(54) Title of the Invention: Tracing data from an asynchronous interface
Abstract Title: Tracing asynchronous data transfers

(57) Apparatus (10) for tracing data (24) from a data bus (20) in a first clock domain (12) operating at a first clock frequency (14) to a trace array (22) in a second clock domain (16) operating at a second clock frequency (18), wherein the first clock frequency is lower than the second clock frequency. The apparatus comprises; (i) change detector means (26) to detect a change of the data on the data bus in the first clock domain, (ii) trigger means (28) responsive to the change detector means (26) to send a trigger pulse (34) to the second clock domain, (iii) pulse synchronization means (30) on the second clock domain responsive to the trigger pulse to synchronize the trigger pulse to the second clock frequency of the second clock domain by a meta-stability latch (36), and (iv) data capture means (32) in the second clock domain responsive to the pulse synchronization means to capture data from the data bus and to store the captured data (25) in the trace array.

Fig. 1
3/4

Start

Change of data? Yes

S102

Send trigger pulse

Pulse sync? Yes

S106

Start capture

Data stable? Yes

S110

Capture data

Store in trace array

End

S112

S113

S114

Fig. 3
The following terms are registered trade marks and should be read as such wherever they occur in this document:

Java
DESCRIPTION

Tracing data from an asynchronous interface

The present invention relates in general to data processing systems, and in particular, to an apparatus for tracing data from an asynchronous interface for debugging purposes.

BACKGROUND

Asynchronous data transfers have become very common in many integrated circuit devices, such as application-specific integrated circuits (ASICs) and system-on-chips (SoCs). In particular, various components or subsystems utilized for the construction of an integrated circuit may independently operate at different frequencies, such as in microprocessors and microcontrollers, where certain components or subsystems have a faster rate of operation than the operating frequencies of other system components or subsystems. Therefore, typically, it is desirable to devise integrated circuits with the ability to support multiple domains, which may operate at different frequencies.

For instance, many integrated circuits include a number of electronic circuits referred to as "clocked logic domains" that operate independently based on electrical "timing" or "clock" signals. Such clock signals are used to control and coordinate the activities of various components or subsystems. Since there will not be a fixed relationship between the active edge of a launch clock and a capture clock, there is a possibility of having setup or hold violations in the capture flip-flop, causing meta-stability. To avoid meta-stability in asynchronous data transfer, a commonly adopted technique is to double latch (also called double stage synchronization, or double flopping) the clock domain crossing signal at the receive domain clock.
frequency. Double flopping involves passing an asynchronous signal through a pair of edge triggered D-Flip-flops or some equivalent storage element. If the receiving clock frequency is considerably less than the transmitting clock frequency, there is a huge latency involved in the double flopping process, often up to 20 or more clock cycles in the higher frequency domain. This situation frequently arises with slower devices, like a Flash Memory controller, being used in ASICs that have a majority of the components running at a much higher clock frequency. Any reduction in the clock domain-crossing overhead significantly reduces the data transfer latencies and increases the overall system performance.

A variety of devices and methods are used in conjunction with the use of debug trace data in a system. The trace data can be passed between several different components in the circuit. Established protocols allow the different components to communicate with each other. One example involves a microprocessor circuit system. As the demand for more powerful and/or faster systems increases, design constraints, such as power consumption and heat dissipation, can become increasingly problematic.

US 8132036 B2 discloses a method and an interfacing circuit for transmitting data between a first clock domain operating at a first clock frequency C1 and a second clock domain operating at a second clock frequency C2. In accordance with this, data is transmitted from the first domain, through an interfacing circuitry, and to the second domain. The interfacing circuitry includes a synchronization section that operates at a third frequency C3, wherein C3 is a whole number multiple of C2. For example, C3 may be an even whole number multiple of C2.

Thus in US 8132036 B2, a clock signal A is used to operate the second clock domain at frequency C2, and a clock signal B is
used to operate the second section of the interfacing circuitry at frequency C3. Each of the clock signals A and B have regular, active edge portions, and each occurrence of one of the active edge portions of clock signal A is clock aligned with one of the active edge portions of clock signal B. Clock signals A and B are source synchronized.

In US 8132036 B2, the synchronization section of the interfacing circuitry includes first and second registers. Clock signal B is applied to both the first and second registers to operate these registers at frequency C3. In an asynchronous data transfer, the higher clock frequency that launches data may be C1 and the lower clock frequency that captures data may be C2. In accordance with this, the interface flip-flops used for double flopping run at a higher source synchronous clock frequency C3. C3 is source synchronized with the low frequency clock C2, and C3 and C2 will always have a common active edge and will be considered synchronous.

