(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 October 2016 (06.10.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/155830 A1

(51) International Patent Classification:
G01R 31/319 (2006.01)

(21) International Application Number:
PCT/EP2015/057273

(22) International Filing Date:
1 April 2015 (01.04.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **ADVANTEST CORPORATION** [JP/JP]; 1-32-1, Asahi-cho Nerima-ku, Tokyo, 179-0071 (JP).

(72) Inventor; and
(71) Applicant *(for US only)*: **HORN, Wolfgang** [DE/DE]; Schmellbachstraße 5B, 70565 Stuttgart (DE).

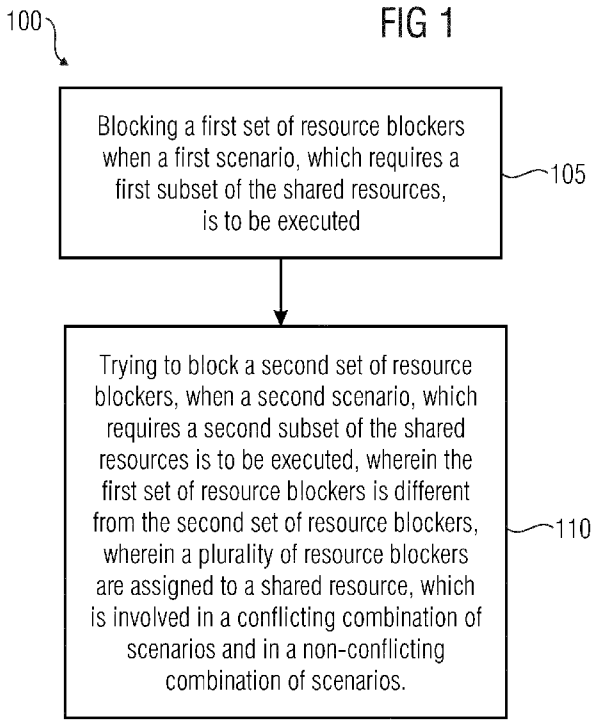
(74) Agents: **BURGER, Markus** et al.; Schoppe, Zimmermann, Stöckeler, Zinkler, Schenk & Partner mbB, Radlkoferstrasse 2, 81373 München (DE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD FOR OPERATING A TEST APPARATUS AND A TEST APPARATUS



(57) Abstract: A method for operating a test apparatus comprising a plurality of shared resources is shown, wherein the plurality of shared resources can be used in different instruments. The method comprises blocking a first set of resource blockers when a first instrument, which requires a first subset of the shared resources, is to be executed. Furthermore, the method tries to block a second set of resource blockers, when a second instrument, which requires a second subset of the shared resources, is to be executed. Therefore, the first set of resource blockers is different from the second set of resource blockers and a plurality of resource blockers are assigned to a shared resource, which is involved in a conflicting combination of instruments and in a non-conflicting combination of instruments.

WO 2016/155830 A1

Published:

— *with international search report (Art. 21(3))*

Method for Operating a Test Apparatus and a Test Apparatus

Specification

Embodiments of the present invention relate to a method for operating a test apparatus using shared resources and the corresponding apparatus. Further embodiments relate to virtual shared resources of virtual resource sharing.

Whenever hardware resources are shared, these are typically expensive and it is desirable to use them as efficiently as possible. In general, this is achieved by locking the resources for a certain period of usage, i.e. to serialize measurements either always or never. As long as the shared resource is locked by a certain thread of execution, no other thread can use the resource even though, depending on the scenario, this may be possible. In software that locking is typically achieved in a multi-threaded environment with the help of a mutex object per shared resource. A second thread of execution which wants to use the shared resource concurrently will try to lock the mutex, and if another thread currently owns that shared resource, the concurrent thread will be blocked until the resource is freed by the first thread.

In more complex situations involving multiple shared resources, it may become necessary to lock multiple mutexes in a certain order to make sure no deadlocks arise in concurrent scenarios. Ensuring the same sequence of locking at all times ensures that no thread of execution owns resources required by the other and vice versa.

If the decision whether a set of resources can be used concurrently depends on setup parameters (e.g. because of a multisite scenario) then the pure locking cannot be applied anymore, but the software may analyze the scenario and decide which resources need to be locked and whether that is possible. Even worse, if two threads execute compatible scenarios (i.e. they can run concurrently) but still both need the shared resource, then they would block each other if simple locking as indicated above would be applied. Not locking the shared resource is not a valid solution in case there are other scenarios which actually do have a conflict with either of the compatible scenarios. The same applies to semaphores, which would allow multiple threads to lock the resource, but it cannot control which threads to give access to.

In other words, some hardware allows concurrent use of a shared resource in specific scenarios (i.e. under specific conditions) while at the same time it disallows using the same shared resource for scenarios with conflicting conditions. Making maximum use of the shared resource, it is advantageous to organize the sharing according to the scenarios.

Therefore, there is the need for an improved approach. It is an objection of the invention to provide an improved concept for using shared resources. This object is solved by the subject matter of the independent claims.

According to a first embodiment, a method for operating a test apparatus comprising a plurality of shared resources is shown, wherein the plurality of shared resources can be used in different scenarios by different instruments. The method comprises blocking a first set of resource blockers when a first instrument, which requires a first subset of the shared resources, is to be executed. Furthermore, the method tries to block a second set of resource blockers, when a second instrument, which requires a second subset of the shared resources, is to be executed. Therefore, the first set of resource blockers is different from the second set of resource blockers and a plurality of resource blockers are assigned to a shared resource, which is involved in a conflicting combination of instruments and in a non-conflicting combination of instruments.

Teachings disclosed herein are based on the fundamental idea that an appropriate blocking of shared resources ensures multiple use of the shared resource for scenarios or instruments which can be operated concurrently and single use of the shared resource if an instrument needs a shared resource exclusively. Therefore, multiple resource blockers may be assigned to the shared resources according to the conflicting and non-conflicting instruments or scenarios. When a thread executes a specific instrument or scenario, it locks for each affected shared resource all resource blockers associated with that instrument or scenario. The shared resource is thus virtually split into multiple resources, where a subset can be locked to exclude specific other instruments from execution. The mapping of the resource blockers can be done by different algorithms, for instance by definition of a scenario compatibility matrix, where each exclusion is represented by a number indexing into the list of resource blockers associated with a shared resource.

