TIV 49999us

ID: 1 IV: 14998768 (= 49996us) TIV 49996us

ID: 0 IV: 14999518 (= 49998us) TIV 49998us

ID: 1 IV: 15004623 (= 50015us) TIV 50015us

ID: 0 IV: 14999613 (= 49999us) TIV 49999us

ID: 1 IV: 14999123 (= 49997us) TIV 49997us

ID: 0 IV: 14999591 (= 49999us) TIV 49999us

ID: 1 IV: 15001675 (= 50006us) TIV 50006us

ID: 0 IV: 14999598 (= 49999us) TIV 49999us

ID: 1 IV: 14998882 (= 49996us) TIV 49996us

ID: 0 IV: 14999542 (= 49998us) TIV 49998us

ID: 1 IV: 15002230 (= 50007us) TIV 50007us

ID: 0 IV: 14999602 (= 49999us) TIV 49999us

ID: 1 IV: 14998883 (= 49996us) TIV 49996us

ID: 0 IV: 14999487 (= 49998us) TIV 49998us

ID: 1 IV: 15000685 (= 50002us) TIV 50002us

ID: 0 IV: 14999620 (= 49999us) TIV 49999us

ID: 1 IV: 14998901 (= 49996us) TIV 49996us

ID: 0 IV: 14999646 (= 49999us) TIV 49999us

ID: 1 IV: 14971719 (= 49906us) TIV 49906us

ID: 0 IV: 14999640 (= 49999us) TIV 49999us

ID: 1 IV: 14999077 (= 49997us) TIV 49997us

ID: 0 IV: 14999497 (= 49998us) TIV 49998us

ID: 1 IV: 15003440 (= 50011us) TIV 50011us

ID: 0 IV: 14999492 (= 49998us) TIV 49998us

ID: 1 IV: 14998887 (= 49996us) TIV 49996us

ID: 0 IV: 14999557 (= 49999us) TIV 49999us

ID: 1 IV: 15001830 (= 50006us) TIV 50006us

ID: 0 IV: 14999576 (= 49999us) TIV 49999us

ID: 1 IV: 14998897 (= 49996us) TIV 49996us

ID: 0 IV: 14999525 (= 49998us) TIV 49998us

ID: 1 IV: 15002008 (= 50007us) TIV 50007us

ID: 0 IV: 14999604 (= 49999us) TIV 49999us

ID: 1 IV: 14998956 (= 49997us) TIV 49997us

ID: 0 IV: 14999522 (= 49998us) TIV 49998us

ID: 1 IV: 15000767 (= 50003us) TIV 50003us

ID: 0 IV: 14999588 (= 49999us) TIV 49999us

ID: 1 IV: 14998982 (= 49997us) TIV 49997us

ID: 0 IV: 14999601 (= 49999us) TIV 49999us

ID: 1 IV: 14968884 (= 49896us) TIV 49896us

ID: 0 IV: 14999578 (= 49999us) TIV 49999us

ID: 1 IV: 14999028 (= 49997us) TIV 49997us

ID: 0 IV: 14999596 (= 49999us) TIV 49999us

ID: 1 IV: 15003451 (= 50012us) TIV 50012us

ID: 0 IV: 14999584 (= 49999us) TIV 49999us

ID: 1 IV: 14999070 (= 49997us) TIV 49997us

ID: 0 IV: 14999532 (= 49998us) TIV 49998us

ID: 1 IV: 15001859 (= 50006us) TIV 50006us

ID: 0 IV: 14999588 (= 49999us) TIV 49999us

ID: 1 IV: 14998891 (= 49996us) TIV 49996us

ID: 0 IV: 14999543 (= 49998us) TIV 49998us

ID: 1 IV: 15001323 (= 50004us) TIV 50004us

ID: 0 IV: 14999520 (= 49998us) TIV 49998us

ID: 1 IV: 14999057 (= 49997us) TIV 49997us

ID: 0 IV: 14999583 (= 49999us) TIV 49999us

ID: 1 IV: 15000734 (= 50002us) TIV 50002us

ID: 0 IV: 14999583 (= 49999us) TIV 49999us

ID: 1 IV: 14998954 (= 49997us) TIV 49997us

ID: 0 IV: 14999505 (= 49998us) TIV 49998us

ID: 1 IV: 14970323 (= 49901us) TIV 49901us

ID: 0 IV: 14999617 (= 49999us) TIV 49999us

ID: 1 IV: 14999006 (= 49997us) TIV 49997us

ID: 0 IV: 14999497 (= 49998us) TIV 49998us

ID: 1 IV: 15004798 (= 50016us) TIV 50016us

ID: 0 IV: 14999508 (= 49998us) TIV 49998us

ID: 1 IV: 14999055 (= 49997us) TIV 49997us