SUMMARY

It is an objective of the invention to provide a hardware efficient apparatus for tracing data from an asynchronous debugging interface.

A further objective is to provide a method for tracing data from an asynchronous debugging interface with a hardware efficient apparatus.

Another objective is to provide a data processing system for executing the method for tracing data from an asynchronous debugging interface with a hardware efficient apparatus.

These objectives are achieved by the features of the independent
claims. The other claims, the drawings and the specification disclose advantageous embodiments of the invention.

According to a first aspect of the invention an apparatus is proposed for tracing data from a data bus in a first clock domain operating at a first clock frequency to a trace array in a second clock domain operating at a second clock frequency, wherein the first clock frequency is lower than the second clock frequency. The apparatus comprises (i) change detector means to detect a change of the data on the data bus in the first clock domain, (ii) trigger means responsive to the change detector means to send a trigger pulse to the second clock domain, (iii) pulse synchronization means on the second clock domain responsive to the trigger pulse to synchronize the trigger pulse to the second clock frequency of the second clock domain by a meta-stability latch, (iv) data capture means in the second clock domain responsive to the pulse synchronization means to capture data from the data bus and to store the captured data in the trace array.

The meta-stability latch, the pulse synchronization means and the data capture means are implemented as the essential tracing interface. The data capture means are further implemented as a latch bank.

The inventive apparatus allows for tracing data from a data bus, which may preferably be implemented as a debug interface operating asynchronously in one clock domain at a first clock frequency, to another clock domain operating at a higher clock frequency. The data bus, e.g., may be operating in a first clock domain with an 'aclk' cycle, which is asynchronous to a core clock 'nclk' at a second clock frequency in a second clock domain. The data bus may be changing its value not faster than every four 'aclk' cycles. This data has to be traced in a trace array inside the second clock domain which, e.g., may be
operating with a '2 x nclk' cycle, synchronous to the core clock 'nclk'. Assumptions for the clock frequencies expressed in cycles may be that 'aclk' is less than 'nclk' cycle time. Because the data bus shall only be traced in a way, such that no data value is lost, the data value may be traced in one or more '2 x nclk' cycles. The advantage of the inventive apparatus is that every 'new' data value is written at least one time into the trace array and that it is written until the next 'new' data value is available to be written.

Advantages of the inventive apparatus are that no buffer array is needed for tracing the data from the data bus and that no handling of asynchronously running read and/or write pointers are necessary. A further advantage is that data and control paths are only going in one direction from the first clock domain to the second clock domain.

The inventive apparatus is implemented in a hardware system where every 'new' data is written once in the trace array, no matter how long it is available on the data bus. Thus each 'new' data need to be read only once. The clocks of the first clock domain with the data bus used for debugging and the second clock domain with the trace array for storing the data to be traced may run completely asynchronously to each other. The data on the data bus are read each time they may change. The phase of these data does not need to be known. The distance between the first clock domain and the second clock domain does not need to be known. The apparatus is able to trace the data on the data bus in stable conditions so that no ambiguities may result concerning previous or succeeding data. Bit width of the data bus is not limited in the inventive apparatus.

According to an advantageous embodiment the data may change its value not faster than every four cycles of the first clock frequency. In this embodiment the second clock frequency may be
a core clock frequency of a computer system or computer net where the apparatus is operating. The core clock frequency, e.g., could exhibit a value of 4 GHz. Then the data on the data bus may be traced to the data capture means with the core clock frequency and written to the trace array with half of the core clock frequency, e.g., 2 GHz in order to guarantee stable conditions with the data to be traced.

Advantageously in this embodiment data capturing with the pulse synchronization means, being part of the tracing interface, may be operating at the second clock frequency, wherein writing the captured data into the trace array is performed at half of the second clock frequency. Thus it is possible to check in a secure way if the data traced on the data bus have been changed or not in order to write the data in the trace array and do not get intermediate data just in the state of being changed on the data bus.

Favorably the inventive apparatus may be configured so that the data and the trigger pulse are flowing from the first clock domain to the second clock domain. This feature represents a major advantage because the synchronization of the data only needs to be executed once and no further signals or pointers need to be sent from one clock domain to the other clock domain. This saves timing resources as well as signal lines and/or bandwidth and also verification effort.

