# Auto-Negotiation and Gigabit Ethernet?

Bill Bunch
National Semiconductor Corp.
Phone: (408)721-7274
Email: Bill.Bunch@nsc.com

# Auto-Negotiation Myths

- **Auto-Negotiation cannot be applied to Fiber Media**
- **Next Page function is broken/takes too long**
- **Auto-Negotiation over fiber must take as long as Auto-Negotiation over copper**
- **Reality: A new connector enables easy specification around these issues for Gigabit over fiber**

IEEE 802.3z Gigabit Ethernet Task Force 5-November-1996
Bill Bunch National Semiconductor Rev. 1.0

Auto-Negotiation is largely a logical means of exchanging 16 bit code words.

The explicit mechanism used to encode the 16 bits in terms of a transmitter and receiver are specified to be link pulses meeting specific timing criteria. However, the reasons for this are largely due to the nature of copper and the installed base that must be dealt with on RJ-45 connectors. Going to fiber changes the playing field.

The Next Page function is correctly specified and has been implemented and simulated by a number of vendors. Most recently this has been done by vendors providing 100BASE-T2 products as well as Isochronous Ethernet (which Auto-Negotiation was invented for in the first place).

Next Pages take a 'long' time on copper only due to the bit rate of the link pulse spacing itself and the ability of the management agent to supply data. In fact, even as it stands today, where Auto-Negotiation exists on both ends on the wire, many Next Pages can be exchanged in the time it takes to detect a legacy 10BASE-T device.

By using a different media/connector the elements that make Auto-Negotiation over copper relatively slow can be eliminated.

## Why Auto-Negotiation?

- **Auto-Negotiation provides a rich set of capabilities for configuring a link on start-up**
  - **Base configuration provides 8 simultaneous and independent 'features' (speeds, modes, etc. ) to co-exist**
  - **Basic Remote Fault support**
  - **Next Page ability allows arbitrary information to be exchanged**
    - **Specific Remote Fault type information exchange**
    - **Redundant link support**
    - **Vendor specific features interoperable with (soon to be) installed base**
    - **Forward compatible extension of co-existing technologies**
- **No need to re-invent the wheel!**

IEEE 802.3z Gigabit Ethernet Task Force 5-November-1996
Bill Bunch National Semiconductor Rev. 1.0

A significant amount of effort went into defining the Auto-Negotiation algorithm to be leveraged across multiple technologies, including non-copper based networks.

The Selector field was included to allow fundamentally different technologies to use the same mechanism, but with entirely different code spaces. For example, the standard already includes selector fields for both IEEE 802.3 and 802.9 with room for another 30 additional MAC subgroups/technologies.

Remote Fault capability was incorporated to allow a port which is able to discover a fault (e.g. receive pair disconnected -> link down on one side) to transfer this information back across the link.

While the basic Remote Fault mechanism only tells that a fault occurred, it does not say what type of fault occurred. For this, one could utilize the Next Page function. Some Next Page encodings already exist in the standard for indicating some specific remote fault types. The code space for additional remote fault types is extremely large.

However, the Next Page function is not limited to sending remote fault information. Code space exists to extend the number of technologies supported, add interoperable proprietary features, or exchange any set of information desired to facilitate configuring the link.

# Auto-Negotiation on Fiber

- **Link pulses and their timing characteristics are not relevant**
  - **Doesn't have to apply to a new media**
  - **Link pulses were chosen for RJ-45 LANs to support interoperation with legacy 10BASE-T**
  - **Transmitter and receiver of Link Code Words can be any mechanism and Arbitration State Diagram will function the same**
  - **No legacy equipment to encompass for Gigabit**
  - **Can change timer values to get desired configuration time**
  - **Arbitration state diagram Link Configuration proposal are logically equivalent**
    - **But...includes Remote Fault and Next Page**



There have been a few proposals for Auto-Negotiation over fiber previously, but relative to 100BASE-FX. One of these uses Auto-Negotiation as is by applying codes that result in 'link pulses' to be generated on Signal Detect. A simple circuit allows PHYs that support 100BASE-TX and/or 100BASE-FX as well as Auto-Negotiation to function without modification.

