Accredited Standards Committee
X3, Information Processing Systems

Doc: X3T10.1/95a225R2
Date: February 2, 1996
Project: X3T10.1/1147
Ref. Doc.: SSA-TL2, 95a225R1
Reply to: Rick Born

To: X3T10.1 Membership
From: Rick Born Symbios Logic
Greg Kapraun Symbios Logic

Subject: SSA Speed Negotiation

Note: This proposal now stands on its own as other opinions and proposals are brought forward.

BACKGROUND

It is expected that as SSA evolves, newer and faster SSA products will want to be compatable with eariler products. An algorithm needs to be defined such that the speed that any two ports communicate can be negotiated between those two ports automatically. This algorithm should be defined in such a way that newer products do not necessarily have to implement the lowest speed (20MB/s) when such products determine to phase out slower speed devices. The alogrithm must also work with existing 20MB/s products that do not implement any speed negotiation algorithm. It is also preferable that the algorithm selects the highest speed that the two devices can communicate at.

PROPOSAL

Each link in the SSA Web negotiates its speed independently. This allows the Web to be heterogeneous; it is another aspect of spatial reuse. For example, a 40 MB/s adapter could be connected to a mixture of 40 MB/s and 20 MB/s disk drives. (For best performance the 40 MB/s drives should be nearer to the adapter than the 20 MB/s drives.)

The link speed is negotiated automatically at power-on. This is essential to allow communication to be established, which is a pre-requisite for configuring or resetting the Web. This proposal will guarantee communication at the highest data rate in common between two nodes if the link is capable of supporting communication at the highest common speed. Each port performs the negotiation algorithm in Figure 1 during the Disabled state, but only if the port is not operational. Negotiation is performed at power-on and after an unrecoverable link error or Total/Absolute Reset; it is not performed for recoverable link errors. Each port must complete speed negotiation before entering the Ready state (otherwise an RR may be lost). If a transmitter timeout occurs before going to Ready state, the port shall return to Disabled state and continue with the speed negotion. If character synchronization is lost during the Enabled state after an unsuccessful speed negotiation, the port shall go to the Sync state in order to begin the speed negotiation again. The operating speed is locked in the Enabled, Ready and Check states.

```
                        Enter Disabled state
                        from the Sync state
                        (Start sending DIS)
                               |
                               |            Y
                        Port Operational? -------------------*
                               |N                             |
                               |                              |
                        Set Tx speed = Maximum supported     |
                        Set Rx speed = Maximum supported     |
                               |                              |
                               |                              |
                               |                              |
                               |<--------------------------* |
                               |                           | |
                        Start Tx sync time-out             | |
                               |                           | |
*----------------------------->|                           | |
|                              |                           | |
|                       Start Rx sync time-out             | |
|                       (at current Rx speed               | |
|                        if Multiple Rx supported)         | |
|                              |                           | |
|        *-------------------->|                           | |
|        |                     |        Y                  | |
|        |                Tx time-out? -------------*      | |
|        |N                    |N                   |      | |
|   (Optional)                 |                  Lower    | |
| Rx time-out? <------- 4 consecutive            Tx speed  | |
|        |          N   commas decoded?             |      | |
|        |Y                    |                    *------* |
|     (Optional)               |Y                             |
|    Cycle Rx speed            |                              |
|    to next value      Set Tx speed = Rx speed               |
|        |                     |                              |
*--------*                     |                              |
                               |<----------------------------*
                               |
                        Wait until 200 characters
                        have been sent at current Tx speed
                               |
                        Enter Enabled state (Start sending FLAG)
```

Figure 1. Speed negotiation algorithm.

A simple and fast method of synchronization trial can be acheived if we vary the Tx sync time-out that a node will attempt synchronization based on its current speed. Standardize on a maximum time for 20MB/s SSA, and then reduce it by half for each higher speed step. Since the lower speed port will remain at that speed longer than the total sum of the dwell times at the higer speeds, the two will synchronize at the highest common speed. Table 1 illustrates this point.

Speed	Rate (MB/sec)	Tx sync time-out (+/- 10%)
1	40	32.768ms
2	Reserved	16.384ms
3	Reserved	8.192ms
4	Reserved	4.096ms
5	Reserved	2.048ms
6	Reserved	1.024ms
7	Reserved	512us
8	Reserved	256us
9	Reserved	128us
10	Reserved	64us
11	Reserved	32us
12	Reserved	16us
13	Reserved	8us
14	Reserved	4us
15	Reserved	2us

Table 1. Tx sync time-out for a given speed.

(Note: 20 does not need a time-out, if you drop to 20 you remain there)

The Rx sync time-out is dependant on the receiver implementation but must be larger than 4 characters and less than the Tx sync time-out if multiple receiver speeds are supported at the current Tx speed.

