US 20080181257A1

(19) **United States**

(12) **Patent Application Publication**

(10) **Pub. No.: US 2008/0181257 A1**

(43) **Pub. Date: Jul. 31, 2008**

Addy et al.

(54) **SYNCHRONISATION IN COMMUNICATION SYSTEMS**

(75) Inventors: **Tim Addy**, Winchester (GB); **Markku Vainikka**, Kiviniemi (FI); **Timo Viero**, Espoo (FI); **William Brockington**, Paignton (GB); **Markku Vahataini**, Maksniemi (FI)

Correspondence Address:
SQUIRE, SANDERS & DEMPSEY L.L.P.
8000 TOWERS CRESCENT DRIVE, 14TH FLOOR
VIENNA, VA 22182-6212 (US)

(73) Assignee: **NOKIA CORPORATION**

(21) Appl. No.: **11/987,760**

(22) Filed: **Dec. 4, 2007**

Related U.S. Application Data

(62) Division of application No. 10/378,653, filed on Mar. 5, 2003, now Pat. No. 7,324,549.

(30) **Foreign Application Priority Data**

Mar. 5, 2002 (GB) 0205142.3

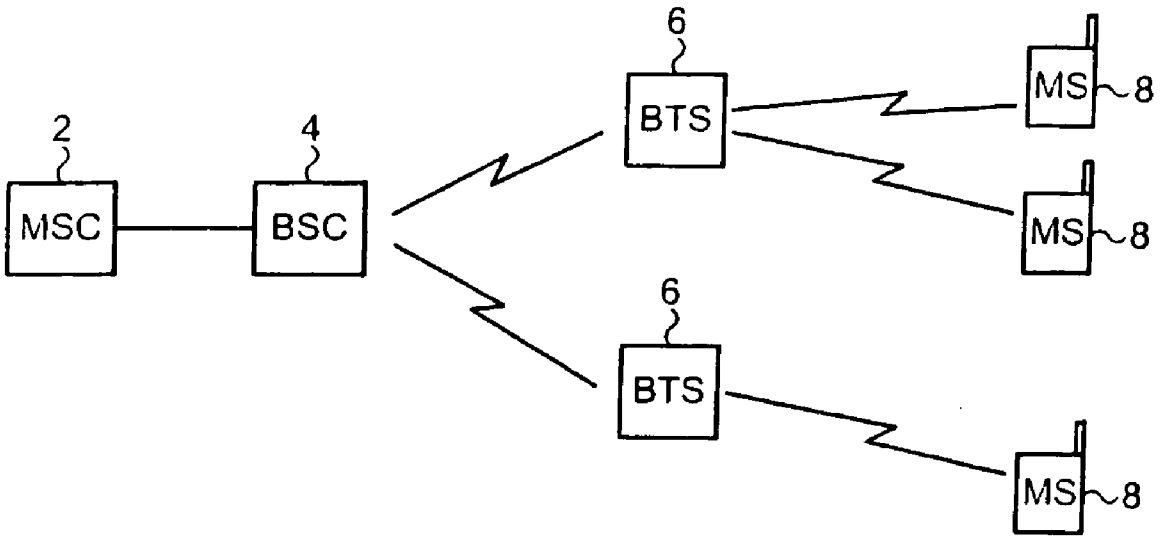
Publication Classification

(51) **Int. Cl.**
H04J 3/06 (2006.01)

(52) **U.S. Cl.** **370/509**

(57) **ABSTRACT**

A method and a communication bus are provided having nodes. Each of the nodes includes a transmitting element to transmit data to another node on the bus, and a receiving element to receive information from another node on the bus. A communication channel is between the transmitting and receiving elements within each node, a transmitter state machine logic controls the synchronisation of the transmitting element, and a receiving state machine logic to control the synchronisation of the receiving element. A storage area maintains the status of synchronisation of the bus. Communication data is synchronised via a bus connecting first and second nodes. A plurality of encoded bytes, each encoded byte represented as a 10 bit code, is transmitted from the first node. The second node receives and decodes the bytes, where any decoding errors in a byte are detected. A synchronised status is indicated.



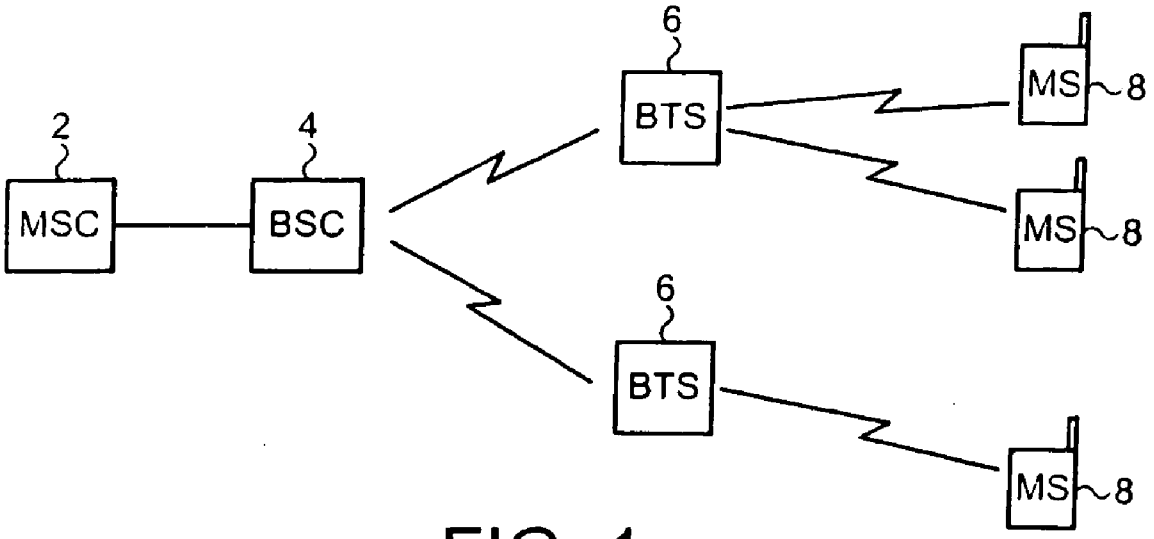

2
4
6
6
MSC
BSC
BTS
BTS
MS
8
MS
8
MS
8

FIG. 1

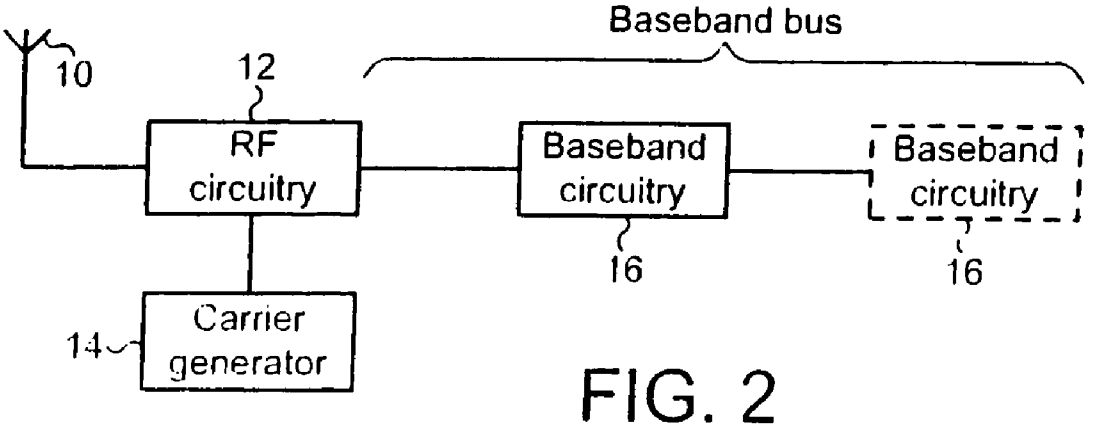

Baseband bus
10
12
RF circuitry
Baseband circuitry
Baseband circuitry
16
16
14
Carrier generator