The term instrument relates to one or more resources (some of which may be shared between different instruments), which are (virtually) forming instruments, wherein a first instrument may comprise a first shared resource, a second instrument may comprise a second shared resource and a third and a fourth instrument may comprise a combination of the first and the second shared resource, e.g. using different amplifications or gains of one of the shared resources. However, the assignment is arbitrary and mutually applicable. Furthermore, a scenario comprises multiple instruments in a specific order. Therefore, instruments may use shared resources concurrently and scenarios may use either instruments or shared resources concurrently, where each concurrency may be conflicting or non-conflicting. If instruments use a shared resource concurrently, obviously, one or more scenarios comprising the instruments also use the shared resource concurrently. Therefore, the terms scenario and instrument are mutually applicable when referring to concurrent usage of shared resources.

Embodiments show the method executing a first test program portion (e.g. a thread) configured to operate (i.e. use) the first instrument. The test program portion comprises checking a first set of resource blockers associated with the first instrument, blocking the first set of resource blockers if none of the resource blockers of the first set of resource blockers are blocked, and selectively performing a test using a tester configuration defined by the first instrument, if it was found in the checking step that none of the resource blockers of the first set of resource blockers are blocked. Furthermore, a second test program portion configured to operate the second instrument is executed. The second test program portion comprises checking a second set of resource blockers associated with a second instrument, blocking the second set of resource blockers if none of the resource blockers of the second set of resource blockers are blocked, and selectively performing a test using a tester configuration defined by the second instrument if it was found in the checking step that none of the resource blockers of the second set of resource blockers are blocked. The second test program portion is executed concurrently with the first test program portion or with a temporal overlap with the first test program portion. In other words, a shared resource may be executed concurrently with another shared resource under a first condition, but under a second condition, the test program needs exclusive access to one of the shared resources. This is advantageous, since an instrument-based (i.e. context-based) usage of shared resources can significantly reduce the execution time of the whole test program. If more than one scenario is performed concurrently, a scenario-based usage may even improve the instrument-based usage.

According to a further embodiment, the method comprises waiting and checking the second set of resource blockers again if one of the resource blockers in the relevant subset is blocked or the method comprises executing a further part of the second test program portion and checking a second set of resource blockers again after the execution of the further part is finished. This is advantageous, since a current test program is further executed if a currently requested resource is blocked. If it is not possible to further execute the test program portion, the method waits until the relevant shared resources for the test program portion are unblocked and further executes the test program portion accordingly.

Embodiments further show the method comprising four different instruments, wherein the first and the second instrument constitute a non-conflicting combination of instruments using the first and the second subset of shared resources, respectively, and wherein a third and a fourth instrument constitute a non-conflicting combination of instruments using both a combination of the first and the second subset of shared resources. Therefore, the first and the second instrument are each in a conflict to both the third and the fourth instrument.

The method further comprises four different resource blockers, wherein each of the resource blockers is assigned to one virtual shared resource. A first and a second virtual shared resource are assigned to the first shared resource and a third and a fourth virtual shared resource are assigned to the second shared resource. The first set of resource blockers comprises the first and the second virtual shared resource, the second set of resource blockers comprises the third and the fourth virtual shared resource, a third set of resource blockers comprises the first and the third virtual shared resource, and a fourth set of resource blockers comprises the second and the fourth virtual shared resource. According to the first and the second test program portion, the method executes a third test program portion configured to operate the third instrument. The third test program portion comprises checking the third set of resource blockers associated with a third instrument, blocking the third set of resource blockers if none of the resource blockers in the third set of resource blockers are blocked, and selectively performing a test using a tester configuration defined by the third instrument if it was found in the checking step that none of the resource blockers in the third set of resource blockers are blocked.

Accordingly, a fourth test program portion configured to operate the fourth instrument is executed. The fourth test program portion comprises checking the fourth set of resource

blockers associated with the fourth instrument, blocking the fourth set of resource blockers if none of the resource blockers of the fourth set of resource blockers are blocked, and selectively performing a test using a tester configuration defined by the fourth instrument, if it was found in the checking step that none of the resource blockers in the fourth set of resource blockers are blocked. At least two of the first, the second, the third, and the fourth test program portions are executed concurrently or with a temporal overlap to one another. This is again advantageous in terms of an execution time of the test program since, in this case, four different test program portions are optimized in terms of accessing two different shared resources.

According to an embodiment, the resource blockers are a bit or a plurality of bits in a storing unit to indicate a usage of the shared resources. This is advantageous, since multiple bits are assigned to a shared resource, indicating instruments where a shared resource may be used concurrently. The resource blocker may be further implemented as a mutex, a semaphore, a lock or a monitor.

According to embodiments, one of the instruments is a virtual device configured to use a signal which is based on one of the shared resources or configured to generate a signal which is based on a combination of signals of multiple shared resources. For example, this is a one-tone generator using one of the shared resources or a two-tone generator using a combination of two of the shared resources.

Moreover, multiple instruments may be operated in one or more scenarios. Therefore, the description referring to instruments may be also applied to scenarios.

Further embodiments show the shared resource as a physical stimulus device. The shared resource may be a signal generator to generate a stimulus signal, which may be an analog stimulus signal, wherein the stimulus signal is e.g. a radio frequency signal. According to another embodiment, the shared resource is a digital to analog converter. The speed of the analog to digital conversion depends on parameters like sampling rate and/or resolution (or quantization) of the digital inbound signal. Therefore, it is advantageous to use the digital to analog converter concurrently, for example for two different digital inbound signals and two different instruments.

The previously described method may be implemented in a computer program to run on a computer.

Furthermore, an embodiment shows a test apparatus comprising a plurality of shared resources which can be used in different instruments. Furthermore, the test apparatus comprises a method for operating the test apparatus comprising blocking a first set of resource blockers when a first instrument, which requires a first subset of the shared resources, is to be executed, trying to block a second set of resource blockers, when a second instrument, which requires a second subset of the shared resources is to be executed, wherein the first set of resource blockers is different from the second set of resource blockers, and wherein a plurality of resource blockers are assigned to a shared resource, which is involved in a conflicting combination of instruments and a non-conflicting combination of instruments.