In an advantageous embodiment the change detector means may comprise an EXCLUSIVE-OR (XOR) circuit. This represents a quite efficient way of detecting data changes due to an EXCLUSIVE-OR (XOR) reduce circuit. Thus timing benefits as well as hardware resource benefits may be implemented due to the inventive embodiment.

Favorably the pulse synchronization means may comprise a
programmable delay for synchronization of the data capture means. By applying such a programmable delay on tracing the data on the data bus, stable conditions may be implemented for ensuring that the data being traced are stable and do not represent intermediate states due to a change from a previous to a succeeding data pattern.

Advantageously the apparatus may be configured so that data capturing into the data capture means is controllable by the programmable delay of the pulse synchronization means. Thus tracing the data from the data bus may avoid to get intermediate data patterns which are just in the way of changing and/or not defined at all. The programmable delay may be optimized for the actual clock frequencies of the first clock domain and the second clock domain.

In a favorable embodiment the apparatus may be configured so that the data capture means allow a direct capture of tracing data from the first clock domain. Advantageously the data may be written in a direct way from the data bus to the trace array, because there are no buffers or registers in the apparatus where the data have to be transferred or shifted in order to be written to the trace array.

According to a further advantageous aspect of the invention a method is proposed for tracing data from a data bus in a first clock domain operating at a first clock frequency to a trace array in a second clock domain operating at a second clock frequency by capturing a change of the data on the data bus, wherein the first clock frequency is lower than the second clock frequency, comprising (i) detecting a change of the data on the data bus of the first clock domain by means of change detector means, (ii) sending a trigger pulse to the second clock domain by means of trigger means responsive to the change detector means, (iii) synchronizing the trigger pulse to the second clock
frequency of the second clock domain by a meta-stability latch by means of pulse synchronization means of the second clock domain, the pulse synchronization means being responsive to the trigger pulse, (iv) capturing data from the data bus and to store the captured data in the trace array by means of data capture means on the second clock domain, the data capture means being responsive to the pulse synchronization means.

The method according to the invention may be executed on an apparatus as described above in order to trace data from a data bus, which may preferably be implemented as a debug interface. The inventive method may be executed on the apparatus where every 'new' data is written once in the trace array, no matter how long it is available on the data bus. Thus each 'new' data need to be read only once. The clocks of the first clock domain with the data bus used for debugging and the second clock domain with the trace array for storing the data to be traced may run completely asynchronously to each other. The data on the data bus may be read each time they may change. The phase of these data does not need to be known. The method may be able to trace the data on the data bus in stable conditions so that no ambiguities may result concerning previous or succeeding data.

Advantageously the inventive method may be executed when the data change its value not faster than every four cycles of the first clock frequency. In this embodiment the second clock frequency may be a core clock frequency of a computer system where the method is executed on. Then the data on the data bus may be traced with the core clock frequency in the tracing interface, i.e. captured in the data capture means, and written to the trace array with half of the core clock frequency in order to guarantee stable conditions with the data to be traced.

Favorably the method may comprise that data capturing with the pulse synchronization means, being part of the tracing
interface, is operating at the second clock frequency, wherein writing the captured data into the trace array is operating at half of the second clock frequency. Thus it is possible to check in a secure way if the data traced on the data bus have been changed or not in order to write the data in the trace array and do not get intermediate data just in the state of being changed on the data bus.

Advantageously the method may further include that the pulse synchronization means comprise a programmable delay for synchronization of the data capture means. By applying such a programmable delay on tracing the data on the data bus, stable conditions may be implemented for ensuring that the data being traced are stable and do not represent intermediate states due to a change from a previous to a succeeding data pattern.

According to a further advantageous aspect of the invention a data processing program for execution in a data processing system is proposed comprising an implementation of an instruction set for performing a method as described above when the data processing program is run on a computer.

Further a favorable computer program product is proposed comprising a computer usable medium including a computer readable program, wherein the computer readable program when executed on a computer causes the computer to perform a method for tracing data from a data bus in a first clock domain operating at a first clock frequency to a trace array in a second clock domain operating at a second clock frequency, wherein the first clock frequency is lower than the second clock frequency, comprising (i) detecting a change of the data on the data bus of the first clock domain by means of change detector means, (ii) sending a trigger pulse to the second clock domain by means of trigger means responsive to the change detector means, (iii) synchronizing the trigger pulse to the second clock frequency.
frequency of the second clock domain by a meta-stability latch by means of pulse synchronization means of the second clock domain, the pulse synchronization means being responsive to the trigger pulse, (iv) capturing data from the data bus and to store the captured data in the trace array by means of data capture means on the second clock domain, the data capture means being responsive to the pulse synchronization means.