FIG. 2

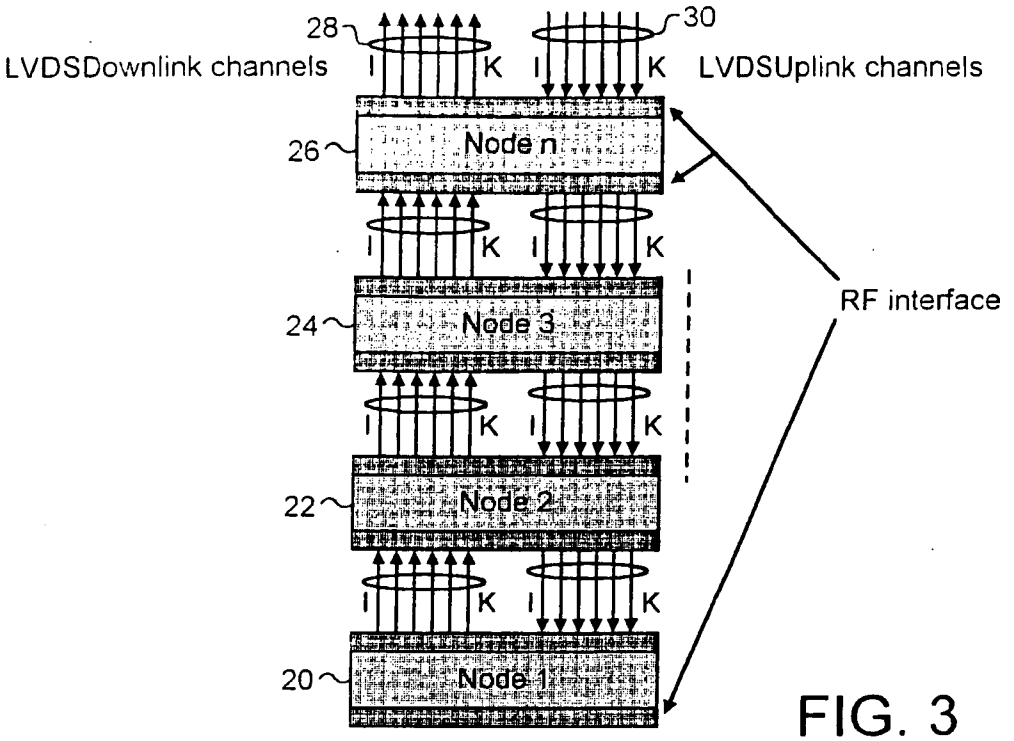


FIG. 3

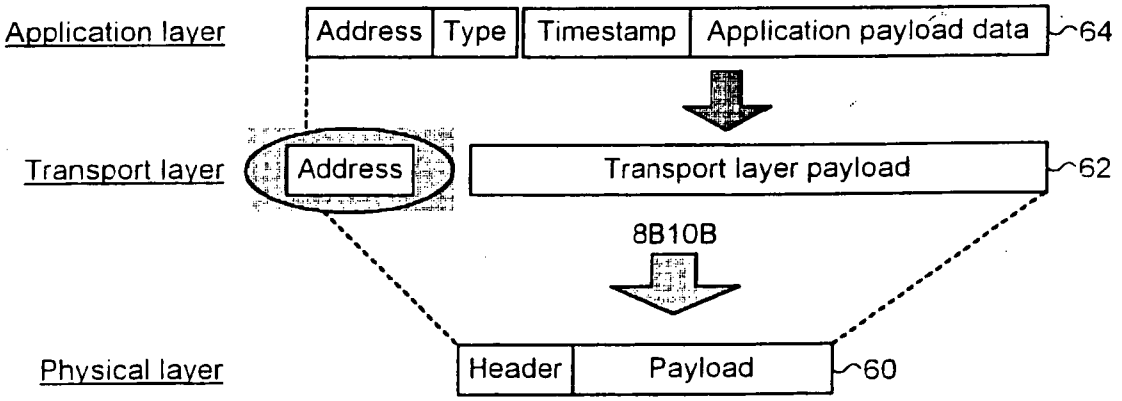


FIG. 4

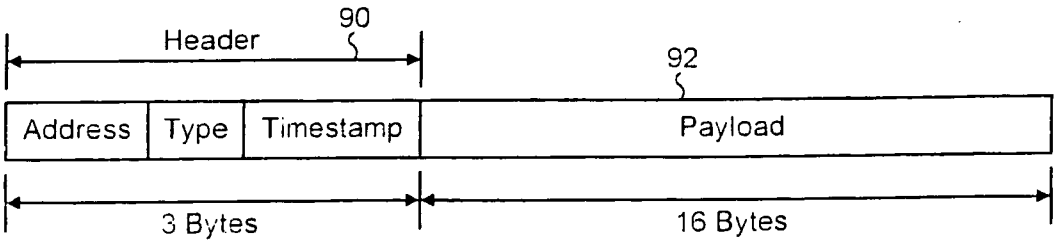


FIG. 6

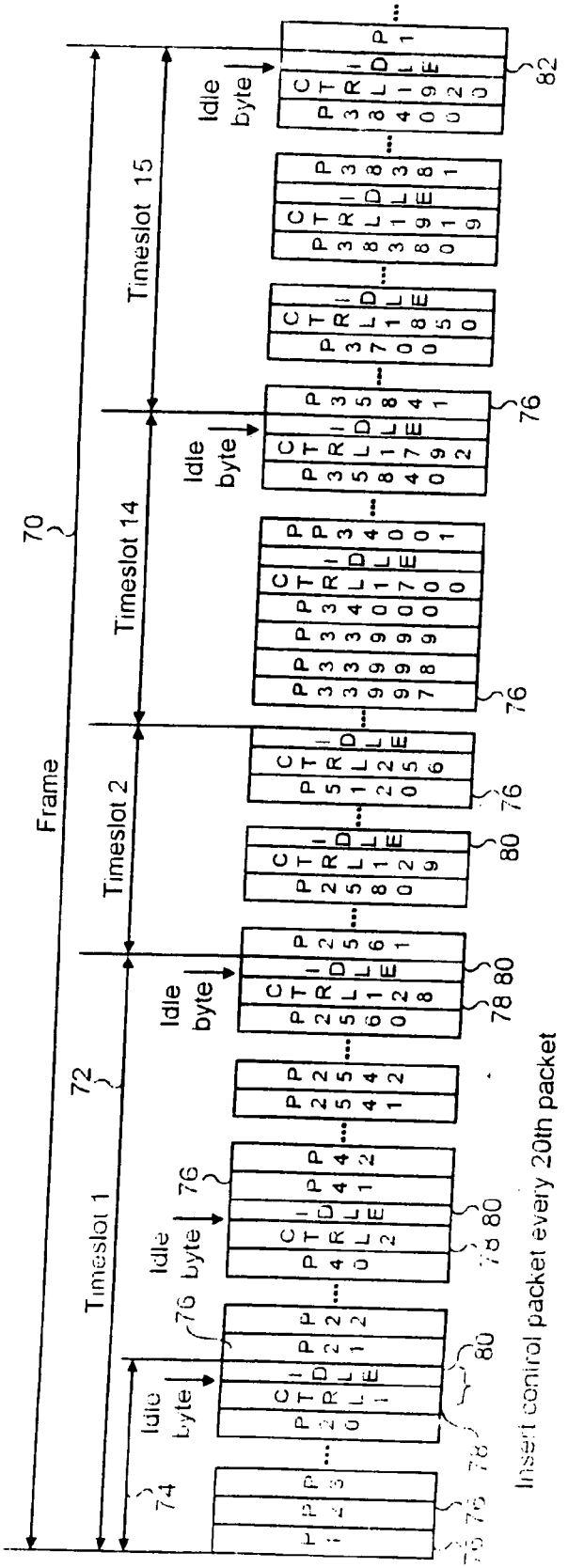

Frame
70
Timeslot 1
72
Timeslot 2
Timeslot 14
Timeslot 15
Idle byte
74
76
78
80
82
Insert control packet every 20th packet