Embodiments of the present invention will be discussed subsequently referring to the enclosed drawings, wherein:

Fig. 1 shows a schematic block diagram of a method for operating a test apparatus;

Fig. 2 shows a schematic block diagram of a test apparatus, which may be operated by the method shown in Fig. 1;

Fig. 3 shows a schematic sequence diagram illustrating an exemplary scenario of the method shown in Fig. 1;

Fig. 4 shows a schematic table indicating the allocations of shared and virtual shared resources corresponding instruments.

Embodiments of the present invention will be discussed in detail below, wherein identical reference numbers are provided to objects having identical or similar functions, so that the description thereof is interchangeable or mutually applicable.

Fig. 1 shows a schematic block diagram of a method for operating a test apparatus comprising a plurality of shared resources, which can be used in different instruments. The method comprises blocking a first set of resource blockers when a first instrument, which requires a first subset of the shared resources, is to be executed in a step 105, and trying to block a second set of resource blockers, when a second instrument, which

requires a second subset of the shared resources is to be executed, wherein the first set of resource blockers is different from the second set of resource blockers in a step 110. Therefore, a plurality of resource blockers are assigned to a shared resource, which is involved in a conflicting combination of instruments and in a non-conflicting combination of instruments.

A resource blocker may be a bit or a plurality of bits in a storing unit to indicate a usage of the shared resources. In a practical application, a mutex, a semaphore, a lock or a monitor may be used, which are already pre-implemented as functions in many high level programming languages. A schematic and exemplary assignment of resource blockers to instruments and shared resources is shown in Fig. 4. The shared resources and the instruments are explained with respect to Figs. 2 and 3.

The term "trying to block" a set of resource blockers refers to checking in a first step whether the resource blockers are already blocked, and blocking the resource blockers if they are not already blocked. If the resource blockers are already blocked, it may be waited until the resource blockers are unblocked or it may be executed a further instrument meanwhile.

A "different" set of resource blockers indicates at least one resource blocker which is different in the first set of resource blockers compared to the second set of resource blockers. A different set of resource blockers is, for example, if the first set of resource blockers comprises a first and a second resource blocker, and the second set of resource blockers comprises only the first resource blocker, or alternatively the first and a third resource blocker.

Fig. 2 shows a schematic block diagram of a test apparatus 5, which may be operated by the method 100 shown in Fig. 1. The test apparatus comprises a shared resource A (SR-A) 10a and a shared resource B (SR-B) 10b. Switches 15a, 15a', 15b, 15b' are configured to provide a stimulus signal 17a, b of the shared resources directly to a first and/or a second device under test 20a, 20b, or alternatively to provide a combined signal 17c to the first and/or the second device under test 20a, 20b, wherein the combined signal 17c is a combination of the stimulus signal 17a, b of the shared resources combined by a tone combiner 25. In other words, the output of the tone-combiner 17c may be routed back to both device under test (DUT) connections 17a, 17b and corresponding switches 15a, 15a', 15b, 15b' can then be used to select the path through the tone-combiner or

individual paths beneath it. Thus, the shared resources can operate as either two single-tone instruments or virtually two dual-tone instruments. The (virtually) different instruments 35a-d, are indicated by the corresponding frames. Using the method 100, the test apparatus 5 is configured to operate multiple instruments using the shared resources.

In a more general case, the term "instrument" can be replaced with the term "scenario", wherein a scenario comprises one or more instruments. Both, the scenario and the instruments, may comprise a plurality of shared resources in multiple combinations, i.e. stimuli and/or measurement devices. Both will be described with respect to Fig. 3. Assuming that instruments in the tester (short for test apparatus) have shared resources 10a, b, which however may sometimes still be used concurrently. Each combination of instruments to be used concurrently from multiple threads despite the shared resources should not try to lock the same mutexes (or shared resources) while instruments or scenarios wherein exclusive access of a shared resource is required, must lock at least one mutex overlapping with the conflicting instruments.

According to an embodiment, the system provides four distinct instruments using two shared resources SR-A and SR-B. Instrument 1 uses SR-A only, instrument 2 uses SR-B only, and instrument 3 and instrument 4 use both shared resources SR-A and SR-B. The instruments 1 and 2 may describe two single source radio frequency stimuli and the instruments 3 and 4 may describe two dual source radio frequency stimuli. Instruments 1 and 2 are certainly usable concurrently, since they do not share resources. Furthermore, instruments 3 and 4 should be usable concurrently (in the radio frequency case special care is taken that this happens only with compatible settings for the dual tone). However, it will not be possible to use instrument 1 or 2 concurrently with instrument 3 or 4, since the routing of signals from the sources either goes through the tone combiner 25 or it does not.

A shared resource 10a or 10b may be a physical input/output device like, e.g. a signal generator to generate an analog stimulus signal 17a, 17b, which may be a radio frequency signal, or a measurement device. According to a further embodiment, the shared resources 10a, 10b may be digital to analog converters configured to convert a digital inbound signal to the analog outbound signal 17a, 17b, e.g. to be used as a stimulus signal for a device under test. The speed of the analog to digital conversion depends on parameters like sampling rate and/or resolution (or quantization) of the digital inbound signal. Therefore, at least one of the digital to analog converters may be used

concurrently, for example for two different digital inbound signals and two different instruments, if e.g. the settings allow a digital to analog conversion which is twice as fast as a further signal processing. In this case, a converted signal may be temporarily stored in a storing unit (or a buffer), e.g. a (flash) memory and further processed from the storing unit.

The idea of how to cope with the instrument-based differentiation of a concurrent or an exclusive use of a shared resource is to virtually split the shared resources 10a, 10b into virtual shared resources. According to the previously described embodiment, a shared resource A is split into a virtual shared resource A1 and a virtual shared resource A2, and a shared resource B is split into a virtual shared resource B1 and a virtual shared resource B2. From now on, simple locking as described in Fig. 4 can achieve exactly the desired behavior. Since instrument 3 and instrument 4 should be able to run concurrently despite their shared resources, they each lock half the resources, i.e. only one out of two associated virtual resources. Furthermore, instrument 1 and instrument 2 require exclusive access to shared resource A and shared resource B, respectively, and therefore lock both virtual shared resources, i.e. the complete shared resource.

