

[USPTO PATENT FULL-TEXT AND IMAGE DATABASE](#)[Home](#)[Quick](#)[Advanced](#)[Pat Num](#)[Help](#)[Bottom](#)[View Cart](#)[Add to Cart](#)[Images](#)

(1 of 1)

United States Patent
Cox, Jr. , et al.**8,826,058**
September 2, 2014

Delay tolerant asynchronous interface (DANI)

Abstract

A Delay-tolerant Asynchronous Interface (DANI) is typically used to make the clock domains for reusable silicon intellectual property (IP) cores completely independent of each other. In fact, a DANI-wrapped IP core usually appears to its environment as if it were clockless. This property is necessary to address the variability in data transmission-time between source and destination. This variability is a result of increased lack of predictability in today's leading-edge manufacturing processes. A DANI wrapper can be applied to the IP core that is the source of data to be transmitted or it can be applied to the IP core that is the destination of that data. The transmission time over the route between source and destination may vary more than a single clock period.

Inventors: **Cox, Jr.; Jerome R.** (Sunset Hills, MO), **Engel; George** (Maryville, IL), **Moscola; James** (Red Lion, PA), **Chaney; Thomas J.** (Bridgeton, MO)

Applicant:

| Name | City | State | Country | Type |
|----------------------------|--------------|-------|---------|------|
| Cox, Jr.; Jerome R. | Sunset Hills | MO | US | |
| Engel; George | Maryville | IL | US | |
| Moscola; James | Red Lion | PA | US | |
| Chaney; Thomas J. | Bridgeton | MO | US | |

Assignee: *Blendics, Inc.* (St. Louis, MO)

Family ID: 51400181

Appl. No.: 14/025,677

Filed: September 12, 2013

Related U.S. Patent Documents

| <u>Application Number</u> | <u>Filing Date</u> | <u>Patent Number</u> | <u>Issue Date</u> |
|---------------------------|--------------------|----------------------|-------------------|
| 61701704 | Sep 16, 2012 | | |

Current U.S. Class: 713/400; 713/600; 713/401

Current CPC Class: G06F 1/12 (20130101); G06F 5/14 (20130101); G06F 5/06 (20130101); G06F 15/7825 (20130101); G06F 2205/102 (20130101)

Current International Class: G06F 1/04 (20060101); G06F 1/10 (20060101)

References Cited [\[Referenced By\]](#)

U.S. Patent Documents

| | | |
|------------------------------|---------------|----------------|
| 6850092 | February 2005 | Chelcea et al. |
| 7310396 | December 2007 | Sabih |
| 8559576 | October 2013 | Ono et al. |
| 2007/0097771 | May 2007 | Chu et al. |
| 2009/0019193 | January 2009 | Luk |
| 2009/0323876 | December 2009 | Ono et al. |

Other References

Quinton et al., "Practical Asynchronous Interconnect Network Design," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, May 2008, pp. 579-588, vol. 16, No. 5, IEEE, New York, NY. cited by applicant .

Santosh Sood, "A Novel Interleaved and Distributed FIFO," Thesis, Nov. 2005, The University of British Columbia, Vancouver, BC, CA (115 pages). cited by applicant.

Primary Examiner: Butler; Dennis M

Attorney, Agent or Firm: The Law Office of Kirk D. Williams

Parent Case Text

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application No. 61/701,704, filed Sep. 16, 2012, which is hereby incorporated by reference in its entirety.

Claims

What is claimed is:

1. An integrated circuit, comprising: a source wrapper configured to provide an asynchronous sending interface for a sending system on the integrated circuit, with the asynchronous sending interface configured to produce a write clock output signal and one or more data output signals, and configured to receive and react to flow control information; a destination wrapper configured to provide an asynchronous receiving interface for a receiving system on the integrated circuit and to produce said flow control information allowing a plurality of data words in flight between the source wrapper and the destination wrapper without an overflow loss in the destination wrapper, with the asynchronous receiving interface configured to receive a write clock input signal and one or more data input signals, wherein the destination wrapper includes an asynchronous first-in, first-out queue (aFIFO) providing an intermediate storage of information received on said data input signals from a first clock domain corresponding to the write clock input signal and provided to the receiving system in a different clock domain corresponding to a read clock received from the receiving system; and signal paths on the integrated circuit configured to communicatively couple the write clock output signal with the write clock input signal and said data output signals with said data input signals, and to provide a flow control signal path communicating said flow control information from the destination wrapper to the source wrapper, with said signal paths providing the write clock input signal and said data input signals with relative timing constraints applied between the write clock input signal and said data input signals.