As will be appreciated by one skilled in the art, aspects of the present invention may be embodied as a system, method or computer program product. Accordingly, aspects of the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system."

Furthermore, aspects of the present invention may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-
ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device. A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as part of a carrier wave. Such a propagated signal may take any of a variety of forms, including, but not limited to, electro-magnetic, optical, or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device.

Program code embodied on a computer readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wire connection, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

Computer program code for carrying out operations for aspects of the present invention may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection
may be made to an external computer (for example, through the Internet using an Internet Service Provider).

Aspects of the present invention are described below with reference to block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

These computer program instructions may also be stored in a computer readable medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article of manufacture including instructions which implement the function/act specified in the block diagram block or blocks.

The computer program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatus or other devices to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the block diagram block or blocks.
Due to a further aspect of the invention, a data processing system for execution of a data processing program is proposed, comprising software code portions for performing a method described above.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

The present invention together with the above-mentioned and other objects and advantages may best be understood from the following detailed description of the embodiments, but not restricted to the embodiments, wherein is shown in:

Fig. 1 an apparatus for tracing data from a data bus in a first clock domain operating at a first clock frequency to a trace array in a second clock domain operating at a second clock frequency according to a preferred embodiment of the invention;

Fig. 2 a timing diagram for tracing data from a data bus in a first clock domain to a trace array in a second clock domain according to the invention;

Fig. 3 a flowchart for tracing data from a data bus in a first clock domain to a trace array in a second clock domain according to the invention; and

Fig. 4 an example embodiment of a data processing system for carrying out the method for tracing data from a data bus in a first clock domain to a trace array in a second clock domain by capturing a change of data on the data bus according to the invention.
DETAILED DESCRIPTION

In the drawings, like elements are referred to with equal reference numerals. The drawings are merely schematic representations, not intended to portray specific parameters of the invention. Moreover, the drawings are intended to depict only typical embodiments of the invention and therefore should not be considered as limiting the scope of the invention.

Figure 1 depicts an apparatus 10 for tracing data 24 from a data bus 20 in a first clock domain 12 operating at a first clock frequency 14 to a trace array 22 in a second clock domain 16 operating at a second clock frequency 18 according to a preferred embodiment of the invention. The apparatus 10 is used for tracing data 24 from a data bus 20 by capturing a change of the data 24 on the data bus 12. The apparatus 10 comprises (i) change detector means 26 to detect a change of the data 24 on the data bus 20 in the first clock domain 12, (ii) trigger means 28 responsive to the change detector means 26 to send a trigger pulse 34 to the second clock domain 16, (iii) pulse synchronization means 30 on the second clock domain 16 responsive to the trigger pulse 34 to synchronize the trigger pulse 34 to the second clock frequency 18 of the second clock domain 16 by a meta-stability latch 36, as well as (iv) data capture means 32 in the second clock domain 16 responsive to the pulse synchronization means 30 to capture data 24 from the data bus 20 and to store the captured data 25 in the trace array 22.

The first clock frequency 14 hereby is lower than the second clock frequency 18. In the embodiment shown the second clock frequency may be a core clock frequency of a computer system or computer net of, e.g., 4 GHz. Then the pulse synchronization and data capture means 42 could be operating at the second clock frequency 18, e.g., 4 GHz, wherein writing the data into the trace array could be performed at half of the second clock
frequency 18, e.g., 2 GHz. The apparatus 10 is configured so that the data 24 and the trigger pulse 34 are flowing from the first clock domain 12 to the second clock domain 16. The change detector means 26 comprise an EXCLUSIVE-OR (XOR) circuit 38, whereas the pulse synchronization means 30 comprise a programmable delay 40 for synchronization of the data capture means 32. The apparatus 10 is configured so that data capturing into the data capture means 32 is controllable by the programmable delay 40 of the pulse synchronization means 30. Further the apparatus 10 is configured so that the data capture means 32 allow a direct capture of tracing data 24 from the first clock domain 12.