This mechanism is universally applicable whenever two instruments should run concurrently even though they make use of the same resources. For each such case, the affected resources need to provide duplicated locks and only half the locks are taken in each instrument. The instruments which should still be locked out from concurrency will simply lock all resources.

Fig. 3 shows a schematic sequence diagram illustrating an exemplary scenario 101 of the method 100. To better illustrate the plurality of certain steps (or instruments) of the method 100, these steps are aligned on the left and the right hand side of the resource blockers 30a to 30d in the middle of the diagram. The resource blocker 30a refers to the virtual resource SR-A1, 30b refers to the virtual resource SR-A2, 30c refers to the virtual resource SR-B1, and 30d refers to the virtual resource SR-B2. The method starts by executing instrument 1 35a. A checking step may optionally be implemented to check the relevant virtual shared resources SR-A1 and SR-A2 30a, 30b. This is optional since the resource blockers are initially not blocked in this specific scenario. In general, the nature of (multi-threaded) concurrent execution implies that the individual scenario cannot know whether it gets executed first. Before selectively performing a test using a tester configuration defined by the first instrument, the first set of resource blockers 40a, 40b,

belonging to the virtual shared resources 30a and 30b, are blocked. While a test using the configuration of instrument 1 is performed, a test using the configuration of instrument 2 35b is to be executed. In a further step, the set of resource blockers 40c, 40d is relevant to the second instrument 35b checked with respect to the state (blocked or unblocked) of the resource blockers 40c, 40d. If none of the checked resource blockers are blocked, the test program blocks these resource blockers. Thereafter, the test using instrument 2 is performed.

While the test using instrument 2 is running, the test using instrument 1 is finished. Therefore, the test program unblocks the relevant resource blockers 40a, 40b to indicate that shared resource A 10a is free.

A further test program portion is trying to perform a test using instrument 3. In a first step, the relevant resource block 40a, 40c are checked. In this case, instrument 2 still blocks the resource blocker of the virtual resource SR-B1 40c implying that the test using instrument 3 cannot be executed at that moment. As indicated in Fig. 3, the test program may simply wait for the second instrument to finish or, not shown in Fig. 3, a further test program portion may be executed meanwhile, if possible. However, once the checking of instrument 3 is positive (because the test program running instrument 2 unblocked the resource blockers 40c and 40d), the relevant resource blockers 40a and 40c are blocked and a test according to instrument 3 35c is performed. Parallel to instrument 3 35c, instrument 4 35d may be operated.

Although this scenario is exemplary and shall not limit the generality, it shows that the shared resources A and B, 10a, 10b can run concurrently in those instruments where concurrency is possible. However, if instruments block each other, an instrument trying to access a shared resource which is already blocked needs to wait for execution at least until the resource blocker is freed.

Introducing a mapping from instrument (or scenario) based decisions of what may run concurrently and what not to basic synchronization mechanism makes the solution more easily portable and less deadlock prone than conventional point solutions. The preferred embodiment virtually splits the locks associated with shared resources and assigns each instrument, which should still own their shared resource exclusively over all other instruments, the whole set of locks, while instruments that should run concurrently with each other will be assigned only a subset of locks. Thus, after this mapping is applied the

problem is reduced to simple serial locking of all assigned locks. Therefore, it may be defined a sequence of locking which is always executed in the same order to avoid deadlocks

Fig. 4 shows a schematic table indicating the allocations of shared and virtual shared resources 10a, b and 30a-d to the corresponding instruments 35a-d. An X indicates a blocking of the resource block and an O indicates an unblocked resource blocker.

As is stated clearly, this mechanism is universally applicable whenever two instruments should run concurrently even though they make use of the same resources. For each such case, the affected resources need to provide duplicated locks (resource blockers) and only half the locks will be taken in each instrument. The instruments which should still be locked out from concurrency will simply lock all virtual shared resources.

Depending on certain implementation requirements, embodiments of the invention can be implemented in hardware or in software. The implementation can be performed using a digital storage medium, for example a floppy disc, a DVD, a Blu-Ray, a CD, a ROM, a PROM, and EPROM, an EEPROM or a FLASH memory, having electronically readable control signals stored thereon, which cooperate (or are capable of cooperating) with a programmable computer system such that the respective method is performed. Therefore, the digital storage medium may be computer readable.

Some embodiments according to the invention comprise a data carrier having electronically readable control signals, which are capable of cooperating with a programmable computer system, such that one of the methods described herein is performed.

Generally, embodiments of the present invention can be implemented as a computer program product with a program code, the program code being operative for performing one of the methods when the computer program product runs on a computer. The program code may, for example, be stored on a machine readable carrier.

Other embodiments comprise the computer program for performing one of the methods described herein, stored on a machine readable carrier.

In other words, an embodiment of the inventive method is, therefore, a computer program having a program code for performing one of the methods described herein, when the computer program runs on a computer.

A further embodiment of the inventive method is, therefore, a data carrier (or a non-transitory storage medium such as a digital storage medium, or a computer-readable medium) comprising, recorded thereon, the computer program for performing one of the methods described herein. The data carrier, the digital storage medium or the recorded medium are typically tangible and/or non-transitory.

A further embodiment of the invention method is, therefore, a data stream or a sequence of signals representing the computer program for performing one of the methods described herein. The data stream or the sequence of signals may, for example, be configured to be transferred via a data communication connection, for example, via the internet.

A further embodiment comprises a processing means, for example, a computer or a programmable logic device, configured to, or adapted to, perform one of the methods described herein.

A further embodiment comprises a computer having installed thereon the computer program for performing one of the methods described herein.

A further embodiment according to the invention comprises an apparatus or a system configured to transfer (for example, electronically or optically) a computer program for performing one of the methods described herein to a receiver. The receiver may, for example, be a computer, a mobile device, a memory device or the like. The apparatus or system may, for example, comprise a file server for transferring the computer program to the receiver.

In some embodiments, a programmable logic device (for example, a field programmable gate array) may be used to perform some or all of the functionalities of the methods described herein. In some embodiments, a field programmable gate array may cooperate with a microprocessor in order to perform one of the methods described herein. Generally, the methods are preferably performed by any hardware apparatus.