2. The integrated circuit of claim 1, wherein the destination wrapper uses a unary code to specify locations within the aFIFO.

3. The integrated circuit of claim 1, wherein said flow control information includes token-based flow control information.
4. The integrated circuit of claim 1, wherein the aFIFO is configured to store a maximum of d data words; and wherein said flow control information allows for a maximum of d data words to be in flight between the source wrapper and the destination wrapper, wherein d is greater than one.
5. The integrated circuit of claim 1, wherein the sending system and receiving systems are operated on different clocks with one or more different operating clock rates.
6. The integrated circuit of claim 1, wherein said signal paths include no intervening pipeline stages.
7. The integrated circuit of claim 1, wherein the destination wrapper does not use a Gray code to determine locations within the aFIFO.
8. The integrated circuit of claim 1, wherein each of the sending and receiving systems is synchronous.
9. The integrated circuit of claim 1, wherein the source wrapper and destination wrapper are configured to communicate only using said signal paths.
10. The integrated circuit of claim 9, wherein the write clock output signal is a sending system clock gated with a sending system data available signal.
11. The integrated circuit of claim 1, wherein the source wrapper is co-located with the sending system, the destination wrapper is co-located with the receiving system, the destination wrapper is remote from the source wrapper.
12. The integrated circuit of claim 11, wherein the write clock output signal is a sending system clock gated with a sending system data available signal.
13. An integrated circuit, comprising: a source wrapper providing an asynchronous sending interface for a sending system on the integrated circuit, with the asynchronous sending interface producing a write clock output signal and a data output signal; a destination wrapper providing an asynchronous receiving interface for a receiving system on the integrated circuit, with the asynchronous receiving interface receiving a write clock input signal and data input signal wherein the destination wrapper includes an asynchronous first-in, first-out queue (aFIFO), and wherein the destination wrapper uses a unary code to specify locations within the aFIFO; and signal paths on the integrated circuit communicatively coupling the write clock output signal and the write clock input signal, and the data output signal and said data input signal, with the signal paths providing the write clock input signal and said data input signal with relative timing constraints applied between the write clock input signal and said data input signal.
14. A method, comprising: in response to receiving flow control information identifying that a destination wrapper can accept a plurality of data words, a source wrapper sending to the destination wrapper a plurality of data words such that at least two of the plurality of data words are overlapping in flight between the source and destination wrappers; wherein said sending a particular data word of the plurality of data words includes providing a write clock signal and a w -bits wide data signal, with w being an integer greater than zero; for each particular data word of the plurality of data words: receiving, by the destination wrapper, the write clock signal and the data signal with relative timing constraints maintained for said sent write clock and data signals; storing, by the destination wrapper, said particular data word communicated in said received data signal in an asynchronous first-in, first-out queue (aFIFO) according to a first clock domain corresponding to said received write clock signal; and receiving, by a receiving system on the integrated circuit, the particular data word from the aFIFO according to a second clock domain according to a read clock signal provided by the receiving system to the destination wrapper.
15. The method of claim 14, comprising receiving, by the source wrapper from a sending system on the integrated circuit, the data word according to the first clock domain according to a write clock signal provided by the sending system to the source wrapper.

globally-asynchronous, locally-synchronous (GALS) approach has been gaining in popularity to overcome this difficult architectural problem. The GALS approach is to partition a system design into decoupled clock-independent modules that can be designed to meet their individual requirements. These independent modules can then be coupled using an asynchronous interconnect network or an asynchronous network-on-chip (ANoC), which improves reliability by simplifying clock-domain crossing timing by using delay-tolerant connection modules. However, the complexity of such interconnect networks (measured in terms of the number of different ways control signals traverse such an interconnect network) grows exponentially instead of linearly as the number of independent control network elements used in implementing the interconnect network is increased. Therefore, providing a reliable interconnect network becomes problematic without a methodology to control this increased complexity.

BRIEF DESCRIPTION OF THE DRAWINGS

The appended claims set forth the features of one or more embodiments with particularity. The embodiment(s), together with its advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

FIG. 1A illustrates a sending system according to one embodiment;

FIG. 1B illustrates a receiving system according to one embodiment;

FIG. 2 illustrates a wrapper destination control according to one embodiment;

FIG. 3A illustrates a head of queue write address unit according to one embodiment;

FIG. 3B illustrates a tail of queue read address unit according to one embodiment;

FIG. 4 illustrates an asynchronous first-in, first-out queue (FIFO) according to one embodiment;

FIG. 5A illustrates a sending system according to one embodiment;

FIG. 5B illustrates a receiving system according to one embodiment;

FIG. 6 illustrates a wrapper source control according to one embodiment;

FIG. 7 illustrates a wrapper destination control according to one embodiment;

FIG. 8 illustrates a token-based flow control according to one embodiment;

FIG. 9A illustrates a an additional stage synchronization unit according to one embodiment; and

FIG. 9B illustrates an additional stage synchronization unit according to one embodiment.