The inventive apparatus 10 solves the problem of tracing data 24 from a data bus 20, which may preferably be implemented as a debug interface operating asynchronously in one clock domain 12, to another clock domain 16 operating at a different clock frequency 18. As is illustrated in Figure 2, the data bus 20, e.g., may be operating in the first clock domain 12 with an 'aclk' cycle, which is asynchronous to a core clock 18 with an 'nclk' cycle. The data bus 20 may be changing its value every 4 'aclk' cycles. This data 24 has to be traced in a trace array 22 inside the second clock domain 16 which, e.g., may be operating with a '2 x nclk' cycle, synchronous to the fast core clock 18. Assumptions for the clock frequencies 14, 18 expressed in cycles may be that 'aclk' cycle time is less than 'nclk' cycle time.

Because the data bus 20 shall only be traced in a way, such that no data value is lost, the data value may be traced in one or more '2 x nclk' cycles. The advantage of the inventive apparatus 10 is that every 'new' data value is written at least one time into the trace array 22 and that it is written until the next 'new' data value is available to be written.

Thus the method for tracing data 24 from a data bus 20 by capturing a change of the data 24 on the data bus 20 comprise
the steps (i) detecting a change of the data 24 on the data bus 20 of the first clock domain 12 by means of change detector means 26, (ii) sending a trigger pulse 34 to the second clock domain 16 by means of trigger means 28 responsive to the change detector means 26, (iii) synchronizing the trigger pulse 34 to the second clock frequency 18 of the second clock domain 16 by a meta-stability latch 36 by means of pulse synchronization means 30 of the second clock domain 16, the pulse synchronization means 30 being responsive to the trigger pulse 34, and (iv) capturing data 24 from the data bus 20 and to store the captured data 25 in the trace array 22 by means of data capture means 32 on the second clock domain 16, the data capture means 32 being responsive to the pulse synchronization means 30.

The data 24, available on the data bus 20 in the first clock domain 12, are fed to the change detector means 26 realized as the EXCLUSIVE-OR (XOR) circuit in the embodiment shown in Figure 1, via a data detection line 50 with the data 24 from the data bus 20. Parallel the data 24 are fed to the latch 46, where after the latch 46 a second data detection line 52 with the data 24 from the data bus 20 is fed to the change detector means 26. Thus it can be detected if the data 24 on the data bus 20 are changed. The output of the change detector means 26 are fed to the trigger means 28, another latch, where a trigger pulse 34 is generated and this trigger pulse 34 as a new data available signal fed through the domain boundary 44 to the second clock domain 16 into the meta-stability latch 36, being part of the tracing interface 42, operating at a frequency half the second clock frequency 18. The trigger pulse 34 thus is synchronized to the second clock frequency 18 of the second clock domain 16 by starting the pulse synchronization via the start pulse synchronization signal 62 in the pulse synchronization means 30, which is implemented with a programmable delay 40. From the pulse synchronization means 30 the trigger pulse 34 as a data capture signal 54 is fed to the data capture means 32 where the
data 24 from the data bus 20 arrived through the latch 48. If the data capture signal 54 as a new data stable signal is fed to the data capture means 32, the data 25 as a data bus trace are sent to the trace array 22 and stored.

In Figure 2 a timing diagram for tracing data 24 from a data bus 20 in a first clock domain 12 to a trace array 22 in a second clock domain 16 according to the invention is shown. The data 24 on the data bus 20, named A, B, C, D in Figure 2, are available on the data bus 20 in the first clock domain 12 with a first clock frequency 14, which is represented in Figure 2 as a short mark. The data 24 may change on the data bus 20 every four cycles, represented as a longer mark 56. If a change in the data 24 is detected by the change detector means 26, a trigger pulse 34 will be sent to the second clock domain 16 and a pulse synchronization of the trigger pulse 34 to the second clock frequency 18 will be executed in order to generate a data capture signal 54, signaling that the new data 25 are valid and may be captured into the trace array 22. The second clock domain is operating with a second clock frequency 18, e.g., a core clock of a computer system or computer net. The pulse synchronization is executed with a pulse synchronization frequency 58 of the second clock frequency 18, whereas writing the captured data 25 into the trace array 22 is executed with a data capture frequency 60 of half of the second clock frequency 18, represented as a long mark. The change of the traced data 25 is marked with 25a, 25b, 25c in Figure 2.