ID: 0 IV: 14999493 (= 49998us) TIV 49998us

ID: 1 IV: 15001814 (= 50006us) TIV 50006us

ID: 0 IV: 14999551 (= 49999us) TIV 49999us

ID: 1 IV: 14999016 (= 49997us) TIV 49997us

ID: 0 IV: 14999651 (= 49999us) TIV 49999us

ID: 1 IV: 15002141 (= 50007us) TIV 50007us

ID: 0 IV: 14999583 (= 49999us) TIV 49999us

ID: 1 IV: 14998989 (= 49997us) TIV 49997us

ID: 0 IV: 14999593 (= 49999us) TIV 49999us

ID: 1 IV: 15000861 (= 50003us) TIV 50003us

ID: 0 IV: 14999605 (= 49999us) TIV 49999us

ID: 1 IV: 14999065 (= 49997us) TIV 49997us

ID: 0 IV: 14999516 (= 49998us) TIV 49998us

ID: 1 IV: 14970771 (= 49903us) TIV 49903us

ID: 0 IV: 14999606 (= 49999us) TIV 49999us

ID: 1 IV: 14998967 (= 49997us) TIV 49997us

ID: 0 IV: 14999628 (= 49999us) TIV 49999us

ID: 1 IV: 15004649 (= 50015us) TIV 50015us

ID: 0 IV: 14999580 (= 49999us) TIV 49999us

ID: 1 IV: 14998862 (= 49996us) TIV 49996us

ID: 0 IV: 14999604 (= 49999us) TIV 49999us

ID: 1 IV: 15001520 (= 50005us) TIV 50005us

ID: 0 IV: 14999591 (= 49999us) TIV 49999us

ID: 1 IV: 14998997 (= 49997us) TIV 49997us

ID: 0 IV: 14999587 (= 49999us) TIV 49999us

ID: 1 IV: 15002203 (= 50007us) TIV 50007us

ID: 0 IV: 14999587 (= 49999us) TIV 49999us

ID: 1 IV: 14998868 (= 49996us) TIV 49996us

ID: 0 IV: 14999618 (= 49999us) TIV 49999us

ID: 1 IV: 15000713 (= 50002us) TIV 50002us

ID: 0 IV: 14999544 (= 49998us) TIV 49998us

ID: 1 IV: 14998955 (= 49997us) TIV 49997us

ID: 0 IV: 14999594 (= 49999us) TIV 49999us

ID: 1 IV: 14971897 (= 49906us) TIV 49906us

ID: 0 IV: 14999548 (= 49998us) TIV 49998us

ID: 1 IV: 14999053 (= 49997us) TIV 49997us

ID: 0 IV: 14999500 (= 49998us) TIV 49998us

ID: 1 IV: 15003725 (= 50012us) TIV 50012us

ID: 0 IV: 14999526 (= 49998us) TIV 49998us

ID: 1 IV: 14998896 (= 49996us) TIV 49996us

ID: 0 IV: 14999570 (= 49999us) TIV 49999us

ID: 1 IV: 15001745 (= 50006us) TIV 50006us

ID: 0 IV: 14999516 (= 49998us) TIV 49998us

ID: 1 IV: 14998978 (= 49997us) TIV 49997us

ID: 0 IV: 14999598 (= 49999us) TIV 49999us

ID: 1 IV: 15001391 (= 50005us) TIV 50005us

ID: 0 IV: 14999624 (= 49999us) TIV 49999us

ID: 1 IV: 14998993 (= 49997us) TIV 49997us

ID: 0 IV: 14999644 (= 49999us) TIV 49999us

ID: 1 IV: 15000724 (= 50002us) TIV 50002us

ID: 0 IV: 14999494 (= 49998us) TIV 49998us

ID: 1 IV: 14998951 (= 49997us) TIV 49997us

ID: 0 IV: 14999505 (= 49998us) TIV 49998us

ID: 1 IV: 14970188 (= 49901us) TIV 49901us

ID: 0 IV: 14999495 (= 49998us) TIV 49998us

ID: 1 IV: 14999092 (= 49997us) TIV 49997us

ID: 0 IV: 14999609 (= 49999us) TIV 49999us

ID: 1 IV: 15004387 (= 50015us) TIV 50015us

ID: 0 IV: 14999554 (= 49999us) TIV 49999us

ID: 1 IV: 14998925 (= 49996us) TIV 49996us

ID: 0 IV: 14999499 (= 49998us) TIV 49998us

ID: 1 IV: 15001986 (= 50007us) TIV 50007us

ID: 0 IV: 14999613 (= 49999us) TIV 49999us

ID: 1 IV: 14999134 (= 49997us) TIV 49997us

ID: 0 IV: 14999608 (= 49999us) TIV 49999us