The above described embodiments are merely illustrative for the principles of the present invention. It is understood that modifications and variations of the arrangements and the details described herein will be apparent to others skilled in the art. It is the intent, therefore, to be limited only by the scope of the impending patent claims and not by the specific details presented by way of description and explanation of the embodiments herein.

Claims

What is claimed is:

1. A method (100) for operating a test apparatus (5) comprising a plurality of shared resources (10a, 10b), which can be used in different instruments (35a-35d), the method (100) comprising:

blocking a first set of resource blockers (40a, 40b) when a first instrument (35a), which requires a first subset of the shared resources (10a), is to be executed;

trying to block a second set of resource blockers (40c, 40d), when a second instrument (35b), which requires a second subset of the shared resources (10b) is to be executed, wherein the first set of resource blockers (40a, 40b) is different from the second set of resource blockers (40c, 40d); and

wherein a plurality of resource blockers (40a-40d) are assigned to a shared resource (10a, 10b), which is involved in a conflicting combination of instruments and in a non-conflicting combination of instruments.

2. The method (100) according to claim 1, the method comprising:

executing a first test program portion (45a) configured to operate the first instrument (35a), the test program portion comprising:

checking a first set of resource blockers (40a, 40b) associated with the first instrument (35a),

blocking the first set of resource blockers (40a, 40b) if none of the resource blockers of the first set of resource blockers are blocked, and

selectively performing a test using a tester configuration defined by the first instrument (35a), if it was found in the checking step that none of the resource blockers of the first set of resource blockers are blocked;

executing a second test program portion (45b) configured to operate the second instrument (35b), the second test program portion comprising:

checking a second set of resource blockers (40c, 40d) associated with a second instrument (35b),

blocking the second set of resource blockers (40c, 40d) if none of the resource blockers of the second set of resource blockers are blocked, and

selectively performing a test using a tester configuration defined by the second instrument (35b), if it was found in the checking step that none of the resource blockers of the second set of resource blockers are blocked;

wherein the second test program portion (45b) is executed concurrently with the first test program portion (45a) or with a temporal overlap with the first test program portion (45a).

3. The method (100) according to claim 2, the method comprising:

waiting and checking the second set of resource blockers (40c, 40d) again if one of the resource blockers in the subset is blocked; or

executing a further part of the second test program portion (45b) and checking the second set of resource blockers (40c, 40d) again after the execution of the further part is finished.

4. The method (100) according to one of claims 2 or 3, the method comprising:

four different instruments (35a-35d), wherein the first and the second instrument (35a, 35b) constitute a non-conflicting combination of instruments using the first and the second subset of shared resources (10a, 10b), respectively, and wherein a third and a fourth instrument (35c, 35d) constitute a non-conflicting combination of instruments using both a combination of the first and the second subset of shared resources (10a, 10b), wherein the first and the second instrument (35a, 35b) are each in a conflict to both the third and the fourth instrument (35c, 35d);

four different resource blockers (40a-40d), wherein each of the resource blockers is assigned to one virtual shared resource (30a-30d), wherein a first and a second virtual shared resource (30a, 30b) are assigned to the first shared resource (10a) and a third and a fourth virtual shared resource (30c, 30d) are assigned to the second shared resource (10b), and wherein the first set of resource blockers (40a, 40b) comprises the first and the second virtual shared resource (30a, 30b), the second set of resource blockers (40c, 40d) comprises the third and the fourth virtual shared resource (30c, 30d), a third set of resource blockers (40a, 40c) comprises the first and the third virtual shared resource (30a, 30c), and a fourth set of resource blockers (40b, 40d) comprises the second and the fourth virtual shared resource (30b, 30d);

executing a third test program portion (45c) configured to operate the third instrument (35c), the third test program portion comprising

checking the third set of resource blockers (40a, 40c) associated with a third instrument (35c),

blocking the third set of resource blockers (40a, 40c) if none of the resource blockers on the third set of resource blockers is blocked, and

selectively performing a test using a tester configuration defined by the third instrument (35c) if it was found in the checking step that none of the resource blockers in the third set of resource blockers are blocked;

executing a fourth test program portion (45d) configured to operate the fourth instrument (35d), the fourth test program portion comprising

checking the fourth set of resource blockers (40b, 40d) associated with the fourth instrument (35d),

blocking the fourth set of resource blockers (40b, 40d) if none of the resource blockers of the fourth set of resource blockers is blocked, and

selectively performing a test using a tester configuration defined by the fourth instrument (35d), if it was found in the checking step that none of the resource blockers in the fourth set of resource blockers are blocked;

wherein at least two of the first, the second, the third and the fourth test program portions are executed concurrently or with a temporal overlap to one another.

5. The method (100) according to one of claims 1 to 4, wherein the resource blockers are a bit or a plurality of bits in a storing unit to indicate a usage of the shared resources (10a, 10b).

6. The method (100) according to one of claims 1 to 5, wherein the resource blockers (40a-40d) are mutexes, semaphores, locks, or monitors.

7. The method (100) according to one of claims 1 to 6, wherein one of the instruments is a virtual device configured to use a signal (17a, 17b) which is based on one of the shared resources (10a, 10b) or configured to generate a signal (17c) which is based on a combination of signals of multiple shared resources (10a, 10b).

8. The method (100) according to one of claims 1 to 7, wherein a scenario is a set of instruments, which are used concurrently.

9. The method (100) according to one of claims 1 to 8, wherein one of the instruments is a one tone generator using one of the shared resources or a two tone generator using a combination of two of the shared resources (10a, 10b).

10. The method (100) according to one of claims 1 to 9, wherein the shared resources (10a, 10b) are a physical stimulus device.

11. The method (100) according to one of claims 1 to 10, wherein the shared resources (10a, 10b) are a signal generator to generate a stimulus signal.

12. The method (100) according to one of claims 1 to 11, wherein the shared resources are a signal generator to generate radio frequency signals.

13. The method (100) according to one of claims 1 to 12, wherein the shared resources are digital to analog converters.

14. Computer program having a program code for programming at least one of the methods according to one of the claims 1 to 13 when the computer program runs on a computer.