FIG. 5

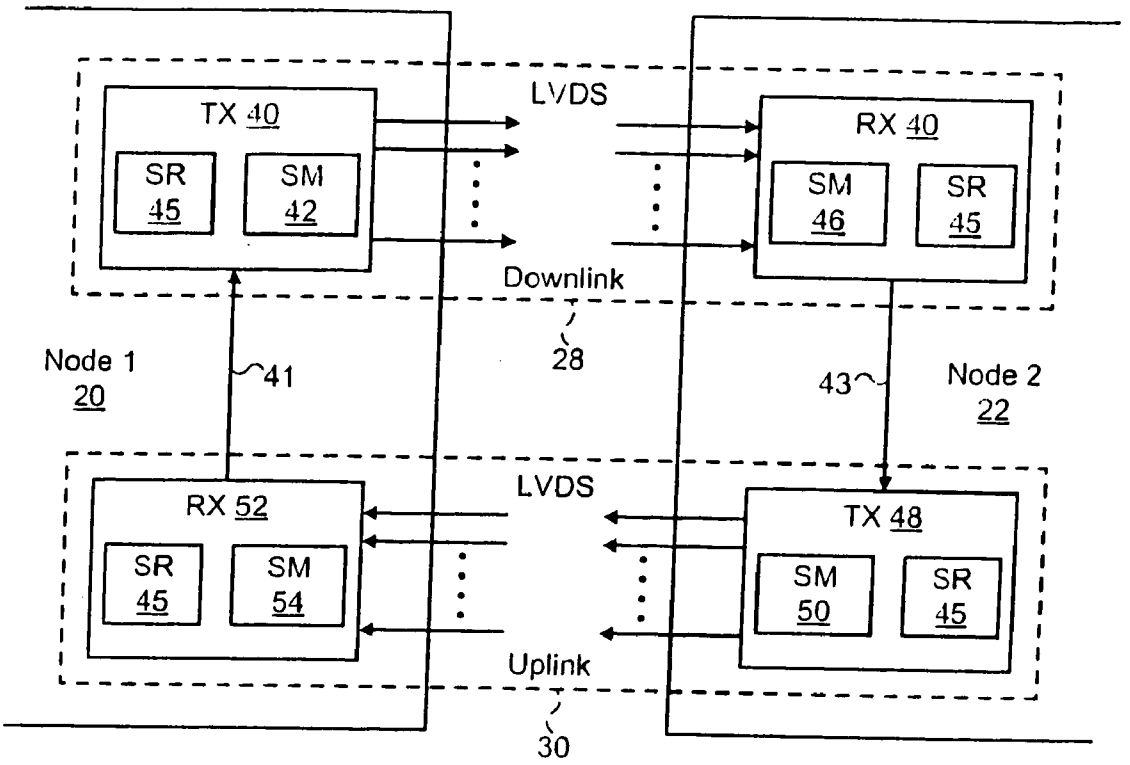


FIG. 7

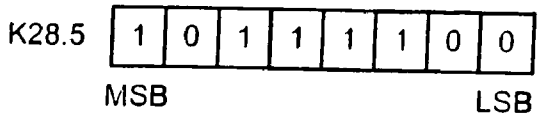


FIG. 10a

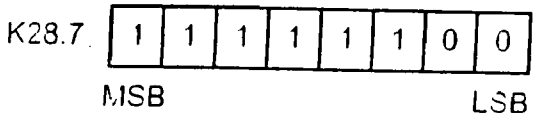


FIG. 10b

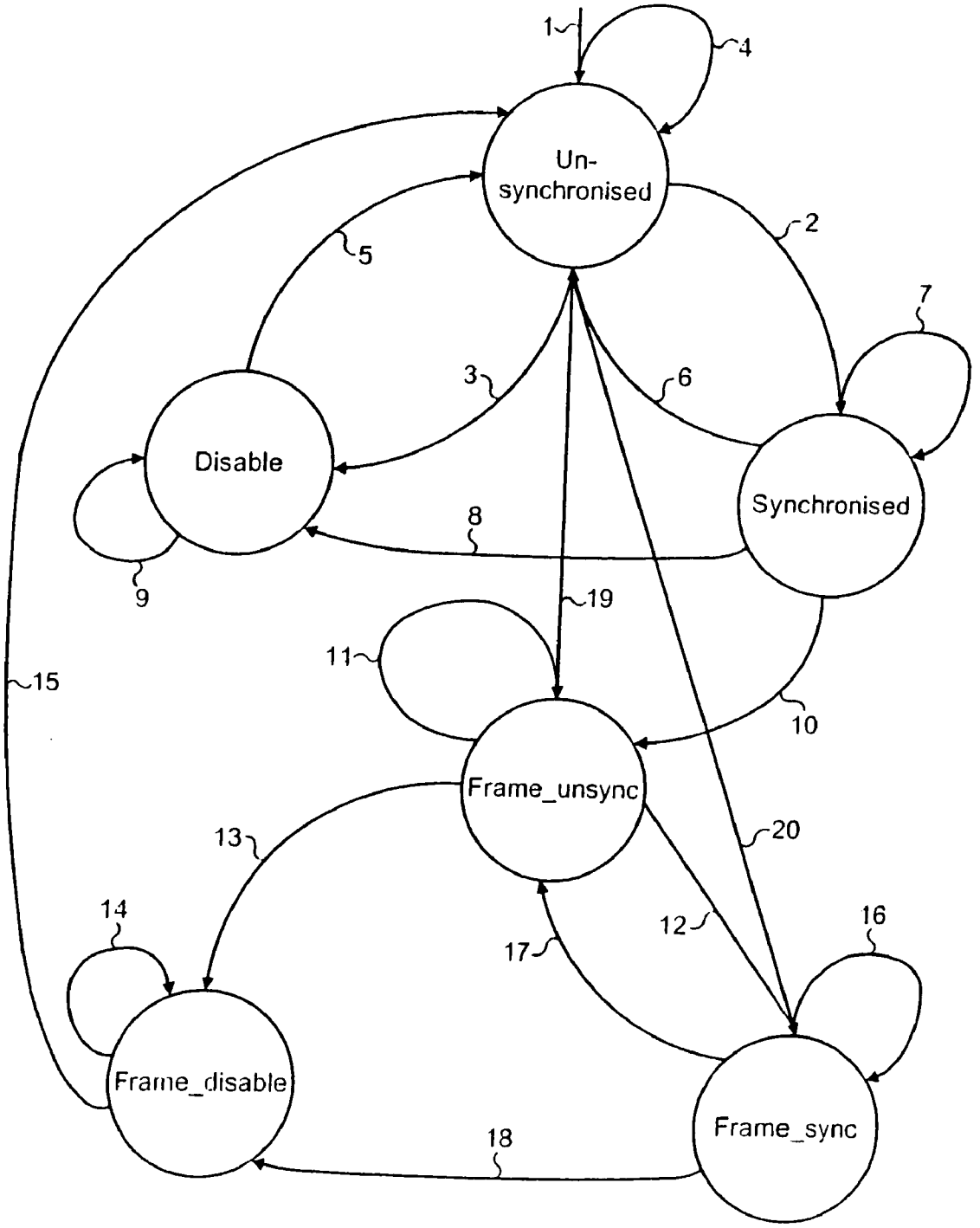

1
4
Un-
synchronised
2
5
7
3
6
Disable
Synchronised
8
9
19
11
15
10
Frame_unsync
13
20
14
12
16
17
Frame_disable
Frame_sync
18

