<u>1394c Low Power Port Functional Requirements</u>

Colin Whitby-Strevens
Apple

11th August 2003

1 Introduction

This note reviews the 1394b low power operation for Beta mode ports, and discusses the functional requirements for 1394c (T-mode, or c-mode???). It does not consider 1394b bi-lingual ports as such ports are not relevant to 1394c "T-mode" operation. The 1394b port state machine is complex, but this note aims to identify only the details essential to discussing the requirements for low power.

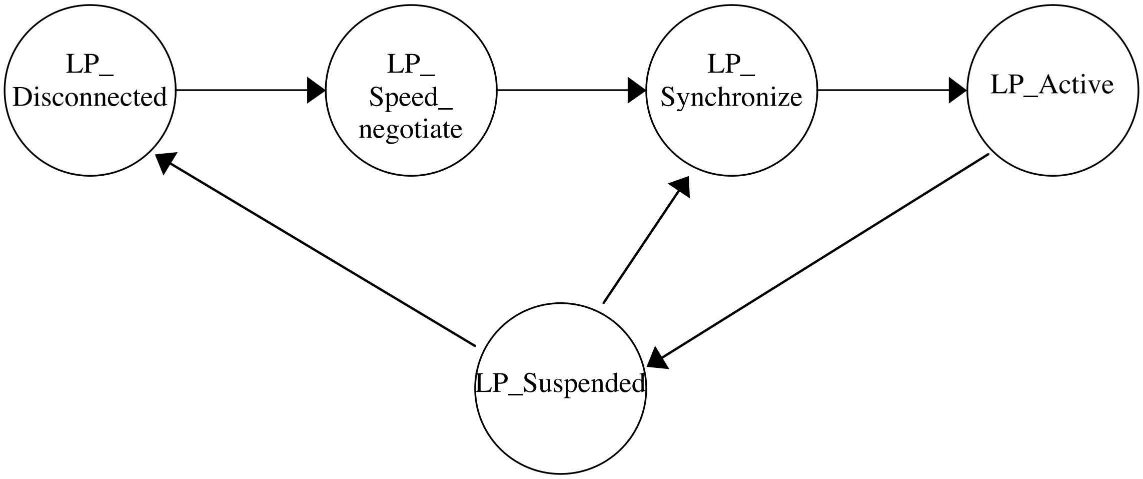
2 1394b Port power states

2.1 Overview

The main port power states are (my terminology, invented for this note):-

LP_Disconnected (includes disabled/disconnected)
LP_Speed_negotiate (a transient state on the way to LP_active)
LP_Synchronize (a transient state on the way to LP_active)
LP_Suspended (includes loop-disabled, standby and disabled/connected)
LP_Active

The permitted transitions are shown as follows:-



Finally there is a state known as "hard disable" in which the port does not engage in any form of signaling or reporting (of incoming signaling).

2.2 *Power state operations*

2.2.1 LP_Disconnected

This is the state that a port enters on chip power reset.

In this state, a port firstly ensures that a peer port perceives a disconnection by ensuring that no signaling is transmitted for approx 84 milliseconds.

It then actively seeks a connection unless “local_plug_present” is FALSE. It does this by transmitting “tones”. A tone comprises a 50-64MHz clock signal and is transmitted for 666 μsec. A tone is transmitted approximately every 42 milliseconds (approx 24 tones per second) – representing a 1.5% duty cycle.

Simultaneously, a signal detect circuit (an energy detector) listens for incoming tones. There is no signal integrity or anaysis performed on the incoming signal.

If a signal is detected, then the port starts up by passing through two transitional states, necessary precursors to entering the active state. The first is LP_Speed_negotiate.

2.2.2 LP_Speed_negotiate

In this state, the operating speed of the connection is negotiated between the two ports. This uses an extension of the toning system described above. The operating speed can be anything from S100 to S3200, depending on the media and silicon capabilities. There are rules in 1394b which place both high and low limits on the permitted operating speeds for a given medium. (This negotiation is probably not relevant to T-port operation.)

Having established the operating speed, the port then moves into a second transitional state – LP_Synchronize.

2.2.3 LP_Synchronize

In this state, the port starts transmission of 8B10B TRAINING arbitration state symbols (these use 8B10B codes as defined in the 1394b spec). These are used by the receiving port to acquire first bit lock (i.e. receive PLL synchronization) and then symbol lock (using K28.5 symbols) and then scrambler synchronization. When a port has acquired synchronization of its receiver, then it changes its transmitter to the transmission of OPERATION arbitration state symbols.

When a port is both transmitting and receiving OPERATION arbitration state symbols, it moves into the LP_active state.

2.2.4 LP_Active

A port in the LP_Active state is fully synchronized and is transmitting and receiving normal 8B10B symbols at the agreed operating speed. It stays in this state indefinitely, and only has one exit transition – to LP_suspended.

When a port enters the LP_active state on a new connection, the PHY state machine will first test it to see if this new connection results in a loop. It uses normal arbitration and signaling to carry out the loop testing function. If this is found to be the case, then PHY state machine immediately places the port into "loop_disabled" – which, from the point of view of low power operation, is one of the LP_suspended states. This is initiated from one end, which sends a protocol message to the peer port and then places the local port in LP_suspended, disabling transmission of normal signaling. The peer port receives the protocol message, passes it to the PHY state machine, which places the port into the suspended state when it detects loss of synchronization. Note that software is not involved in this process, indeed the PHY does not even need an active link layer.