15. A test apparatus (5) comprising:

a plurality of shared resources (10a, 10b), which can be used in different instruments (35a-35d);

wherein a first set of resource blockers (40a, 40b) is blocked when a first instrument (35a), which requires a first subset of the shared resources (10a), is to be executed;

wherein it is tried to block a second set of resource blockers (40c, 40d), when a second instrument (35b), which requires a second subset of the shared resources (10b) to be executed, wherein the first set of resource blockers is different from the second set of resource blockers; and

wherein a plurality of resource blockers (40a-40d) are assigned to a shared resource, which is involved in a conflicting combination of instruments and a non-conflicting combination of instruments.

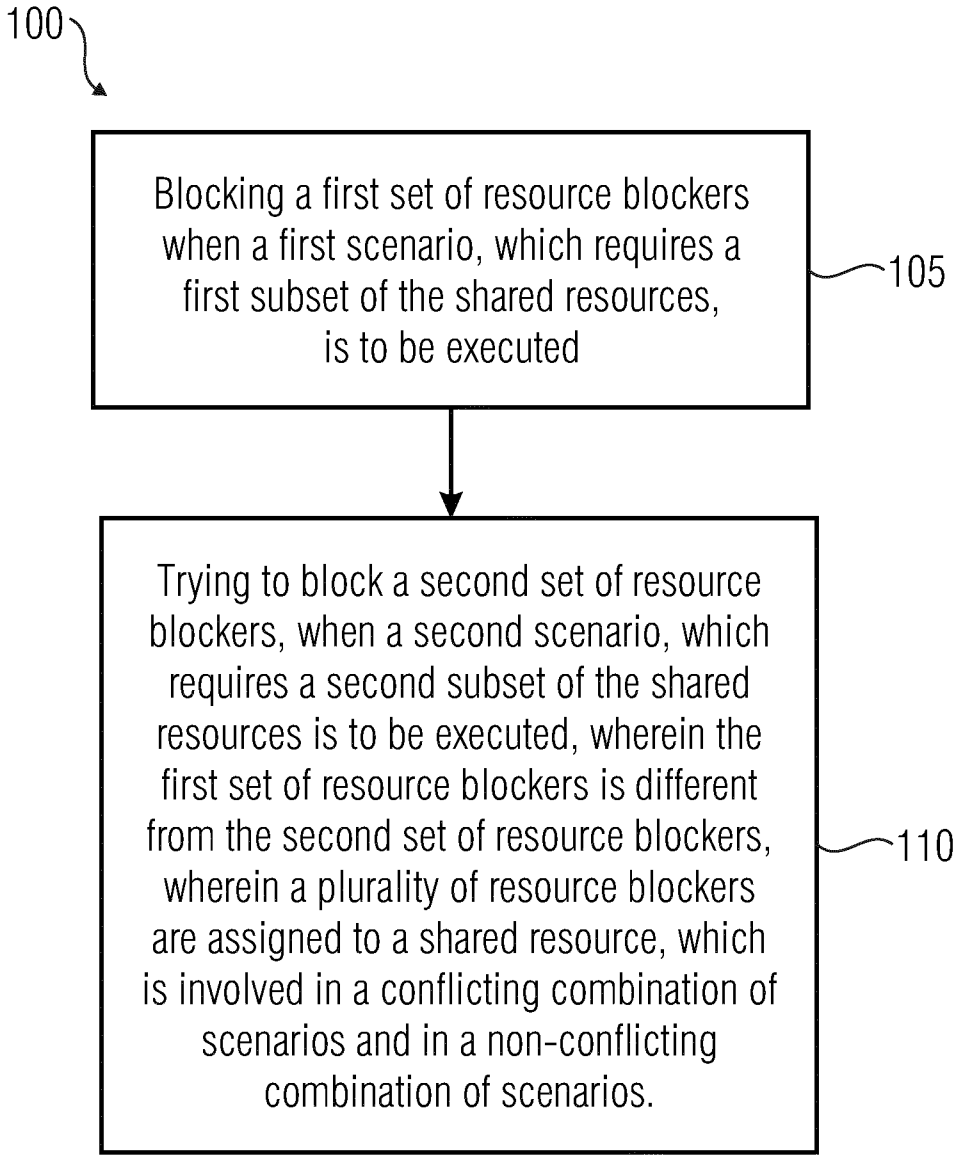

100
Blocking a first set of resource blockers when a first scenario, which requires a first subset of the shared resources, is to be executed
105
Trying to block a second set of resource blockers, when a second scenario, which requires a second subset of the shared resources is to be executed, wherein the first set of resource blockers is different from the second set of resource blockers, wherein a plurality of resource blockers are assigned to a shared resource, which is involved in a conflicting combination of scenarios and in a non-conflicting combination of scenarios.
110