Another method proposed uses special sequences of 5B codes to transport the Auto-Negotiation information. FEFI is an example of current standard usage of this type of mechanism. Yet another proposal used this method and was intended to be a generic out-of-band communication mechanism multiplexed with data (specifically for Flow Control before the decision was made to do MAC based Flow Control messaging). This type of proposal is very similar to the current TSC proposals.

This basic concept allows a hybrid of the two mechanisms - leverage the Auto-Negotiation format and sequencing, but use 10B codes to transport the information.

# Reduction of Arbitration Machine

- **Break_link_timer adds significantly to copper based Auto-Negotiation time**
  - **The purpose of this timer is to ensure that 10BASE-T devices enter link down at the start of Auto-Negotiation**
  - **No need for this timer on fiber;Negotiation restarted by detection of C-code**
  - **Make break_link_timer_done=true (i.e. fix the time-out value to zero)**
- **Parallel Detection does not apply to Gigabit**
  - **Parallel Detection allows automatic detection where Auto-Negotiation exists on only one end on the link (i.e. pre-existing legacy systems)**
  - **Gigabit would require the link start-up on both ends (as does 100BASE-T2)**



The break_link_timer is between 1.2 and 1.5 seconds! This severely limits the time taken to bring down and reconfigure the link. However, the reason for this time is to account for legacy 10BASE-T devices that powered up in the Link Good state. This time ensures that a remote 10BASE-T device will be starved of both data and link pulses long enough to enter the Link Fail state. Therefore, data and link configuration information are never interleaved, avoiding additional design constraints. Gigabit has no need for such a time period because 1. there are no legacy devices to worry about, 2. the configuration mechanism would be required on both ends, and 3. the link is up prior to configuration info exchange.

Parallel Detection is the path through the Arbitration State Diagram which allows a non-Auto-Negotiation device to be detected. Here, Auto-Negotiation looks at the link_status coming from the PMAs, whose receivers are monitoring the signals on the cable in ‘parallel’ (hence the name). If the PMA indicates that it recognizes the signals by asserting link_status=OK, then a further qualification is made before configuration is completed. This has no relevance to Gigabit Auto-Negotiation since it would be required on both ends of the link.

## How long to Configure?

- **Time to exchange one LCW is approximately**
  **(14 LCWs) x (16bits/LCW) x (.8ns/bit) = 179.2ns**
  - **Max of ~89Mb/s rate including Next Pages (60,000 pages in 10 ms!)**
- **Link is up prior to negotiation**
  - **Will not wait in FLP LINK GOOD CHECK before completing**
- **Speed of the management entity driving next page exchange is the limiting factor**
  - **Next Page process is held up in NEXT PAGE WAIT state if ‘management’ agent does not have the next word ready**
  - **Not a problem for hardware implementations**
  - **Software based management could slow this down**
  - **Could specify timing bounds for time critical Next Page exchanges**



Let’s look at what holds up the Auto-Negotiation process.

First, there is the break_link_timer. Once this becomes zero, negotiation can begin immediately upon release from management.

Since Parallel Detection is not relevant and Auto-Negotiation is required on both ends of the link, the 16 bit Link Code Words (LCWs) are sent from both sides and the other path through the Arbitration state diagram is taken. A sequence of receiving three matching words, sending acknowledgment, and receiving acknowledgment ensures that an LCW is exchanged correctly. The speed of this process is a function of the rate of the chosen signalling method. Using FLPs, one LCW is sent every 24 ms (worst case). For Gigabit, the LCWs can be sent back to back for a rate of one LCW every 180ns!

After the exchange of the base page, if the Next Page bit was set by both sides (Gigabit could make this a requirement if desired), then each device drops down to the NEXT PAGE WAIT state until the management entity on both sides has loaded the next set of info. Depending on the implementation, this time might be fast or slow. For a purely software implementation, this time could be quite high and have an impact on the negotiation speed. Gigabit could specify a maximum negotiation time if this becomes a critical parameter. This would have the effect of requiring a particular speed of the management interface.