Figure 3 depicts a flowchart for tracing data 24 from a data bus 20 in a first clock domain 12 to a trace array 22 in a second clock domain 16 according to the invention. The tracing process starts at step S100. Then the change detector means 26 are continuously checking in step S102 if there are new data 24 available on the data bus 20. If this is the case a trigger pulse 34 is generated by the trigger means 28 in step S104. This
trigger pulse 34 is sent to the meta-stability latch 36 in order to perform pulse synchronization in the pulse synchronization means 30 after having started the pulse synchronization by the start pulse synchronization signal 62. If this step S106 is successful (Yes) data capturing is started in step S108. If the data capture in step S110 is stable (Yes) the trigger pulse 34 as a data capture signal 54 is sent to the data capture means 32 and thus the data 24 are captured, step S112, and stored in the trace array 22 in step S113. This ends the tracing process in step S114.

Referring now to Figure 4, a schematic of an example of a data processing system 210 is shown. Data processing system 210 is only one example of a suitable data processing system and is not intended to suggest any limitation as to the scope of use or functionality of embodiments of the invention described herein. Regardless, data processing system 210 is capable of being implemented and/or performing any of the functionality set forth herein above.

The data processing system 210 is capable of running a computer program product comprising a computer usable medium including a computer readable program, wherein the computer readable program when executed on a computer system 212 causes the computer system 212 to perform a method for tracing data 24 from a data bus 20 in a first clock domain 12 operating at a first clock frequency 14 to a trace array 22 in a second clock domain 16 operating at a second clock frequency 18 by capturing a change of the data 24 on the data bus 20, wherein the first clock frequency 14 is lower than the second clock frequency 18, comprising (i) detecting a change of the data 24 on the data bus 20 of the first clock domain 12 by means of change detector means 26, (ii) sending a trigger pulse 34 to the second clock domain 16 by means of trigger means 28 responsive to the change detector means 26, (iii) synchronizing the trigger pulse 34 to
the second clock frequency 18 of the second clock domain 16 by a
meta-stability latch 36 by means of pulse synchronization means
30 of the second clock domain 16, the pulse synchronization
means 30 being responsive to the trigger pulse 34, (iv)
capturing data 24 from the data bus 20 and to store the captured
data 25 in the trace array 22 by means of data capture means 32
on the second clock domain 16, the data capture means 32 being
responsive to the pulse synchronization means 30.

In data processing system 210 there is a computer system/server
212, which is operational with numerous other general purpose or
special purpose computing system environments or configurations.
Examples of well-known computing systems, environments, and/or
configurations that may be suitable for use with computer
system/server 212 include, but are not limited to, micro-
controllers, personal computer systems, server computer systems,
thin clients, thick clients, handheld or laptop devices,
multiprocessor systems, microprocessor-based systems, set top
boxes, programmable consumer electronics, network PCs,
minicomputer systems, mainframe computer systems, and
distributed cloud computing environments that include any of the
above systems or devices, and the like.

Computer system/server 212 may be described in the general
context of computer system executable instructions, such as
program modules, being executed by a computer system.
Generally, program modules may include routines, programs,
objects, components, logic, data structures, and so on that
perform particular tasks or implement particular abstract data
types. Computer system/server 212 may be practiced in
distributed cloud computing environments where tasks are
performed by remote processing devices that are linked through a
communications network. In a distributed cloud computing
environment, program modules may be located in both local and
remote computer system storage media including memory storage
devices.

As shown in Fig. 4, computer system/server 212 in data processing system 210 is shown in the form of a general-purpose computing device. The components of computer system/server 212 may include, but are not limited to, one or more processors or processing units 216, a system memory 228, and a bus 218 that couples various system components including system memory 228 to processor 216. Bus 218 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus.

Computer system/server 212 typically includes a variety of computer system readable media. Such media may be any available media that is accessible by computer system/server 212, and it includes both volatile and non-volatile media, removable and non-removable media.

System memory 228 can include computer system readable media in the form of volatile memory, such as random access memory (RAM) 230 and/or cache memory 232. Computer system/server 212 may further include other removable/non-removable, volatile/non-volatile computer system storage media. By way of example only, storage system 234 can be provided for reading from and writing to a non-removable, non-volatile magnetic media (not shown and typically called a "hard drive"). Although not shown, a magnetic disk drive for reading from and writing to a removable, non-volatile magnetic disk (e.g., a "floppy disk"), and an optical disk drive for reading from or writing to a removable, non-
volatile optical disk such as a CD-ROM, DVD-ROM or other optical media can be provided. In such instances, each can be connected to bus 218 by one or more data media interfaces. As will be further depicted and described below, memory 228 may include at least one program product having a set (e.g., at least one) of program modules that are configured to carry out the functions of embodiments of the invention.