DESCRIPTION OF EXAMPLE EMBODIMENTS

1. Overview

Disclosed are, inter alia, methods, apparatus, computer-storage media, mechanisms, and means associated with a delay-tolerant asynchronous interface. One embodiment includes an integrated circuit, comprising: a source wrapper providing an asynchronous sending interface to a sending system on the integrated circuit, with the asynchronous sending interface producing a write clock output signal and a data output signal; a destination wrapper providing an asynchronous receiving interface to a receiving system on the integrated circuit, with the asynchronous receiving interface receiving a write clock input signal and a data input signal; and signal paths on the integrated circuit communicatively coupling the write clock output signal and the write clock input signal, and the data output signal and the data input signal, with the signal paths providing said received write clock input and data input signals with a relative timing said produced between said write clock output and data output signals.

In one embodiment, the destination wrapper includes an asynchronous first-in, first-out queue (aFIFO) providing an intermediate storage of information received on the data input signal lines from a first clock domain with timing corresponding to the write clock input signal and provided to the receiving system operating in a different clock domain that is timed by a read clock received from the receiving system. In one embodiment, the destination wrapper uses a unary code, not a Gray code, to determine locations within the aFIFO. In one embodiment, the destination wrapper produces token-based flow control information provided to the source wrapper over a flow control signal path for controlling sending of information from the source wrapper to the destination wrapper. In one embodiment, each of the sending and receiving systems is synchronous.

2. Description

A Delay-tolerant Asynchronous Interface (DANI) is typically used to make the clock domains for reusable silicon intellectual property (IP) cores completely independent of each other. In fact, a DANI-wrapped IP core usually appears to its environment as if it were clockless. This property is necessary to address the variability in data transmission-time between source and destination. This variability is a result of the lack of predictability of the properties of transistors and their interconnections in today's leading-edge, integrated-circuit manufacturing processes. The term "asynchronous" is used in referring to the wrappers because they provide a non-synchronous interface between sending and receiving systems. One embodiment employs dual clocking of components in the asynchronous interfaces.

A DANI wrapper is applied to the IP core that is the source of data to be transmitted or it can be applied to the IP core that is the destination of that data. The transmission time over the route between source and destination may vary, both within and among integrated circuits and be more than a single clock period in duration. The source of data may be synchronous and the destination for that data may also be synchronous, but may be operating at a different clock frequency and/or phase. However, this invention also applies if the source, destination or both have an irregular clock and/or are asynchronous.

There are many possible embodiments of a DANI. Note, the term "one embodiment" is used herein to reference a particular embodiment, wherein each reference to "one embodiment" may refer to a different embodiment, and the use of the term repeatedly herein in describing associated features, elements and/or limitations does not establish a cumulative set of associated features, elements and/or limitations that each and every embodiment must include, although an embodiment typically may include all these features, elements and/or limitations. Also, typically same figure numbers used in different figures typically refer to the same thing in each figure; and typically the last two digits of a three-digit reference number correspond to a same thing but in different embodiments.

One embodiment is expressed as a hierarchical set of block diagrams. At the top level there are two alternative cases: DANI without flow control. A wrapper for the destination IP core that can be used when the source clock frequency is never greater than the destination clock frequency. A trivial wrapper for the source may also be included. DANI with flow control. Wrappers applied to both source and destination IP cores that can be used no matter the relationship between source and destination clock frequencies.

Section 1 reviews the case without flow control. The flow-control case in Section 2 then requires only a few additional ideas. Section 3 reviews some synchronization issues. Section 4 discusses some practical issues related to signal integrity. Section 5 reminds the reader of the vast number of embodiments of the teachings described herein.

1. DANI without Flow Control.

FIG. 1A illustrates a source wrapper and sending system 100 of a first clocked domain according to one embodiment. FIG. 1B illustrates a destination wrapper and receiving system 150 of a second clock domain according to one embodiment. One embodiment communicatively couples write clock signal 131 and w data lines 141 of FIGS. 1A and 1B to provide a reliable interface between two independently clocked domains. This design problem is called "clock domain crossing" and is a notoriously difficult task. Conventional solutions compromise either reliability or efficiency.