FIG. 8

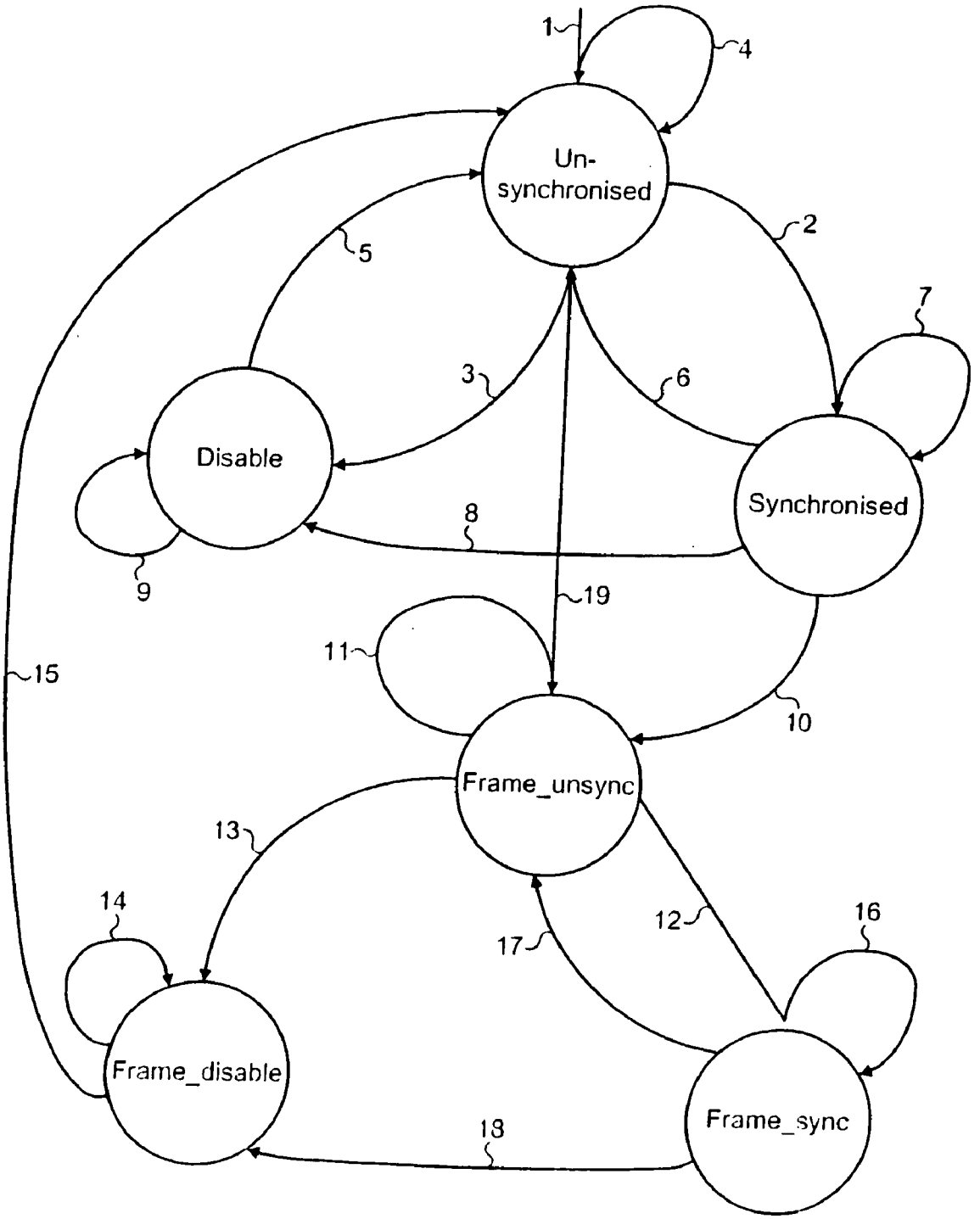

1
4
Un-
synchronised
2
5
7
3
6
Disable
Synchronised
8
9
19
11
15
10
Frame_unsync
13
14
12
16
17
Frame_disable
Frame_sync
18

FIG. 9

SYNCHRONISATION IN COMMUNICATION SYSTEMS

REFERENCE TO RELATED APPLICATIONS

[0001] This application is a divisional application of U.S. patent application Ser. No. 10/378,653 filed on Mar. 5, 2003, which claims priority to British Patent Application Serial No. 0205142.3 filed Mar. 5, 2002 in Great Britain. The subject matter of the earlier filed applications is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

[0002] 1. Field of the Invention

[0003] The present invention is concerned with synchronisation of baseband communications in a wireless communications network.

[0004] 2. Description of the Related Art

[0005] Within a base transceiver station of a wireless communications network, a bus protocol is used to communicate between different nodes. The present invention is concerned particularly but not exclusively with communication between baseband (BB) and radio frequency (RF) nodes in the base transceiver station. Nodes are implemented in a plurality of different ways, and in the following description it is understood that the term "node" implies any appropriate hardware unit, for example an ASIC, processor or FPGA, etc.

[0006] The bus protocol used between the different nodes of the base transceiver station is used to transfer digitised transmitter (TX) and receiver (RX) samples as well as other information.

[0007] The present invention addresses the problem of synchronising a bus, in particular a high speed bus operating a bus protocol used for communication between different nodes in a base transceiver station.

[0008] It is a further aim of the present invention is to provide a frame format used in conjunction with synchronisation methods for synchronising communications on a multi-mode communications bus, which does not require complex circuitry.

SUMMARY OF THE INVENTION

[0009] In general terms, the invention relates to bus synchronisation using idle codes, with the possibility of detecting 8bl10b decoding status. In the described embodiment initial synchronisation and synchronisation at run time is discussed. The position in a frame and value of the idle code is utilised.

[0010] According to one aspect of the present invention Is a provided a method of transmitting data at a line rate from a wireless interface to a bus operating at a bus rate, the method comprising transmitting the data in a packet format consisting of a plurality of frames each having a plurality of time slots, each time slot having successive message groups, wherein each message group comprises a plurality of data messages containing said data and an idle code containing no said data; wherein the number of idle codes in each frame is selected so that the bus rate matches the line rate.

[0011] According to a further aspect of the present invention there is provided a method of transmitting data at a line rate from a wireless interface to a bus operating at a bus rate, the method comprising transmitting the data in a packet format consisting of a plurality of frames each having a plurality of time slots, each time slot having successive message groups, wherein each message group comprises a plurality of data messages containing said data and an idle code containing no said data; wherein the number of idle codes in each frame is selected so that the bus rate matches the line rate.