Program/utility 240, having a set (at least one) of program modules 242, may be stored in memory 228 by way of example, and not limitation, as well as an operating system, one or more application programs, other program modules, and program data.

Each of the operating system, one or more application programs, other program modules, and program data or some combination thereof, may include an implementation of a networking environment. Program modules 242 generally carry out the functions and/or methodologies of embodiments of the invention as described herein.

Computer system/server 212 may also communicate with one or more external devices 214 such as a keyboard, a pointing device, a display 224, etc.; one or more devices that enable a user to interact with computer system/server 212; and/or any devices (e.g., network card, modem, etc.) that enable computer system/server 212 to communicate with one or more other computing devices. Such communication can occur via Input/Output (I/O) interfaces 222. Still yet, computer system/server 212 can communicate with one or more networks such as a local area network (LAN), a general wide area network (WAN), and/or a public network (e.g., the Internet) via network adapter 220. As depicted, network adapter 220 communicates with the other components of computer system/server 212 via bus 218. It should be understood that although not shown, other hardware and/or software components could be used in conjunction with computer
system/server 212. Examples, include, but are not limited to: microcode, device drivers, redundant processing units, external disk drive arrays, RAID systems, tape drives, and data archival storage systems, etc.

The block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments of the present invention. In this regard, each block in the block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical functions. It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams, and combinations of blocks in the block diagrams, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.
CLAIMS

1. Apparatus (10) for tracing data (24) from a data bus (20) in a first clock domain (12) operating at a first clock frequency (14) to a trace array (22) in a second clock domain (16) operating at a second clock frequency (18), wherein the first clock frequency (14) is lower than the second clock frequency (18), comprising
   (i) change detector means (26) to detect a change of the data (24) on the data bus (20) in the first clock domain (12),
   (ii) trigger means (28) responsive to the change detector means (26) to send a trigger pulse (34) to the second clock domain (16),
   (iii) pulse synchronization means (30) on the second clock domain (16) responsive to the trigger pulse (34) to synchronize the trigger pulse (34) to the second clock frequency (18) of the second clock domain (16) by a meta-stability latch (36),
   (iv) data capture means (32) in the second clock domain (16) responsive to the pulse synchronization means (30) to capture data (24) from the data bus (20) and to store the captured data (25) in the trace array (22).

2. Apparatus according to claim 1, wherein the data (24) change its value not faster than every four cycles of the first clock frequency (14).

3. Apparatus according to claim 1 or 2, wherein data capturing with the pulse synchronization means (30) is operating at the second clock frequency (18), wherein writing the captured data (24) into the trace array (22) is performed at half of the second clock frequency (18).
4. Apparatus according to any one of the preceding claims, being configured so that the data (24) and the trigger pulse (34) are flowing from the first clock domain (12) to the second clock domain (16).

5. Apparatus according to any one of the preceding claims, wherein the change detector means (26) comprise an EXCLUSIVE-OR (XOR) circuit (38).

6. Apparatus according to any one of the preceding claims, wherein the pulse synchronization means (30) comprise a programmable delay (40) for synchronization of the data capture means (32).

7. Apparatus according to claim 4, being configured so that data capturing into the data capture means (32) is controllable by the programmable delay (40) of the pulse synchronization means (30).

8. Apparatus according to any one of the preceding claims, being configured so that the data capture means (32) allow a direct capture of tracing data (24) from the first clock domain (12).

9. Method for tracing data (24) from a data bus (20) in a first clock domain (12) operating at a first clock frequency (14) to a trace array (22) in a second clock domain (16) operating at a second clock frequency (18), wherein the first clock frequency (14) is lower than the second clock frequency (18), comprising
   (i) detecting a change of the data (24) on the data bus (20) of the first clock domain (12) by means of change detector means (26),
(ii) sending a trigger pulse (34) to the second clock domain (16) by means of trigger means (28) responsive to the change detector means (26),

(iii) synchronizing the trigger pulse (34) to the second clock frequency (18) of the second clock domain (16) by a meta-stability latch (36) by means of pulse synchronization means (30) of the second clock domain (16), the pulse synchronization means (30) being responsive to the trigger pulse (34),

(iv) capturing data (24) from the data bus (20) and to store the captured data (25) in the trace array (22) by means of data capture means (32) on the second clock domain (16), the data capture means (32) being responsive to the pulse synchronization means (30).