FIG 1

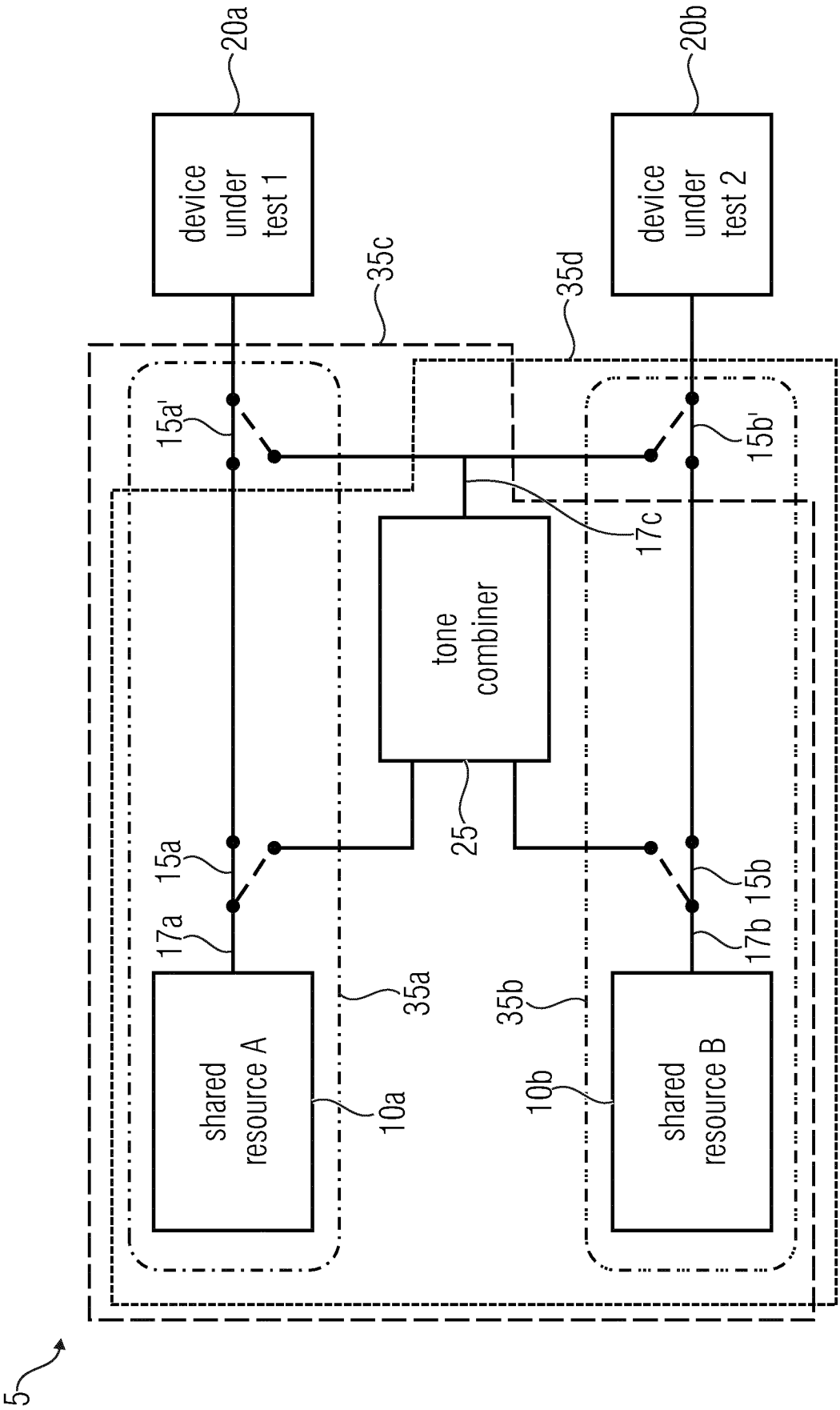

20a
20b
device under test 1
device under test 2
35c
35d
15a'
15b'
17c
tone combiner
25
15a
15b
17a
17b
35a
35b
shared resource A
shared resource B
10a
10b
5

FIG 2

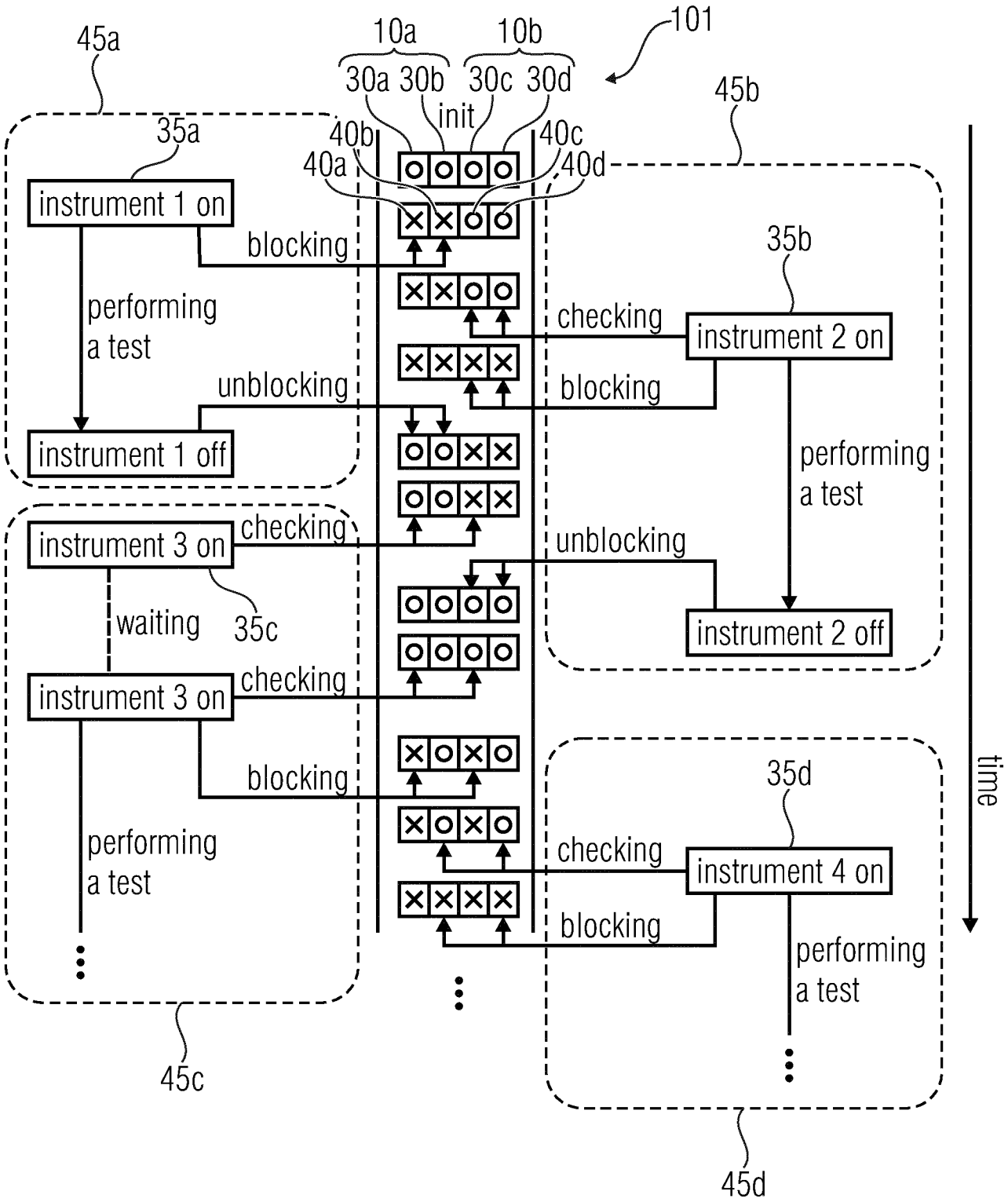

45a
10a
10b
101
30a 30b
30c 30d
init
45b
35a
40b
40a
40c
40d
instrument 1 on
blocking
35b
performing
a test
checking
instrument 2 on
unblocking
blocking
instrument 1 off
performing
a test
instrument 3 on
checking
unblocking
waiting
35c
instrument 2 off
instrument 3 on
checking
blocking
35d
performing
a test
checking
instrument 4 on
blocking
performing
a test
time
45c
45d