[0012] According to yet another aspect of the present invention there is provided a communication bus operable at a bus rate and having at least a first node and a second node that are linked by communication channels for transmitting at said bus rate data generated at a line rate, said first node having a transmitting element and said second node having a receiving element, wherein the transmitting element of said first node is operable to transmit data in a packet format consisting of a plurality of frames each having a plurality of time slots, each time slot having successive message groups, wherein each message group comprises a plurality of data messages containing said data and an idle code containing no said data; wherein said number of idle codes in each frame is selected so that the bus rate matches the line rate and wherein the receiving element of the second node is arranged to detect said idle codes for synchronisation purposes.

[0013] According to a still further aspect there is provided a method of synchronising a data communication over a bus in a packet format, said data having been generated at a line rate over a wireless interface consisting of a plurality of frames each having a plurality of time slots, each time slot having successive message groups, wherein each message group comprises a predetermined number of data messages containing said data and an idle code containing no said data, the method comprising detecting at a bus node said idle codes until a predetermined number of said idle codes have been detected indicating successful synchronisation.

[0014] According to a still further aspect there is provided a method of synchronising data communication via a bus connecting first and second nodes comprising: transmitting from the first node a plurality of bytes, each byte representing a 10 bit sequence as an 8 bit code; receiving and decoding said bytes at the second node, whereby any 8b10b encoding errors in a byte are detected; and indicating a synchronised status for the bus based on the detection of received bytes which do not contain 8b10b decoding errors.

[0015] According to a still further aspect there is provided a method of synchronising data communication via a bus connecting first and second nodes comprising: transmitting from the first node a plurality of bytes, each byte representing a 10 bit sequence as an 8 bit code; receiving and decoding said bytes at the second node, whereby any 8b10b encoding errors in a byte are detected; and indicating an unsynchronised status for the bus based on the detection of received bytes containing 8b10b decoding errors.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] For a better understanding of the present invention and to show how the same may be carried into effect, reference will now be made by way of example to the accompanying drawings in which:

[0017] FIG. **1** shows the basic structure of a wireless communications network;

[0018] FIG. **2** shows the context of the present invention for use at baseband frequencies;

[0019] FIG. **3** shows an embodiment of the architecture of the physical baseband bus of the present invention;

[0020] FIG. **4** shows the baseband bus protocol stack according to an embodiment of the present invention;

[0021] FIG. **5** shows a frame format according to an embodiment of the present invention;

[0022] FIG. **6** shows an embodiment of the message structure of the present invention;

[0023] FIG. **7** shows two communicating nodes of the baseband bus;

[0024] FIG. **8** shows a state transition diagram of the logic implemented within the receiving elements of each baseband node;

[0025] FIG. **9** shows a state transition diagram of the logic implemented within the transmitting elements of each baseband node; and

[0026] FIGS. **10***a* and **10***b* show the bit patterns for the idle codes according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0027] FIG. **1** shows the traditional elements of a wireless communications network. A Mobile Switching Center (MSC) **2** acts as an interface with other networks, for example the Public Switched Telephone Network (PSTN). The MSC **2** controls a plurality of Base Station Controller's (BSC) **4**, where each BSC **4** in turn controls plurality of Base Transceiver Stations (BTS) **6**. Each BTS **6** has a certain coverage area known as a "cell", which is used to communicate with mobile stations in their respective coverage areas.

[0028] FIG. **2** shows the typical components of a transceiver used in wireless networks where radio signals are transmitted and received using antennae **10**. These radio signals are transferred at a carrier frequency determined by a carrier generator **14**. Spectrum allocations are licensed and will vary depending on the country involved and the type of air interface being used, for example WCDMA, GSM, etc. In a typical transceiver, for example as would be found at each of the BTS's **6** of FIG. **1**, there is a baseband circuitry **16** concerned with the processing of baseband signals. These baseband signals are then converted into a carrier signal by RF circuitry **12** for transmission by the antennae **10**. The embodiments of the present invention discussed herein are in the context of the baseband circuitry **16**, where a bus protocol connects BB and RF nodes together.

[0029] FIG. **3** shows the physical architecture of the baseband bus where nodes **20**, **22**, **24**, **26** are chained in a point-to-point manner according to a first embodiment of the present invention. Each of the nodes can be implemented by using an Application Specific Integrated Circuit (ASIC). The nodes communicate using a first set of communication channels **28** in an Uplink direction and over a second set of communication channels in the opposite or Downlink direction **30**. Each of the nodes are shown as having a plurality of communication channels referred to herein as links, i.e. 1 to k, in either direction. Some of the nodes may have an RF interface. Other nodes are baseband nodes with no RF interface.

[0030] Packetised data may be sent over the baseband bus. As can be seen from FIG. **3** the bus is implemented using a plurality of links. A single link is however possible. In a preferred embodiment the baseband bus consists of point-to-point connections forming a chained bus. In this embodiment, the point-to-point connections are achieved using Low Voltage Differential Signalling (LVDS).

[0031] The baseband bus uses a three-layer protocol with fixed length messages. Any information to be sent over the baseband bus is packed into messages of known type. The three layers are shown in FIG. **4**. The physical layer **60** is responsible for the transmission of messages and includes framing, coding and serialisation of the messages. The transport layer **62** is responsible for the end-to-end delivery of the messages or routing of the messages. The application layer **64** provides the mapping of different types of packets to the payload.

[0032] In CDMA applications, data at the application layer is continuous, but for transfer over the bus, the continuous data of the application layer is time sliced into short messages that are transferred over the high-speed physical layer. At the receiving node a continuous stream is recovered.

[0033] FIG. **5** shows a frame **70** of the physical layer with a certain packet format being sent over the bus in both the uplink and downlink directions. The frame has a fixed 10 ms period. Frames are inserted consecutively onto the bus. A frame is split into fifteen timeslots **72** where each timeslot contains a plurality of message groups **74**. Each message group **74** has a fixed predetermined number of data messages **76**, one control message **78** and an IDLE message **80**.

[0034] A preferred embodiment of the message structure **76** is shown in FIG. **6**. These messages are transmitted over the physical layer **60** shown in FIG. **4**. In this embodiment, the messages are of a fixed length of 19 bytes having a header portion **90** of 3 bytes and a payload portion **92** of 16 bytes. Thus, all messages including control and data have the same message definition. For one embodiment as will be described hereafter, the idle message is in the form of an idle byte.

[0035] When there is no data message to transmit, that is, no message has been received from the transport layer for a given time slot, then the physical layer **60** transmits an empty message, which can be implemented by transmitting "1" bits for the entire message. The physical layer at the receiving node will detect the existence of an empty message and rejects such messages, thereby making these messages invisible to the upper protocol layers **62**,**64**.

[0036] In the embodiment shown in FIG. **5**, the message group **74** comprises one control message **78** inserted after every twenty data messages **76** and an IDLE byte **80** inserted after the control message **78**. The same IDLE byte **80** is used at the end of each message group **74** with the exception that a special IDLE byte **82** Is used in the final timeslot to identify the end of the frame. The significance of the special IDLE byte **82** will be discussed later.

[0037] In the present embodiment, the bus speed is chosen to be 768 Mbps. A derivative of the BTS reference system clock is used as a clock for the baseband bus and the physical layer **60** of the bus protocol is synchronised to the BTS system clock. However, application layers of the bus protocol can operate asynchronously with respect to the timing of the physical layer, which is especially useful for GSM or EDGE application where data is not continuous but instead is transmitted in bursts and is inherently asynchronous.

[0038] For the present embodiment, consider the situation of a WCDMA uplink. Consider a signal described in terms of its in-phase component (I) and its quadrature component (Q) where the I and Q values are each 8 bits.

[0039] At a sample rate of 7.68 Msps (Mega samples per second), this gives a payload rate of 7.68M*(8*2)=122.88 Mbps (Mega bits per second). Since the packet has a 3 byte header and 16 byte payload, the packet rate is 122.88*(19/16)=145.92 Mbps. After an 8b10 coding scheme is used, the line rate is 145.92M*(10/8)=182.4 Mbps.