Specifically referring to FIG. 1A, sending system 110 produces the three signals of data 113 (w-bits wide), a

Shown in FIG. 3A is Head register H.sub.W 372 and shown in FIG. 3B is Tail register T.sub.R 380 used in one embodiment to synchronize communications between two independently clocked domains. The Head register H.sub.W 372 is composed of a shift register with d flip-flops (e.g., typically corresponding to the maximum number of entries that can be stored in an aFIFO 180 of FIG. 1B). The first $d-1$ shift-register flip-flops, FF.sub.1, FF.sub.2, . . . FF.sub. $d-1$, shift their Q outputs to the D input to the right. FF.sub. d shifts its Q output back to the D input of FF.sub.1. Thus, for example for $d=4$ and starting with the register initialized to all zeros we have the sequence:

0000.fwdarw.1000.fwdarw.1100.fwdarw.1110.fwdarw.1111.fwdarw.0111.fwdarw.0-
011.fwdarw.0001.fwdarw.0000

This sequence is a unary code that is fixed in length and repeats cyclically, stepping forward on each rising edge of write clock 131. Note that H.sub.W 372 contains a code for which a transition from 1 to 0 or from 0 to 1 in the example sequence of four bits identifies a unique aFIFO location that is used to construct a four-bit address pointer. This rule applies except for the 1111 and 0000 cases when the right-most bit is the pointer. In one embodiment, a gray code, lookup table, and/or other sequence generator is used instead of the unary code described supra.

This particular, fixed-length unary code has the property that only one bit changes at each step in the sequence and can be easily generalized to any number of bits d . The property of the code wherein only a single-bit changes on each rising edge of the write clock facilitates the synchronization that takes place in H.sub.R 274.

Referring to FIG. 2, on each rising edge of read clock 191 all the bits of H.sub.W 272 (which is H.sub.W 372 of one embodiment) are copied to H.sub.R 274, a register synchronized to the receive clock (e.g., read clock 191). This synchronization step assumes that a single read clock cycle allows sufficient settling time to achieve the desired mean time between failures (MTBF). However, if an increased MTBF is required, added clock cycles can be inserted to increase the effective settling time. Alternative such schemes are described in Section 3. It is important to recall that only one bit of H.sub.W 372 of FIG. 3A changes at a time in one embodiment. It does not matter if a transition is missed because the next clock will catch it. However, if the changing bit of H.sub.W 372 remains metastable throughout the allowed settling-time, a synchronization failure may occur.

Referring to FIG. 3B, tail register T.sub.R 380 is like H.sub.W 372 (of FIG. 3A), except it steps on read clock 191 and has an active enable signal instead of being fixed high. T.sub.R 380 uses the same d -bit unary code, as do H.sub.W 372 and H.sub.R 274 (of FIG. 2). When the aFIFO 180 (of FIG. 1B) is empty, the codes in H.sub.R 274 and T.sub.R 380 are identical and both synchronized to read clock 191 so that the empty signal 173 (of FIG. 2) is false (e.g. empty is true). When the codes in H.sub.R 274 and T.sub.R 280 are not identical they can be compared and a empty signal 173 generated. This empty signal 173 is used to enable the T.sub.R register 380 so that it does not move ahead in its cycle unless the aFIFO 180 has data to be read. The U.fwdarw.X decoder 276 and 286 (of FIG. 2) takes the codes used in the H.sub.R 274 and T.sub.R 380 registers and decodes them by converting to a "one-hot" code suitable for enabling a single register in the aFIFO 480. One embodiment uses the conversion defined by the following equations:

$X_{sub.i} = U_{sub.i} \text{sym} U_{sub.i+1}; i=1, 2, \dots, d-1$
 $X_{sub.d} = U_{sub.d} \text{sym} \dots_{sub.1}; i=d-1$
 An example conversion from U.fwdarw.X for $d=4$ is 0000.fwdarw.0001, 1000.fwdarw.1000, 1100.fwdarw.0100, 1110.fwdarw.0010, 1111.fwdarw.0001, 0111.fwdarw.1000, 0011.fwdarw.0100, 0001.fwdarw.0010.

H.sub.W register 372 (of FIG. 3A) shifts on every rising edge of write clock 131. The details of the T.sub.R register 380 (of FIG. 3B) are similar except that it shifts on the rising edge of read clock 191 unless the empty signal 173 is not asserted.

