

TPT Severe Issues

Introduction

=====

The following document contains a list of known severe issues of TPT. By severe issues we mean issues/bugs in particular versions of TPT that:

1. might cause malfunctions in the behavior of TPT
2. are hard or even impossible to find by the TPT user herself/himself
3. cause the risk that bugs/defects in a SUT (system under test) are not detected by TPT in cases where TPT would have been able to reveal these bugs/defects in the SUT without the aforementioned malfunction in the behavior of TPT.

Usually these severe issues address the situations where the problem might appear and have well-defined workarounds.

ISSUE # 30102

=====

TITLE:

When comparing INT64 signals using Min/Max or Signal Comparison assesslet with values larger than 2^{53} or smaller than -2^{53} , the computation can incorrectly compare the signal with the reference(s) which might lead to PASSED results even if the specified bounds are exceeded.

ISSUE DETECTION:

11-January-2021

AFFECTED VERSIONS OF TPT:

TPT 8 - TPT 16

PRECONDITIONS:

The Min/Max or Signal Comparison assesslet is used with INT64 signals with values larger than 2^{53} or smaller than -2^{53} .

DETAILS:

If signal or reference values are larger than 2^{53} or smaller than -2^{53} , the difference can be missed because the values are converted and compared as double values. (Since values larger/smaller than $2^{53}/-2^{53}$ cannot be converted to double without losing precision, floating point precision problems might occur in such cases.)

EFFECT OF THE ISSUE:

Values outside of the specified bounds might be overlooked by TPT leading to PASSED results, but should be FAILED.

WORKAROUND:

Avoid using INT64 signals in Min/Max or Signal Comparison assesslets if the values are larger than 2^{53} or smaller than -2^{53} .

RESOLVED IN:
TPT 15u4, TPT 16u1

ISSUE # 30063
=====

TITLE:
When iterating in assessment scripts via an inlined loop and applying signal processing functions like TPT.average(), TPT.min(), ... with changing argument in each loop, the result of the first iteration is incorrectly used for the following iterations.

ISSUE DETECTION:
16-December-2020

AFFECTED VERSIONS OF TPT:
TPT 8 - TPT 16

PRECONDITIONS:
The usage of the signal processing function must be applied on an expression or a signal that is being changed while iterating through an inlined loop or list comprehension.

DETAILS:
For faster computation of timed expressions of the form
foo(t) := TPT.average(...)
results of signal processing functions are being cached.
The cached result is invalidated as soon as a new line is reached.
When iterating through an inlined loop, the same expression is evaluated multiple times.
If during this inlined iteration a variable of the expression is changed, the cached result of the signal processing function is used instead of a newly calculated.

Affected are inlined expressions of the form
for x in range(n): print TPT.min(...+x)
while x < n: x=x+1;print TPT.min(...+x)
as well as list comprehension:
my_list = [TPT.min(...+x) for x in range(n)].

EFFECT OF THE ISSUE:
Old cached values are used instead of recalculating the value in every iteration, which results in wrong computation results.

WORKAROUND:
Avoid using inlined loops or list comprehensions and use loops with indentation in multiple code lines instead.

RESOLVED IN:
TPT 15u4 TPT 16u1

ISSUE # 30064
=====

TITLE:
When running assesslets in parallel using the multicore feature and both assesslets

access the same signals imported from the same signal file, manipulation of such a signal can be seen in the other parallel test run.

ISSUE DETECTION:
16-December-2020

AFFECTED VERSIONS OF TPT:
TPT 16

PRECONDITIONS:
The number of cores in the Execution Configuration dialog must be greater than 1 and some assesslet executed in parallel must use the same signals imported from the same signal file, e.g. via TPT.readRecord() or Import Measurements Assesslet and at least one assesslet must manipulate such a signal. These assesslets may be the same assesslet linked to multiple test cases.

DETAILS:
For faster execution imported signal files are cached. The cache is invalidated as soon as a signal is manipulated. When running assesslets in parallel multiple assesslets can get a reference to the same signal before it is manipulated. So any manipulation of the signal is shared between these assesslets.

EFFECT OF THE ISSUE:
Assesslets see values and time intervals of signals set and created by other parallel running assesslets.

WORKAROUND:
When manipulation of imported signals (from external measurement files) in the assessment is needed, do not run those test cases with multiple cores.

RESOLVED IN:
TPT 16u1

ISSUE # 31080
=====

TITLE:
When using the C/C++ Platform to connect a bit-field for which data type in TPT differs from the C-Code the generated test frame might read or write the data incorrectly.

ISSUE DETECTION:
09-July-2021

AFFECTED VERSIONS OF TPT:
TPT 15 - TPT 16

PRECONDITIONS:
The C/C++ Platform is used to connect a C-Variable with a bit-field. The declared data type for the bit field differs from the data type used for the corresponding struct element in the TPT declaration.

DETAILS:
When TPT generates the get/set functions to access the bit field data via the public TPT-VM-API function "tpt_vmapi_bindSignalGetSet" or "tpt_vmapi_bindSignalGetSet_v2", TPT mistakenly used the type

specified

in TPT when accessing the incoming pointer. Instead the data type passed to "tpt_vmapi_bindSignalGetSet"/"tpt_vmapi_bindSignalGetSet_v2" as fixed point data type shall be used.

EFFECT OF THE ISSUE:

At test execution with the C/C++ platform the data for a bit field value may be written to or read from the SUT incorrectly. If the bit-size of the data type in TPT and the type of the bit field does not match, the memory with the SUT or the TPT VM may be corrupted at test runtime.

WORKAROUND:

Manually check and adjust the code generated by C/C++ Platform in case of bit-field data types to use the correct data type within get/set function used with "tpt_vmapi_bindSignalGetSet" or "tpt_vmapi_bindSignalGetSet_v2".

RESOLVED IN:

TPT 16u3