[0040] FIG. **3** shows a plurality, i.e. 1 to k links being used to communicate in either direction. Each link supports four paths so that each uplink, of the group of uplinks **28**, Is required to support four uplink paths, giving a line rate of

182.4M*4=729.6 Mbps per link. if control messages are inserted every twentieth packet, this gives a line rate of 729.6*(21/20)=766.08 Mbps.

[0041] However, a bus speed of 768 Mbps has been chosen. Therefore an extra 768−766.08=1.92 Mbps is needed in order to match the line rate to the bus speed. 1-0 achieve this, taking into account the 8bl Ob coding, 1.92M*(8/10)=1.536 Mbps of "plain" data needs to be inserted, which is 1.536 Mbps/8=192000 "plain" bytes per second. Each frame has a time period of 10 ms, therefore 192000/100=1920 IDLE bytes per frame are inserted. Each frame has 15 time slots resulting in 1920/15=128 IDLE bytes per time slot. There are 2560 data messages per time slot, which means 128/2560=1 byte per 20 messages should be an IDLE byte in order to match the tine rate to the bus speed.

[0042] Therefore, by insertion of IDLE bytes it becomes possible to match the line rate to be an integer multiple of the system clock rate and alleviates the need for additional complex circuitry needed to account for a mismatch between the line rate and the bus speed.

[0043] FIG. **7** shows the LVDS point-to-point connections in an uplink direction **30** and a downlink direction **28** between a first node **20** and a second node **22** of the bus. Each LVDS point-to-point connection corresponds to each of the 1 to k links shown in the uplink **28** or downlink directions **30** of FIG. **3**. Each node **20**, **22** comprises a transmission element **40**, **48** and a receiving element **44**, **52**. In the downlink case, a transmission element **40** transmits information from a first node **20** to a receiving element **44** in a second node **22** using LVDS connections. In the uplink direction, a transmitting element **48** transmits information from a second node **22** to the receiving element **52** of a first node **20**. A communication channel **41** exists between the transmitter **40** and receiver **52** of the first node **20**. Also, a communication channel **43** exists between receiver **44** and transmitter **48** of the second node **22**. These communication channels may be used by receivers on a node to inform the transmitters if a loss of synchronisation occurs.

[0044] It can also been seen that each of the transmitting 40, 48 and receiving elements **44**, **52** have their own respective state machine logic **42**, **46**, **50**, **54**. FIG. **8** is a state transition diagram showing the state machine logic **46**, **54** of the receiving elements **44**, **52**. FIG. **9** Is a state transition diagram showing the state machine logic **42**, **50** of the transmitting elements **40**, **48**. Tables 1, 2 and 3 given below can be used to interpret these state transition diagrams.

[0045] Table 1 below provides a definition of the signals used in the state machine for synchronisation.

TABLE 1

Signal	Definition	Active state
NdFifo Valid	Node clock domain Fifo valid signal to indicate that the fifo is passing valid bytes to the rx decoder	Active high = '1', default = '0'
FRAME_IDLE_VALID	Frame IDLE byte (K28.7) is received to indicate that a new frame boundary is present	Active high = '1' for a single cycle when a K28.7 IDLE byte is received. Default = '0'
SET_RUN_TIME_MODE	Value '1' enables run time mode, '0' disables run time mode. Messages are transferred over the bus in run-time mode.	Active high = '1', default = '0'
ENABLE_BUS_TRANSCEVER	The bus transceiver is enabled by value '1'. This enables the transmitter to start sending IDLE.	Active high = '1', default = '0'
RESTART_FROM_DISABLE	This signal resets the state machine when the state is either DISABLE OR FRAME_DISABLE.	Single cycle Active high pulse = '1', default = '0'

[0046] It should be noted that the IDLE code **80** inserted at the end of each message group **74** is referred to herein as the "K28.5" IDLE byte, whereas the special IDLE code **82** inserted at the end of each frame **70** is referred to herein "K28.7" IDLE byte. FIG. **10***a* and FIG. **10***b* show the bit patterns that make up the K28.5 and K28.7 idle bytes respectively, in the 8 bit domain. These bit patterns are known as so-called "comma characters", which are uniquely chosen to indicate possible errors.

[0047] These codes (and other data bytes) are transmitted as 10 bits using an 8b10b encoding scheme, for example as described in "A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code", by Widmer and Franaszek, IBM J. Res. Develop. Vol. 27 No. 5, September 1983. The transmitter has means for encoding the 8b bytes into 10b codes and the receiver has means for decoding the codes and for error checking.

[0048] Table 2 below defines the state transitions and triggers required for these transitions for the state machine logic **46**, **54** of the receiving elements **44**, **52**.

TABLE 2

Transition	Trigger	Comment
1	ENABLE_BUS_TRANSCEIVER = '1' (values changes from '0' to '1')	Message Group counter reset.
2	SYNC_T valid Message Groups received	SYNC_T message groups of valid IDLE bytes have been received
3	DISABLE_T Message Groups or ENABLE_BUS-TRANSCEIVER = '0'	Timeout counter has been reached without SYNC_T valid Message Groups being received or Application layer eforces the state to DISABLE.
4	Not (2 or 3)	<SYNC_T valid Message Groups have been received and timeout count has been reached
5	RESTART_FROM_DISABLE = '1'	Application layer has acknowledged that the state machine is in the DISABLE state and restarts the synchronization process
6	UN SYNC_T consecutive invalid Message Groups received	Synchronization lost due to UN SYNC_T consecutive invalid Message Groups received
7	Not (6 or 8 or 10)	Physical layer synchronization is maintained and valid IDLE bytes are being received.
8	ENABLE_BUS_TRANSCEIVER = '0'	The transceiver is DISABLED. This transition is forced by Application layer.
9	RESTART_FROM_DISABLE = '0'	Wait in the DISABLE for the Application layer to generate a RESTART_FROM_DISABLE signal.
10	SET_RUN_TIME_MODE = '1' & FRAME_IDLE_VALID	Run time mode has been activated and a K28.7 frame boundary IDLE byte received to indicate the start of empty messages.
11	Not (12 or 13)	Stay in the Frame_UNSYNC_ state
12	FRAME_SYNC_T consecutive valid message groups are received	
13	FRAME_DISABLE_T invalid message groups are received or ENABLE_BUS_TRANSCEIVER = '0' or SET-RUN-TIME = '0'	Transition to the FRAME_DISABLE state as FRAME_DISABLE_T invalid message groups have been received or Application layer halts the transceiver.
14	Not 15	Wait for a RESTART_FROM_DISABLE signal.
15	RESTART_FROM_DISABLE = '1'	The static machine has been reset by Application layer (i.e. restart the initialization process).
16	Not (17 or 18)	Valid Message Groups are being received. Normal message reception state.
17	FRAME_UNSYNC_T consecutive invalid message groups are received.	Whilst in the FRAME_UNSYNC state, FRAME_UNSYNC_T consecutive invalid message groups are received.
18	ENABLE_BUS_TRANSCEIVER = 0 or SET_RUN_TIME = '0'	Transceiver DISABLE
20	GO_TO_FRAME_SYNC = 1 & SET_RUN_TIME_MODE = 1 & FRAME_IDLE_VALID	For test purposes: Run time mode has been activated and a K28.7 frame boundary IDLE byte received to indicate the start of the empty packet bytes.
19	SET_RUN_TIME_MODE = '1' & FRAME_IDLE_VALID	When in UNSYNCHRONIZED state, run time mode has been activated and a K28.7 frame boundary IDLE byte received to indicate the start of empty messages.