Shown in FIG. 4 is an aFIFO 480 used in one embodiment. As shown, aFIFO 480 uses d registers, each w -bits wide. When w -wide data (141) are transmitted on the rising edge of write clock 131, only one of the d registers is write-enabled as determined by the d -bit write enable signal 171. The Q outputs of all the registers 482 are multiplexed (490) together and only the register selected by the d -bit read enable signal 172 is presented as output w -bit wide data 181.

If care is not taken in laying out an integrated circuit, the temporal relationship among the w -bit data lines

Embodiment 900 is a familiar two-stage synchronizer 900 instantiated for each of the d bits in H.sub.R 274 (FIG. 2). It replaces the H.sub.R block 274 in FIG. 2. The MTBF is much larger because of a larger $t_{sub.S}$ and a smaller $T_{sub.W}$. In fact, $t_{sub.S} = 2t_{sub.R} - t_{sub.L} - t_{sub.SU}$, an increase of $t_{sub.R}$ over the single stage case. The smaller value of $T_{sub.W}$ and the value of τ have to be determined from simulation using specific circuit parameters. However, these changes are small compared to the effect of the increase in the value of the exponent.

In embodiment 920 of FIG. 9B, the extra stage of synchronization follows the logic used to calculate inequality between H.sub.R 274 and T.sub.R 280. The value of $t_{sub.S}$ is unchanged from that of FIG. 9A, but the values of $T_{sub.W}$ and τ may be different. Simulation is used to determine their values in one embodiment. If additional settling time is required, a synchronizer with more than two stages may be used in either embodiment 900 of FIG. 9A or embodiment 920 of FIG. 9B.

It might seem that the indeterminacy resulting from marginal triggering of the flip-flop in embodiment 920 of FIG. 9B could lead to an erroneous output 173 stating that the aFIFO 180 (FIG. 1B) had data available when in fact it was empty. If this were true, it would negate the advantage in MTBF because the extra clock period of settling time would not be available. However, such a circumstance cannot occur because transitions from data available to empty only occur as a result of advancing the T.sub.R register 280 (FIG. 2), an action that does not produce metastability. Only advancement of the H.sub.W register 272 (FIG. 2) can produce metastability and the resulting indeterminacy is benign.

Which design is best will depend on circuit parameters. However, embodiment 920 of FIG. 9B requires only one additional flip-flop, whereas embodiment 900 of FIG. 9A requires d extra flip-flops. For example, when $d=4$ the embodiment 920 of FIG. 9B requires only one additional flip-flop while the embodiment 900 of FIG. 9A requires four. For typical bus widths such that $w \gg d$, the increase of d flip-flops for embodiment 900 of FIG. 9A is only a small fractional increase in required resources.

Note, this analysis discussed supra applies to other embodiments, such as that of wrapper destination control 770, including logic 790, of FIG. 7; wrapper destination control 270, including logic 290, of FIG. 2; and wrapper source control, including logic 696, of FIG. 6.

4. Signal Integrity Issues.

The write clock line 131 and data bus 141 (of FIGS. 2A-B, with this analysis applying to other one embodiments of FIGS. 5A-B) may travel over a substantial portion of the integrated circuit as indicated by the ellipsis in the lines. Transitions on data bus 141 occur at the frequency of rising edges of the clock. However, transitions on write clock line 131 occur at twice that frequency and as a result may be subject to threats to signal integrity, particularly for long runs. It is desirable that write clock line 131 and data bus 141 have the same upper frequency limit.

It is also desirable to have the source wrapper 520 launch the data 541 and the write clock 531 with a well-defined phase relationship to each other. This simplifies the application of relative timing constraints and can be done if all signals are similarly registered at the source wrapper 520. However, registering the data is difficult to do when the clock line must have twice as many transitions as the data lines.

Both of these issues can be addressed by reducing the frequency of write clock 531 to half that of source clock 511. One scheme for accomplishing this frequency division is by including a toggle flip-flop in the source control 530 of FIG. 1. At the destination wrapper 560 it then becomes necessary to shift H.sub.W (in destination control 570) and load the aFIFO registers (580) on both the rising and falling edges of write clock 531. Even numbered bits in H.sub.W and even numbered registers in aFIFO 580 will then have their clock inputs inverted. As a result the depth d of aFIFO 580 must be an even number in one embodiment.

In an alternative scheme, two toggle flip-flops are included at the source control 530 of FIG. 1, one toggling on the rising clock edge and one on the falling edge. The two half-frequency clock lines are transmitted to the destination and, by combining them in an XOR gate, the original clock frequency can be recovered.

These alternative schemes for reducing the transmitted source-synchronous clock frequency have different