ID: 1 IV: 15001879 (= 50006us) TIV 50006us

ID: 0 IV: 14999491 (= 49998us) TIV 49998us

ID: 1 IV: 14999034 (= 49997us) TIV 49997us

ID: 0 IV: 14999574 (= 49999us) TIV 49999us

ID: 1 IV: 15000858 (= 50003us) TIV 50003us

ID: 0 IV: 14999584 (= 49999us) TIV 49999us

ID: 1 IV: 14998965 (= 49997us) TIV 49997us

ID: 0 IV: 14999619 (= 49999us) TIV 49999us

ID: 1 IV: 14970493 (= 49902us) TIV 49902us

ID: 0 IV: 14999498 (= 49998us) TIV 49998us

ID: 1 IV: 14999020 (= 49997us) TIV 49997us

ID: 0 IV: 14999583 (= 49999us) TIV 49999us

ID: 1 IV: 15003134 (= 50010us) TIV 50010us

ID: 0 IV: 14999591 (= 49999us) TIV 49999us

ID: 1 IV: 14998904 (= 49996us) TIV 49996us

ID: 0 IV: 14999594 (= 49999us) TIV 49999us

ID: 1 IV: 15001539 (= 50005us) TIV 50005us

ID: 0 IV: 14999482 (= 49998us) TIV 49998us

ID: 1 IV: 14999060 (= 49997us) TIV 49997us

ID: 0 IV: 14999504 (= 49998us) TIV 49998us

ID: 1 IV: 15002198 (= 50007us) TIV 50007us

ID: 0 IV: 14999597 (= 49999us) TIV 49999us

ID: 1 IV: 14998951 (= 49997us) TIV 49997us

ID: 0 IV: 14999519 (= 49998us) TIV 49998us

ID: 1 IV: 15000830 (= 50003us) TIV 50003us

ID: 0 IV: 14999580 (= 49999us) TIV 49999us

ID: 1 IV: 14999188 (= 49997us) TIV 49997us

ID: 0 IV: 14999592 (= 49999us) TIV 49999us

ID: 1 IV: 14969502 (= 49898us) TIV 49898us

ID: 0 IV: 14999632 (= 49999us) TIV 49999us

ID: 1 IV: 14998992 (= 49997us) TIV 49997us

ID: 0 IV: 14999613 (= 49999us) TIV 49999us

ID: 1 IV: 15003412 (= 50011us) TIV 50011us

ID: 0 IV: 14999636 (= 49999us) TIV 49999us

ID: 1 IV: 14999022 (= 49997us) TIV 49997us

ID: 0 IV: 14999605 (= 49999us) TIV 49999us

ID: 1 IV: 15001838 (= 50006us) TIV 50006us

ID: 0 IV: 14999516 (= 49998us) TIV 49998us

ID: 1 IV: 14999038 (= 49997us) TIV 49997us

ID: 0 IV: 14999601 (= 49999us) TIV 49999us

ID: 1 IV: 15001419 (= 50005us) TIV 50005us

ID: 0 IV: 14999552 (= 49999us) TIV 49999us

ID: 1 IV: 14999005 (= 49997us) TIV 49997us

ID: 0 IV: 14999531 (= 49998us) TIV 49998us

ID: 1 IV: 15000889 (= 50003us) TIV 50003us

ID: 0 IV: 14999583 (= 49999us) TIV 49999us

ID: 1 IV: 14998992 (= 49997us) TIV 49997us

ID: 0 IV: 14999575 (= 49999us) TIV 49999us

ID: 1 IV: 14970950 (= 49903us) TIV 49903us

ID: 0 IV: 14999643 (= 49999us) TIV 49999us

ID: 1 IV: 14999049 (= 49997us) TIV 49997us

ID: 0 IV: 14999495 (= 49998us) TIV 49998us

ID: 1 IV: 15004717 (= 50016us) TIV 50016us

ID: 0 IV: 14999592 (= 49999us) TIV 49999us

ID: 1 IV: 14999082 (= 49997us) TIV 49997us

ID: 0 IV: 14999490 (= 49998us) TIV 49998us

ID: 1 IV: 15001802 (= 50006us) TIV 50006us

ID: 0 IV: 14999589 (= 49999us) TIV 49999us

ID: 1 IV: 14999108 (= 49997us) TIV 49997us

ID: 0 IV: 14999495 (= 49998us) TIV 49998us

ID: 1 IV: 15001311 (= 50004us) TIV 50004us

ID: 0 IV: 14999588 (= 49999us) TIV 49999us

ID: 1 IV: 14999089 (= 49997us) TIV 49997us

ID: 0 IV: 14999574 (= 49999us) TIV 49999us

RANGE 49888-50016us (128us)

Global timer tests ...