FIG 3

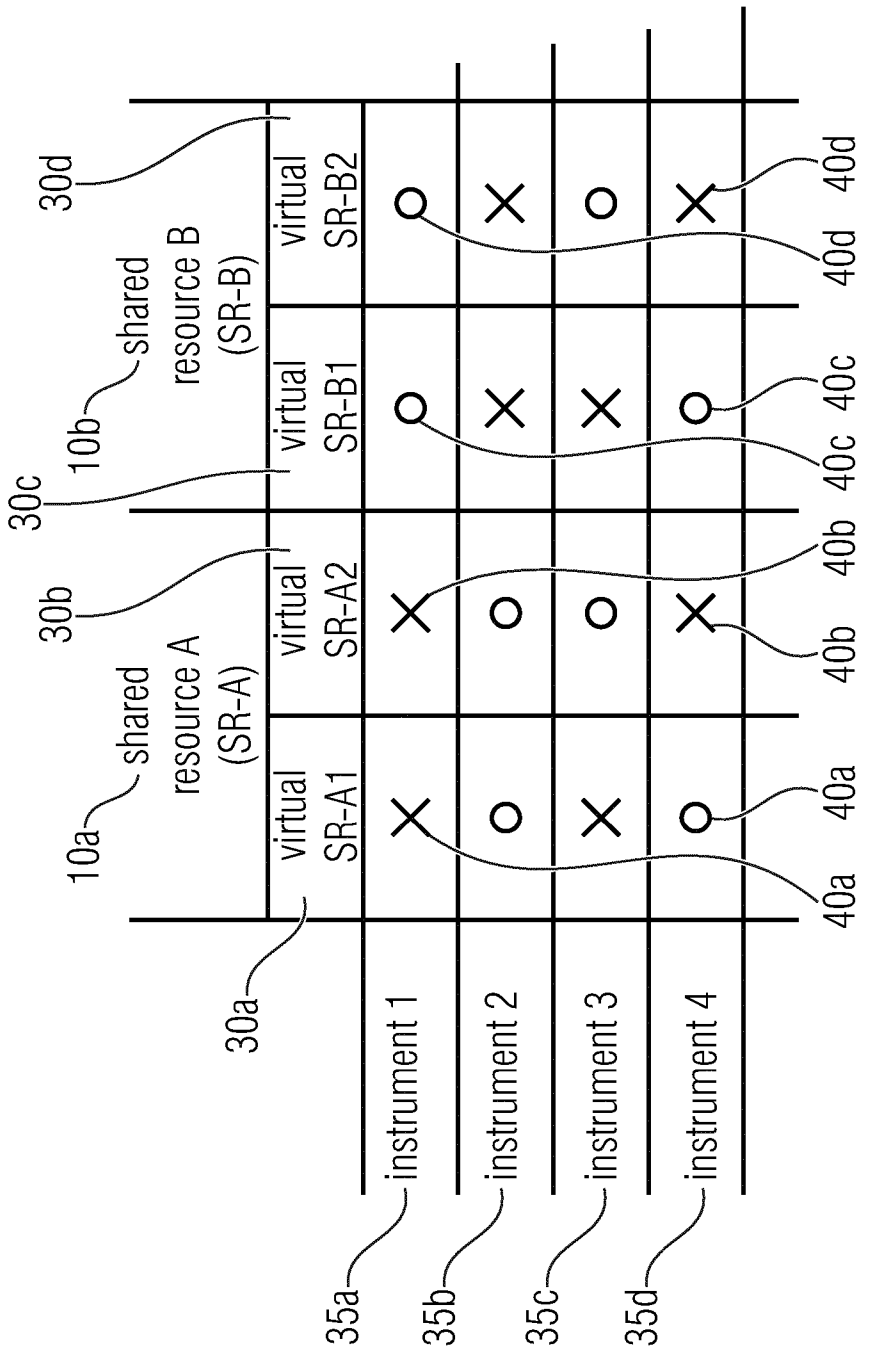

10a
shared resource A (SR-A)
10b
shared resource B (SR-B)
30a
30b
30c
30d
virtual SR-A1
virtual SR-A2
virtual SR-B1
virtual SR-B2
instrument 1
instrument 2
instrument 3
instrument 4
35a
35b
35c
35d
40a
40b
40c
40d

FIG 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/057273

A. CLASSIFICATION OF SUBJECT MATTER
INV. G01R31/319
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G01R

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	LEI LIU ED - PASSERINI K ET AL: "Finite automata for evaluating testbed resource contention", WIRELESS TELECOMMUNICATIONS SYMPOSIUM, 2007. WTS 2007, IEEE, PISCATAWAY, NJ, USA, 26 April 2007 (2007-04-26), pages 1-6, XP031285843, ISBN: 978-1-4244-0696-8	1,15
A	the whole document -----	2-14
A	US 2013/006567 A1 (HORN WOLFGANG [DE]) 3 January 2013 (2013-01-03) paragraph [0030] - paragraph [0054] -----	1-15
	-/--	

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 November 2015	30/11/2015

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Meggyesi, Zoltán

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/057273

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	NIEUWOUDT I ET AL: "Algorithms for a shared resource scheduling problem in which some level of conflict is tolerable", JOURNAL OF SCHEDULING, KLUWER ACADEMIC PUBLISHERS, BO, vol. 15, no. 6, 29 September 2012 (2012-09-29), pages 681-702, XP035130398, ISSN: 1099-1425, DOI: 10.1007/S10951-012-0291-Z Chapter 1-4 -----	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2015/057273

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013006567 A1	03-01-2013	CN 102906579 A	30-01-2013
		DE 112009005340 T5	13-09-2012
		TW 201142558 A	01-12-2011
		US 2013006567 A1	03-01-2013
		WO 2011072724 A1	23-06-2011