[0049] Table 3 below defines the state transitions and triggers required for these transitions for the state machine logic **42**, **50** of the transmitting elements **40**, **48**.

TABLE 3

Transition	Trigger	Comment
1	ENABLE_BUS_TRANSCEIVER = '1' (values changes from '0' to '1')	LOS in receiver is initially "1"
2	LOS = '0'	Receiver has acknowledged that SYNC_T IDLE bytes have been received. Receiver has acknowledged the transition to SYNCHRONISED state by changing LOS to "0"
3	ENABLE_BUS_TRANSCEIVER = '0'	Transmitter has been DISABLE.
4	ENABLE_BUS_TRANSCEIVER = '0' and LOS = '1'	Wait for Receiver to acknowledge receipt of SYNC_TIDLEs or transmitter DISABLE.
19	GO TO FRAME SYNC = '1'	Go to Frame UNSYNC state
5	RESTART FROM DISABLE = '1'	Transmitter has been reset and can transition to UNSYNCHRONISED and start sending IDLE.
6	LOS = '1'	Receiver has generated a Loss Of Signal.
7	LOS = '0' AND ENABLE SUS TRANSCEIVER = '1' and SET RUN-TIME MODE '0'	Send IDLE bytes (with K28.7 at Frame boundary).
8	ENABLE BUS TRANCEIVER = '0'	Transition to the DISABLE state.
9	RESTART FROM DISABLE = '0'	Stay in Disable state
10	SET RUN-TIME MODE = '1' and FClk = '1'	Transition to the FRAME_UNSYNC when the transmitter has sent a K28.7 (last byte in the frame.
11	LOS = '1' AND (ENABLE BUS TRANCEIVER) = '1' OR SET RUN-TIME MODE_'1')	Stay in Frame_Unsync state
12	LOS = '0'	Receiver has acknowledged that it has received FrameUnsyncT valid Message Groups.
13	ENABLE BUS TRANSCEIVER = '0'	Go to FrameDisa e state as the Transmitter has been DISABLE.
14	RESTART FROM DISABLE = '0'	Stay in Disable state.
15	RESTART FROM DISABLE = '1'	Restart the transmitter go to the UNSYNCHRONISED state.
16	LOS = '0' AND ENABLE BUS TRANSCEIVER = '1' AND SET TUN-TIME MODE = '1'	Stay in Frame_Sync state.
17	LOS = '1'	Receiver has de-asserted the Loss Of Signal input so the transmitter must send empty messages.
18	ENABLE BUS TRANSCEIVER_ '0' OR SET RUN-TIME MODE_ '0'	Disable Transmitter

[0050] Broadly speaking, there are two synchronisation algorithms which are applied, i.e. initial synchronisation and frame synchronisation. Initial synchronisation allows an initial check on the link quality of the bus, whereas frame synchronisation allows continuous monitoring of the link quality when the bus is in run time mode. The synchronisation algorithms can report link status information of the bus to upper layers of the protocol stack.

[0051] Initial synchronisation is performed when a bus node is booting up. The purpose of the initial synchronisation

is to determine the status of each bus interface. That is, checking the status of a node's transmitting and receiving elements. Synchronisation may be unsuccessful due to a missing neighbouring node or a failure of the link.

[0052] In the present embodiment, the sequence of steps for initial synchronisation is the following:

[0053] Set state to UNSYNCHRONIZED.

[0054] Reset the message group counter to zero.

[0055] Start transmitting a constant stream of IDLE bytes from the transmitting element of any node, e.g. 20.

[0056] Start reading the IDLE bytes at the receiving element of any node, e.g. 22.

[0057] A message group is considered to be valid when all the IDLE bytes are properly received and there are no 8b10b decoding errors. Otherwise, a message group is considered to be invalid. It should be appreciated that there are (21 messages*19 bytes per message)+1 IDLE byte=400 bytes in a message group.

[0058] When in the state UNSYNCHRONIZED and a value of SYNC T consecutive valid message groups have been received, the state of the state machine then is set to the SYNCHRONISED state.

[0059] When in the state SYNCHRONIZED and a value of UNSYNC T consecutive invalid message groups have been received, the state of the state machine then is set to the UNSYNCHRONISED state. Also, the message group counter is set to zero.

[0060] When in the state UNSYNCHRONIZED and a value of DISABLE T message groups have been received, the state of the state machine then is set to the DISABLE state. The value of DISABLE T is larger or equal to the value of UNSYNC T. When either of the transmitting or receiving elements of a node enter the DISABLE state, the application layer **64** is informed by an interrupt which then can restart the synchronization procedure.

[0061] The above synchronization algorithm can be generalized by considering validity of consecutive received bytes instead of message groups. Furthermore, synchronization can be based on any transmitted data and the success or failure of 8b10b decoding; not just transmission and reception of IDLE bytes.

[0062] The physical layer **60** contains a status register **45** for each transmitting and receiving element of each node of the bus indicating the synchronisation status. For example, DISABLE (000001), UNSYNCHRONISED (000010), SYNCHRONISED (000100). Other state encodings may be used. Regarding the operation of the transmitter during initial synchronization, IDLE bytes are sent in UNSYNCHRONIZED and SYNCHRONIZED states. Note that during initial synchronisation, only IDLE bytes are transmitted to the bus. This is not the case in run-time operation when data is transferred over the bus.

[0063] After the physical layer **60** has been configured into a run-time mode by the application layer (parameter SET_RUN-TIME_MODE is set equal to 1), frame synchronisation can be performed. In run-time mode, messages (e.g. data, control or even empty) are transferred over the bus. In run-time mode, receiver synchronization of a transceiver is started immediately. When the value of parameter SET_RUN-TIME_MODE is changed from 1 to 0, state of the transceiver Is changed to FRAME_DISABLE.

[0064] In frame synchronisation, each transmitting element **40**, **48** synchronises the frame timing with the baseband bus frame clock Furthermore, the status of the frame synchronisation in each receiving element **44**, **52** is constantly monitored. The end of each frame is identified from the unique IDLE byte K28.7. This unique IDLE byte allows one to calculate the received frame offset as well as monitoring of frame synchronization status.

[0065] In the present embodiment, frame synchronisation is applied to all the transmitting and receiving elements of the bus nodes when entering the run time mode and the sequence of steps for frame synchronisation is the following:

[0066] Set the state of the state machine to FRAME UNSYNCHRONIZED.

[0067] Reset the message group counter to zero.

[0068] Start transmitting empty or other valid messages from the transmitting element **40**, **48**.

[0069] With reference to the baseband bus frame clock, read the IDLE byte of each message group from the received byte stream using the receiving element **44**, **52**. The IDLE byte must be the last byte of the message group and any other IDLE bytes are considered to be errors.

[0070] When the IDLE byte of a message group has been properly received and no 8b10b decoding errors are present, consider that message group to be valid. Otherwise, the received message group is invalid.

[0071] When in the state FRAME UNSYNCHRONIZED and FRAME SYNC T consecutive valid message groups have been received, set the state to FRAME SYNCHRONISED.

[0072] When in the state FRAME SYNCHRONIZED and FRAME UNSYNC T consecutive invalid Message Groups have been received, set the state to FRAME UNSYNCHRONISED and reset the message group counter to zero.

[0073] When in the state FRAME UNSYNCHRONIZED and FRAME DISABLE T message groups have been received, set the state to FRAME DISABLE.