expectedTS 30000000 expectedGT 30000000

gt interval : 30228407 (gtticks) 100761 (us)

gt interval : 30372942 (gtticks) 101242 (us)

gt interval : 30381775 (gtticks) 101272 (us)

gt interval : 30376362 (gtticks) 101254 (us)

gt interval : 30375542 (gtticks) 101251 (us)

gt interval : 30386749 (gtticks) 101289 (us)

gt interval : 30382852 (gtticks) 101275 (us)

gt interval : 30375887 (gtticks) 101252 (us)

gt interval : 30385175 (gtticks) 101284 (us)

gt interval : 30377029 (gtticks) 101256 (us)

RANGE Global Timer 30228407-30386749 ticks (158342 ticks)

RANGE Timestamp 100761-101289 us (528 us)

expectedTS 30000000 expectedGT 15000000

gt interval : 15106340 (gtticks) 100709 (us)

gt interval : 15192460 (gtticks) 101282 (us)

gt interval : 15190009 (gtticks) 101266 (us)

gt interval : 15188232 (gtticks) 101254 (us)

gt interval : 15191574 (gtticks) 101277 (us)

gt interval : 15188852 (gtticks) 101259 (us)

gt interval : 15192130 (gtticks) 101280 (us)

gt interval : 15186913 (gtticks) 101246 (us)

gt interval : 15193392 (gtticks) 101289 (us)

gt interval : 15187784 (gtticks) 101251 (us)

RANGE Global Timer 15106340-15193392 ticks (87052 ticks)

RANGE Timestamp 50354-50644 us (290 us)

expectedTS 30000000 expectedGT 10000000

gt interval : 10057592 (gtticks) 100576 (us)

gt interval : 10128789 (gtticks) 101287 (us)

gt interval : 10126793 (gtticks) 101267 (us)

gt interval : 10125401 (gtticks) 101254 (us)

gt interval : 10128074 (gtticks) 101280 (us)

gt interval : 10125630 (gtticks) 101256 (us)

gt interval : 10127999 (gtticks) 101279 (us)

gt interval : 10127785 (gtticks) 101277 (us)

gt interval : 10125740 (gtticks) 101257 (us)

gt interval : 10128182 (gtticks) 101281 (us)

RANGE Global Timer 10057592-10128789 ticks (71197 ticks)

RANGE Timestamp 33525-33762 us (237 us)

expectedTS 30000000 expectedGT 7500000

gt interval : 7543604 (gtticks) 100581 (us)

gt interval : 7593498 (gtticks) 101246 (us)

gt interval : 7595280 (gtticks) 101270 (us)

gt interval : 7595938 (gtticks) 101279 (us)

gt interval : 7594277 (gtticks) 101256 (us)

gt interval : 7593752 (gtticks) 101250 (us)

gt interval : 7596012 (gtticks) 101280 (us)

gt interval : 7594584 (gtticks) 101261 (us)

gt interval : 7597920 (gtticks) 101305 (us)

gt interval : 7594200 (gtticks) 101256 (us)

RANGE Global Timer 7543604-7597920 ticks (54316 ticks)

RANGE Timestamp 25145-25326 us (181 us)

expectedTS 30000000 expectedGT 6000000

gt interval : 6036231 (gtticks) 100604 (us)

gt interval : 6075396 (gtticks) 101256 (us)

gt interval : 6076957 (gtticks) 101282 (us)

gt interval : 6076274 (gtticks) 101271 (us)

gt interval : 6075226 (gtticks) 101253 (us)

gt interval : 6076811 (gtticks) 101280 (us)

gt interval : 6075534 (gtticks) 101259 (us)

gt interval : 6075050 (gtticks) 101250 (us)

gt interval : 6076583 (gtticks) 101276 (us)

gt interval : 6077031 (gtticks) 101283 (us)