The above speed negotiation algorithm requires the two ports to become synchronized before begining the speed negotiation. This requires the addition of the Sync state to the state transitions.

```
                 Sync <----------------------*
                   |                         |
                   |                         |
                Disabled <--------*          |
                   |              |          |
                   |              |          |
           *--- Enabled           |          |
           |       |              |          |
           |       |              |          |
  Rx Error |     Ready            |          |
           |       |              |          |
           |       |             Y|      N   |
           *---> Check ----- Recoverable? ----*
```

Figure 2. State transitions

While in the Sync state, the outbound line sends a 10MHz signal for a minimum of 10us. When the port recognizes 10MHz on its inbound it then stops transmitting any signal on the outbound line for a minimum of 200ns (2 periods at 10MHz). Once it recognizes that the remote port has stopped transmitting the 10MHz signal, then the port transitions from the Sync state to the Disabled state and begins speed negotiation. If the port has been sending 10MHz for 100ms and not recognized 10MHz on its inbound line, it shall transition from the Sync state to the Disabled state and begin speed negotiation. (Note: this timeout is needed for communicating with present 20MB/s nodes and other single speed nodes that do not support speed negotiation and will not prceed through the Sync state on failed ERP or at Power-up). There is also currently a 25ms timeout defined in the Link ERP procedure and thus acts to justify the numbers chosen to begin the table. Upon leaving the Sync state the port will be in Privileged mode and the Operational Flag shall be cleared.

Upon exiting the Check state, the speed negotiation is performed if the port is not Operational.

While in Enabled state, the port shall go to Disabled or Sync state if the receiver detects a loss of sync or 10MHz is detected, respectively.

The remaining state transitions happen as previously defined in the standard.

Notes:

1) The Receiver must be able to decode either of the 2 valid comma characters (DIS or FLAG) during speed negotiation, since the remote port may have already entered the Enabled state.
2) The Transmitter starts at the maximum speed supported and it does not step down until the Receiver has determined the incoming speed from the remote node or until the Tx sync time-out has expired which defines the maximum amount of time a part will stay at one frequency before switching to the next frequency. The transmitter shall not skip any step-down speeds once they have been defined in the standard, or it ensures that the Tx sync time-out in the table is adhered to. Once the transmitter has reached its lowest supported speed it remains at that speed until sync is achieved or a code violation causes the device to detect an unrecoverable Link ERP and shall enter the Sync state again.
3) The Receiver cycles through the supported speeds indefinitely until speed negotiation is successful. (There is no value in terminating the negotiation process without establishing communication.) When Tx sync time-out has expired, communication at the higher frequency has been determined to be unlikely and the Tx speed is reduced to allow a possible connection at a lower speed.
4) There is no requirement for all future implementations to support operation at 20MB/s. It is sufficient for the two ports on each link to have one operating speed in common. For example, a 100/40MB/s node would `connect' to a 40/20MB/s node at 40MB/s.
5) The time-out for decoding 4 consecutive commas is determined by the receiver implementation. It should not be greater than 200 characters for consistency with the maximum time allowed by SSA-PH1 to acquire character synchronization.
6) SSA-TL1 specifies that the minimum duration of the Disabled state must be 200 characters. Since it is the number of edges that are presented in 200 DIS characters that allow a receiver to acquire character synchronization, the part should always guarantee that at least 200 DIS characters have been sent in the Disabled state at the current Tx speed prior to transitioning to Enabled state.
7) A Total Reset frame cannot be transmitted or received until speed negotiation has completed. To minimize the possibility that a node cannot be reset, the negotiation process should preferably be implemented in hardware rather than firmware.
8) If two 20/40 MB/s ports erroneously agreed to operate at 20 MB/s they would continue in this state indefinitely and performance would be degraded. This situation is unlikely but it could be detectedby the Master using the information in the Query_port_reply SMS; the Master could then force the speed to be negotiated again by issuing a Total_reset frame to one of the nodes.
9) The algorithm can support an implementation option whereby a port which is experiencing a high number of link errors forces the link to operate at a lower speed. To achieve this the port would simply cause an unrecoverable link error by entering the Disabled state, and during speed negotiation it would then start its Transmitter and Receiver at a lower speed.
10) In a heterogeneous environment theavailable inter-connections may not be designed to operate at the highest speed supported by the nodes. For example, the user may wish to plug some new 20/40 MB/s disk drives into an existing 20 MB/s backplane. Therefore a multispeed port should provide a configuration facility (e.g. jumpers or a software utility) for manually selecting the maximum operating speed during installation.