[0074] The value of FRAME DISABLE T is always larger or equal to the value of FRAME UNSYNC T. The status register **45** maintains an indication of the status of the frame where the status FRAME DISABLE, FRAME UNSYNCHRONIZED, and FRAME_SYNCHRONIZED correspond to the states 001000, 010000, and 100000 respectively. Other state encodings may also be used. When the transmitting or receiving elements enter the FRAME_DISABLE state, the application layer is informed by an interrupt, which can then restart the synchronization procedure. Regarding the operation of the transmitting elements during frame synchronization, valid messages are sent in the FRAME SYNCHRONIZED state, whereas empty messages are sent in the FRAME UNSYNCHRONIZED and FRAME DISABLE states.

[0075] The synchronisation operation is now described for each respective state.

[0076] UNSYNCHRONISED

[0077] Restart the message group counter.

[0078] The transmitting element starts sending IDLE bytes.

[0079] The LOS is set to '1 '.

[0080] The receiving element waits to receive data.

[0081] Valid bytes are beginning to be passed.

[0082] Initial byte synchronization is performed using the consecutive K28.5 idle code

[0083] The receiving element will start counting valid message groups. If SYNC T consecutive valid message

groups are received, then the state machine transitions to the SYNCHRONIZED state.

[0084] If DISABLE T message groups are received (with 400 bytes in each), then the state machine transitions to the DISABLE state and the Receiver and Transmitters are disabled.

[0085] If ENABLE BUS TRANSCEIVER=O is received, then the state machine transitions to the DISABLE state.

[0086] If SET RUN TIME MODE=1, then the state machine transitions to the FRAME UNSYNCHRONISED state.

[0087] SYNCHRONISED

[0088] Reset the message group counter.

[0089] Set the LOS to '0'

[0090] If UNSYNC_T consecutive invalid message groups are received, then the state machine transitions to the UNSYNCHRONISED state.

[0091] If SET RUN TIME MODE=1, then the state machine transitions to the FRAME UNSYNCHRONISED state.

[0092] If ENABLEBUS_TRANSCEIVER=O, then the state machine transitions to the DISABLE state.

[0093] DISABLE

[0094] Stop all counters.

[0095] Set the LOS to

[0096] In this state, the state machine of the receiving element can only transfer to the UNSYNCHRONISED state when RESTART FROM DISABLE=1.

[0097] FRAME SYNC

[0098] Set the LOS to '0'.

[0099] Restart the message group counter.

[0100] Constantly check the frame synchronization using the K28.7 IDLE byte.

[0101] If FRAME UNSYNC T consecutive invalid message groups are received, then the state machine transitions to the FRAME UNSYNCHRONISED state.

[0102] If ENABLE BUS TRANSCEIVER=0 or SET RUN TIMEMODE=0, then the state machine transitions to the FRAME DISABLE state.

[0103] In the FRAME SYNC state, a valid message group exists when a K28.5 or K28.7 IDLE byte code is at byte **399** and there are no invalid IDLE codes in bytes **0** to **398** and no 8bl Ob decoding errors are present.

[0104] FRAME UNSYNC

[0105] Set the LOS to '1'.

[0106] Restart the message group counter.

[0107] If FRAME_SYNC_T consecutive valid message groups are received, then the state machine transitions to the FRAME SYNCHRONIZED state.

[0108] If FRAME DISABLE T invalid message groups are received, then the state machine transitions to the FRAME DISABLE state.

[0109] In the FRAME_UNSYNC state, a valid message group exists when a K28.5 or K28.7 IDLE byte code is at byte **399** and there are no invalid

[0110] IDLE codes in bytes **0** to **398** and no 8b106 decoding errors are present.

[0111] If ENABLE_BUSTRANSCEIVER=O or SET RUN TIME MODE=O, then the state machine transitions to the FRAME DISABLE state.

[0112] FRAME DISABLE

[0113] Stop all counters.

[0114] Set the LOS to '1'.

[0115] In summary, the idle bytes inserted into the frames at the physical layer level are to synchronise the line rate of data transmission to the bus rate set up by the system clock. Also, synchronisation algorithms using these idle bytes to perform different types of synchronisation algorithms. For initial synchronisation, before run time mode, the quality of the communication links between the nodes are tested by transmitting message groups that consist purely of idle codes instead of data messages. The receiving elements then check the received idle codes and if all idle codes (i.e. 400 idle bytes in this embodiment) have been received correctly, then that message group is said to be valid. If SYNC_T consecutive valid message groups are received then initial synchronisation has been achieved, For frame synchronisation, the first algorithm is when the bus is in run time mode but the frames are unsynchronised. Data messages and an idle message now make up the message groups that are transmitted. However, now a message group is considered to be valid when an idle code exists (either K28.5 or K28.7) at the last byte of the message group (i.e. byte **399**) and there are no invalid IDLE codes in the remainder of the message group (i.e. bytes **0** to **398**) and no 8b10b decoding errors are present in the message group. Frame synchronisation is once FRAME_SYNC_T valid consecutive message groups have been received. Also, once frame synchronisation has been achieved it is important to maintain synchronisation. This is accomplished by using the unique idle byte (K28.7) at the end of each frame, which allows one to calculate the received frame offset.

[0116] It should be appreciated that each transmitting or receiving element of each node of the bus can independently assume any of the states described herein.

[0117] It should also be appreciated that FIG. **3** shows a plurality of links, i.e. 1 to k. Thus, it should be understood that the present invention is scaleable to adapt to different data rates.

[0118] It should be appreciated that the frame structure shown in FIG. **5** is one embodiment of the present invention. In this embodiment is that the special IDLE code **82** is inserted at the end of a frame **70**. It should be appreciated that the baseband bus is a multi-mode bus and in conjunction with the layered protocol stack it is intended to support a variety of different air interfaces such as GSM or EDGE. With regards to the IDLE bytes **80**, it should be appreciated that the position of the IDLE byte within the message group **74** may vary in different implementations. Furthermore, the special IDLE byte **82** might be implemented at other locations in each frame, for example at the start as opposed to the end of the frame **70**: all that is needed is that it is in a predetermined known location. Also, the IDLE codes are 1 byte in length in the described embodiment, however for a different embodiment these IDLE codes could be scaled in length so as to match a different frame format.

[0119] It should be appreciated that the implementation of the nodes of the communications bus, shown in FIG. **3**, are not necessarily limited to ASIC's and can also be implemented using other logical devices, for example a Field Programmable Gate Array (FPGA) devices.

1. A communication bus having nodes that are linked by communications channels, wherein a first set of said channels transfers information in one direction while a second set of channels transfers information in the opposite direction, each of said nodes comprising:

a transmitting element for transmitting data to another node on said bus;

a receiving element for receiving information from another node on said bus;

a communication channel between said transmitting and receiving elements within each node;

transmitter state machine logic for controlling the synchronisation of said transmitting element;

receiving state machine logic for controlling the synchronisation of said receiving element; and

a storage area for maintaining the status of synchronisation of said bus.

2. A communication bus according to claim **1**, wherein the transmitter and receiving state machine logic are configured to report the link status of the communication bus.

3. A method of synchronising data communication over a communication bus, the communication bus having nodes that are linked by communication channels, wherein a first set of channels transfers information in one direction while a second set of channels transfers information in the opposite direction the method comprising:

transmitting data from a first node on the communication bus over a communication channel;

receiving data at a second node in the communication bus;

controlling synchronising transmitting the data with transmitter state machine logic;

controlling synchronising receiving the data with receiving state machine logic;

maintaining the status of synchronisation of the communication bus;

4. A method of synchronising data communication according to claim **3** wherein the method further comprises reporting link status of the communication bus.

* * * * *