RANGE Global Timer 6036231-6077031 ticks (40800 ticks)

RANGE Timestamp 20120-20256 us (136 us)

expectedTS 30000000 expectedGT 5000000

gt interval : 5044821 (gtticks) 100896 (us)

gt interval : 5064262 (gtticks) 101284 (us)

gt interval : 5061864 (gtticks) 101237 (us)

gt interval : 5062825 (gtticks) 101256 (us)

gt interval : 5064596 (gtticks) 101291 (us)

gt interval : 5063617 (gtticks) 101272 (us)

gt interval : 5063117 (gtticks) 101262 (us)

gt interval : 5063548 (gtticks) 101270 (us)

gt interval : 5062627 (gtticks) 101252 (us)

gt interval : 5065918 (gtticks) 101318 (us)

RANGE Global Timer 5044821-5065918 ticks (21097 ticks)

RANGE Timestamp 16816-16886 us (70 us)

expectedTS 30000000 expectedGT 4285714

gt interval : 4324117 (gtticks) 100895 (us)

gt interval : 4340472 (gtticks) 101277 (us)

gt interval : 4338860 (gtticks) 101240 (us)

gt interval : 4340713 (gtticks) 101283 (us)

gt interval : 4339406 (gtticks) 101252 (us)

gt interval : 4340642 (gtticks) 101281 (us)

gt interval : 4340630 (gtticks) 101281 (us)

gt interval : 4339342 (gtticks) 101251 (us)

gt interval : 4340763 (gtticks) 101284 (us)

gt interval : 4339434 (gtticks) 101253 (us)

RANGE Global Timer 4324117-4340763 ticks (16646 ticks)

RANGE Timestamp 14413-14469 us (56 us)

expectedTS 30000000 expectedGT 3750000

gt interval : 3775856 (gtticks) 100689 (us)

gt interval : 3797946 (gtticks) 101278 (us)

gt interval : 3796678 (gtticks) 101244 (us)

gt interval : 3797908 (gtticks) 101277 (us)

gt interval : 3796905 (gtticks) 101250 (us)

gt interval : 3798076 (gtticks) 101282 (us)

gt interval : 3798515 (gtticks) 101293 (us)

gt interval : 3796527 (gtticks) 101240 (us)

gt interval : 3798662 (gtticks) 101297 (us)

gt interval : 3796539 (gtticks) 101240 (us)

RANGE Global Timer 3775856-3798662 ticks (22806 ticks)

RANGE Timestamp 12586-12662 us (76 us)

expectedTS 30000000 expectedGT 3333333

gt interval : 3356881 (gtticks) 100706 (us)

gt interval : 3376083 (gtticks) 101281 (us)

gt interval : 3375910 (gtticks) 101277 (us)

gt interval : 3375026 (gtticks) 101250 (us)

gt interval : 3375277 (gtticks) 101258 (us)

gt interval : 3375781 (gtticks) 101273 (us)

gt interval : 3375061 (gtticks) 101251 (us)

gt interval : 3377068 (gtticks) 101312 (us)

gt interval : 3375258 (gtticks) 101257 (us)

gt interval : 3375949 (gtticks) 101278 (us)

RANGE Global Timer 3356881-3377068 ticks (20187 ticks)

RANGE Timestamp 11189-11256 us (67 us)

expectedTS 30000000 expectedGT 3000000

gt interval : 3021229 (gtticks) 100707 (us)

gt interval : 3038368 (gtticks) 101278 (us)

gt interval : 3037347 (gtticks) 101244 (us)

gt interval : 3038671 (gtticks) 101289 (us)

gt interval : 3038395 (gtticks) 101279 (us)

gt interval : 3037438 (gtticks) 101247 (us)

gt interval : 3038489 (gtticks) 101282 (us)

gt interval : 3037680 (gtticks) 101256 (us)

gt interval : 3039438 (gtticks) 101315 (us)

gt interval : 3037880 (gtticks) 101262 (us)

RANGE Global Timer 3021229-3039438 ticks (18209 ticks)

RANGE Timestamp 10070-10131 us (61 us)

RTEST: SUCCESS : T\_FRQCHG test completed O.K.

1. In current implementations timer frequency needs to be > 1Mhz, and the times stamp is 64bit [↑](#footnote-ref-1)
2. When global timer is used [↑](#footnote-ref-2)