If the connection does not form a loop, or if the port has been entered from (indirectly) the LP_Suspended state, then the port is used to carry normal 1394 bus traffic. This may involve starting with a "bus_reset". Note that "bus_reset" is totally transparent to low power operation.

A port can also enter the LP_suspended state if it detects sudden loss of synchronization (for example, if the user pulls out the plug). In this case, a 'force_disconnect" flag is set, and the port will visit LP_suspended only temporarily on its way to LP_disconnected. Loss of synchronization is detected if a sufficiently large number of "bad" 8B10B symbols are received in rapid succession.

Finally, a port can enter the LP_suspended state when commanded so to do by software. Software sends a PHY command packet addressed to a specific port on a specific PHY. This packet is sent with normal arbitration, and so is indeed received by all active PHYs. The addressed PHY performs some checks, and issues a response packet. If the response packet indicates that the command will be accepted, then it sends a specific control symbol on the port to be placed into LP_suspended, and then waits for loss of synchronization on the port. When it detects loss of synchronization, it places the port into the LP_suspended state. The peer port receives the symbol, and immediately ceases transmission and places the port into the LP_suspended state.

2.2.5 LP_suspended

Although a port has been placed in a low power mode, it is important for the 1394 plug-and-play functions that it can maintain connectivity status. Said differently, it is important to detect when a port has been disconnected (or, equivalently, when the peer port has been placed in hard_disable or the peer PHY powered off).

While in the LP_suspended state, the port issues connection tones in exactly the same way as described above under LP_disconnected – i.e. 24 tones per second, each a 666 usec burst of energy.

The port also maintains activation of its signal_detect circuitry, seeking incoming tones.

If the “force_disconnect” flag is set, then the port immediately transits to the LP_disconnected state, in which all signaling is disabled for a sufficiently long period for the peer port to detect this.

If a port “misses” two consecutive tones, then it transits to the LP_disconnected state.

There are two ways in which a port can transit to the LP_synchronize state (in which it will start to transmit training symbols at the port operating speed).

The first way is by software command addressed to the port.

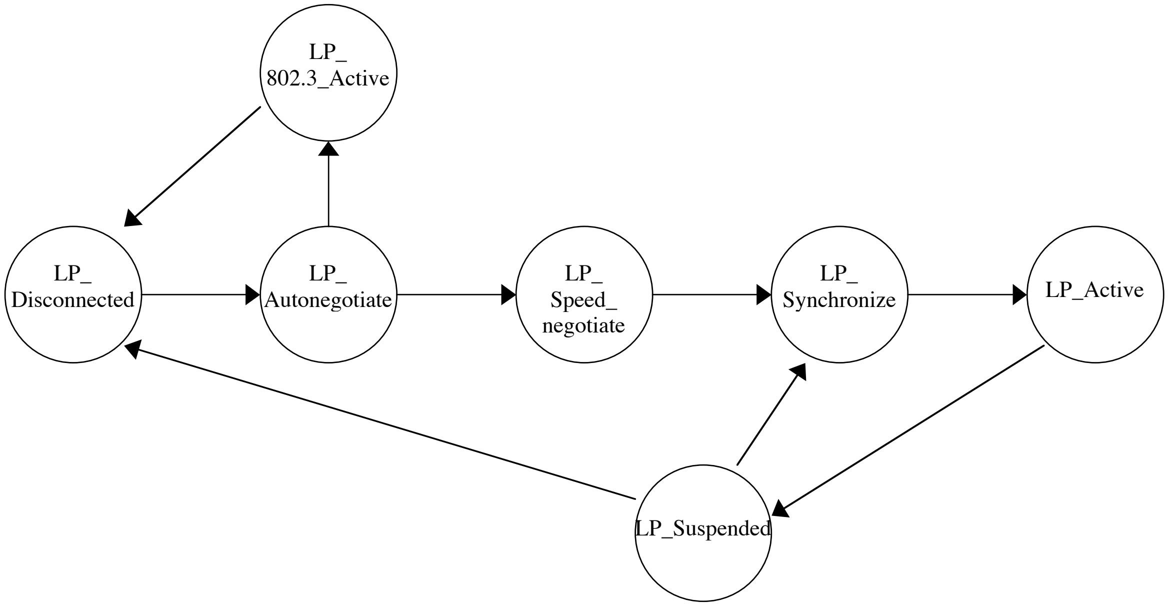
The second way is on receipt of continuous signaling (not just intermittent tones). This indicates that the peer port has transited to the LP_synchronize state and is sending a training sequence. If an incoming tone persists for more than 2 * 666 usec, then this is regarded as an instruction to resume from the peer port.

2.3 Disabled states

The disabled states operate within the same framework as given above, and, while making the port state machine more complex, do not affect the use of signaling as described above. (More could be written, but is probably confusing rather than enlightening).

3 1394c port power states

As a starting point for discussion, the following is suggested as an enhancement for 1394c.


LP_
802.3_Active
LP_
Disconnected
LP_
Autonegotiate
LP_
Speed_
negotiate
LP_
Synchronize
LP_Active
LP_Suspended