10. Method according to claim 9, wherein the data (24) change its value not faster than every four cycles of the first clock frequency (14).

11. Method according to claim 9 or 10, wherein data capturing with the pulse synchronization means (30) is operating at the second clock frequency (18), wherein writing the captured data (24) into the trace array (22) is performed at half of the second clock frequency (18).

12. Method according to claim 9 to 11, wherein the pulse synchronization means (30) comprise a programmable delay (40) for synchronization of the data capture means (32).

13. A data processing program (240) for execution in a data processing system (210) comprising an implementation of an instruction set for performing a method according to any one of the claims 9 to 12 when the data processing program (240) is run on a computer system (212).
14. A computer program product comprising a computer usable medium including a computer readable program, wherein the computer readable program when executed on a computer system (212) causes the computer system (212) to perform a method for tracing data (24) from a data bus (20) in a first clock domain (12) operating at a first clock frequency (14) to a trace array (22) in a second clock domain (16) operating at a second clock frequency (18), wherein the first clock frequency (14) is lower than the second clock frequency (18), comprising

(i) detecting a change of the data (24) on the data bus (20) of the first clock domain (12) by means of change detector means (26),

(ii) sending a trigger pulse (34) to the second clock domain (16) by means of trigger means (28) responsive to the change detector means (26),

(iii) synchronizing the trigger pulse (34) to the second clock frequency (18) of the second clock domain (16) by a meta-stability latch (36) by means of pulse synchronization means (30) of the second clock domain (16), the pulse synchronization means (30) being responsive to the trigger pulse (34),

(iv) capturing data (24) from the data bus (20) and to store the captured data (25) in the trace array (22) by means of data capture means (32) on the second clock domain (16), the data capture means (32) being responsive to the pulse synchronization means (30).

15. A data processing system (210) for execution of a data processing program (240) comprising software code portions for performing a method according to any one of the claims 9 to 12.
**Patents Act 1977: Search Report under Section 17**

**Documents considered to be relevant:**

<table>
<thead>
<tr>
<th>Category</th>
<th>Relevant to claims</th>
<th>Identity of document and passage or figure of particular relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>1, 4-6, 9, 10, 12-14</td>
<td>US2009/225915 A1 (HASSAN et al.), see abstract and Figure 3B</td>
</tr>
<tr>
<td>X</td>
<td>1, 4-6, 9, 10, 12-14</td>
<td>US2005/285640 A1 (NAKAMURA), see abstract and Figure 4</td>
</tr>
</tbody>
</table>

**Categories:**

<table>
<thead>
<tr>
<th>X</th>
<th>Document indicating lack of novelty or inventive step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Document indicating lack of inventive step if combined with one or more other documents of same category.</td>
</tr>
<tr>
<td>&amp;</td>
<td>Member of the same patent family</td>
</tr>
<tr>
<td>A</td>
<td>Document indicating technological background and/or state of the art.</td>
</tr>
<tr>
<td>P</td>
<td>Document published on or after the declared priority date but before the filing date of this invention.</td>
</tr>
<tr>
<td>E</td>
<td>Patent document published on or after, but with priority date earlier than, the filing date of this application.</td>
</tr>
</tbody>
</table>

**Field of Search:**

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC:

- G06F; H04L

Worldwide search of patent documents classified in the following areas of the IPC:

The following online and other databases have been used in the preparation of this search report:

- EPODOC, WPI, TXTE

**International Classification:**

<table>
<thead>
<tr>
<th>Subclass</th>
<th>Subgroup</th>
<th>Valid From</th>
</tr>
</thead>
<tbody>
<tr>
<td>G06F</td>
<td>0001/12</td>
<td>01/01/2006</td>
</tr>
<tr>
<td>G06F</td>
<td>0005/06</td>
<td>01/01/2006</td>
</tr>
<tr>
<td>G06F</td>
<td>0011/36</td>
<td>01/01/2006</td>
</tr>
<tr>
<td>G06F</td>
<td>0013/14</td>
<td>01/01/2006</td>
</tr>
<tr>
<td>G06F</td>
<td>0013/38</td>
<td>01/01/2006</td>
</tr>
</tbody>
</table>