Once all pages have been exchanged, the FLP LINK GOOD CHECK state is entered and waits for link_status=OK from the PMA. Since this is true for Gigabit prior to the start of Auto-Negotiation, there is no further delay.

## Ease of Standardization

- **Could add new Optical Specification section to Clause 28**
- **Could add a Gigabit specific annex to Clause 28**
- **Could add a pointer to Clause 28 with changes/exceptions in a new Gigabit clause**

IEEE 802.3z Gigabit Ethernet Task Force
Bill Bunch National Semiconductor

5-November-1996
Rev. 1.0

The low frequency signalling chosen for transporting the configuration information was a function of the worst case cable type (CAT3) and legacy systems. However, we were careful not to preclude other signalling methods for other media…as long as they do not co-exist on the same connector.

Ideally, you want to have a single basic signalling scheme that can be leveraged across everything. However, there is no reason that new LANs that will never co-exist on the same point-to-point link (i.e. a Gigabit fiber transceiver hooked to a 100BASE-TX Auto-Negotiation device) need to follow the low frequency signalling scheme defined previously.

While Clause 28 includes the specifications for the transmitter and receiver of Link Code Words (LCWs) for RJ-45 LANs, it is a simple matter to substitute a new transmitter and receiver more applicable to a high bandwidth channel and meet the interface requirements of the Arbitration State Diagram (which controls the sequencing of Auto-Negotiation).

# Arbitration Interface

- **Map link_status_[Gig]=FAIL to detection of C-code**
- **Map link_status_[Gig]=READY to IDLE code detection**
- **Map link_status_[Gig]=OK to *valid* IDLE code detection**
  - **Potentially the same as link_status_[Gig]=READY if no hysterisis is required**

# Transmit Function

- **Disable C-code transmission when:**

  **flp_link_good = true + mr_main_reset=true + mr_autoneg_enable=false + transmit_disable=true**

**Otherwise:**

- **Transmit C-codes with contents of config register when transmit_ability = true**
- **Transmit C-codes with ack bit set when transmit_ack=true**
- **Transmit 6-8 more C-codes after complete_ack = true**
  - **Set ack_finished=true afterwards**

# Receive Function

- **Ignore incoming data when flp_link_good =true**

**While flp_link_good=false:**

- **Monitor received words for correct generation of:**
  - **acknowledge_match**
  - **consistency_match**
  - **ability_match**
- **Set flp_receive_idle =true if an IDLE code is detected**

## Conclusions

- **Leveraging the work done in Clause 28, Auto-Negotiation for Gigabit over fiber is feasible**
- **Gigabit Auto-Negotiation can be 'dynamic' on a time scale of <1 microsecond**
- **Auto-Negotiation provides remote fault and arbitrary information exchange**
  - **Can solve redundant links (i.e  out of band 'dynamic' configuration)**
  - **Can support additional standard functions**
  - **Can support proprietary features in an interoperable fashion.**
- **Ensures synergy with Copper based Gigabit Ethernet**

IEEE 802.3z Gigabit Ethernet Task Force 5-November-1996
Bill Bunch National Semiconductor Rev. 1.0

Auto-Negotiation provides a lot of value in terms of features and functions. In addition, there is now quite a bit of experience within the industry with regard to implementation details, models, simulation platforms, etc.

While Auto-Negotiation is not considered to be dynamic since it ensures that data and configuration are not interleaved (during the negotiation process), the link CAN be re-configured after a link has been established. For Auto-Negotiation as it stands, this is a relatively slow process and it can be argued that this process is not dynamic. For Gigabit Auto-Negotiation, this process takes <1 microsecond even including 5 Next Pages. Unless one requires a greater degree of dynamics, Auto-Negotiation is suitable for this application.

Since arbitrary pieces of information can be sent across the link at a rate of almost 89Mb/s, it is hard to imagine that we would have much trouble solving redundant links, buffered repeater configuration, repeater/node indications, or anything else that is felt to be important.