To: Richard Taborek

Important Characteristics of the word-striped Hari Proposal

A list of reasons, why I believe a word-striped Hari is an attractive technical solution for the 10 GbE and Fibre Channel application is summarized here:

- The word-striped Hari keeps a 4-byte transport structure intact through all levels and configurations of data transmission with a uniform orientation (parallel/serial). (For idles, skips, and initialization, the byte-striped proposal uses on the 4 lanes a combination of parallel and serial formats in a single transport layer resulting in a two dimensional operating space, which is at the root of many of the difficulties for defining suitable `code blocks' and arriving at simple models and implementations). The word-striped proposal does not allow any parallel formats in the serial transmission domain. No deviations from this structure are needed for the benefit of synchronization (byte, word, frequency compensation) regardless of the number of transmission lanes. So any protocol which is compatible with the 4-byte wide structure can readily be accommodated and others can be made compatible without much pain, as is the case for 10 GbE framing per Howard Frazier, without its skip and idle definitions.
- It has a much wider native skew tolerance making special deskew circuits superfluous and saving significant high-speed (baud and fractional baud intervals) circuit complexity and associated power consumption.
- Much simpler, less state-ful logic, much cleaner interfaces, and more flexibility in mapping link protocols onto the link coding, fewer special cases of the "can remove X bytes if and only if they a re preceded Y bytes and followed by Z bytes" variety.
- Idle or skip insertion is done independent of the coding, and independent of the number of lanes (An idle is a 4-byte word inserted into the stream of 4-byte words whenever necessary -- no variation/dependency in format vs. number of lines in the interface). These operations are done in the stream of deserialized, word boundary aligned words, in coded or uncoded form.
- Frame format and coding (except for disparity control) independent of the number of lanes.
- Word sync between every packet with minimum Ethernet IPG.
- Much greater flexibility in control sequences -- don't run out of control characters, since link control is done with control words instead of a single K character.
- Almost all the logic is done on a per-word basis, rather than a per-byte basis. -- Simpler logic clocking.
- It's very intuitive, and easier to describe, construct, and simulate than the per-byte interfaces.

- Doing line de-crossing (i.e. to get lines 0-3 to connect to lines 3-0 in mirrored order, with the de-crossing done inside the chips) is trivially easy, adds 0 ns of extra latency, and adds a very few extra gates. Line de-crossing on the chip can save significant area on the card in complex systems.

Detailed Reply to your note `Hari Byte vs. Word Striping', dated 26 Nov 1999

In a previous short note, I expressed my general disagreement and difficulties with the arguments presented by you on the reflector. As promised, this note goes more into details. In the following, I make just short references to your note for orientation purposes, and the reader of this note may find it useful, to first obtain a printed version of your note from the HSSG Email Reflector archive to better follow the arguments.

Figure 5 - Word striping for 10 GbE: Your Figure does not accurately reflect what was presented in the Ritter presentation on page 16. We use 2-letter symbols for the various types of words and make in that figure no statement with regard to what the sequence of bytes is. Indeed the sequencing of bytes may be chosen freely to obtain certain advantages as long as it is consistent, especially with regard to the location of the comma character. Your 4-letter word symbols such as KddS imply a certain byte ordering which is not a requirement. Also, by implication you suggest that 4 K characters must be present in an aligned format across the 4 lanes, which is also not a requirement and frequently not the case.

Hari Functions Common to Byte- and Word-Striping:

Technology Limits for 0.25 micron CMOS *:* Stating that something is within the limits does not necessarily mean that it is easy. At a 3.125 Gbaud serial rate, there may be painful compromises necessary in terms of yields for 6 sigma design, power consumption, signal to noise ratio and link distance across cards, connectors and back-planes. We are not aware if anyone stated that operation at *312 MHz* is generally difficult. What is true is that some functions, such as single step byte and word alignment for a single 12.5 Gbaud link, can be difficult with the Sync words which result from direct serialization of current Hari definitions. For practical reasons, those functions are performed with standard CMOS TTL circuits at a clock rate of 312.5 MHz. If Hari definitions unnecessarily complicate circuits outside the Hari domain and add to cost, they do become an issue of standards.

Striping Evaluation Criteria:

1. MAC stream mapping*:* Your contention that 4 MAC Words (16 bytes) must be received before data can be forwarded is not true. The serialization of the first byte of the first word can start as soon as the first byte is encoded which is no later than with byte striping. One parallel word interval later (3.2 ns), the next word starts on the next lane and so on. This is called staggered word striping and was merely mentioned orally by Dr. Ritter in his Nov. Kauai presentation in the interest of time,

and the relevant foils have not been included in the meeting minutes by mistake. Copies of 2 foils are attached to this note. The main body of his presentation shows non-staggered words so readers would have an easier time to understand how the deskewing works. Once one understands how a non-staggered arrangement works, it is easy to progress to the staggered version which is the actual proposed implementation. It is true that with staggered striping, there are still more latches, but this is offset by the lack of alignment circuits (see below under item 3). However, what really matters, especially for stand-alone transceiver chips which are usually I/O and power limited, is the power dissipation. The number of transitions from register operations is identical for both cases, so the power dissipation is comparable on this account. Overall, the word striping approach saves noticeable power because of the absence of skew correction circuits which all must operate at relatively high speed for the byte-striped case. Taking these differences into account, it is questionable whether the byte striped approach saves circuit area and a fair assessment would recognize that byte striping as described in your memo takes more power.

2. Striping Latency: It is true that word striping generates more intrinsic latency but it is not as serious as you described. At least for word striping, the latency necessary for skew compensation is absorbed by the systematic delay and not in addition as you claim. For any links over media, this added latency is negligible, so it is not important for 10 Gb Ethernet of Fibre Channel.

3. Skew Compensation: You aptly describe and reveal how complicated and high-speed circuit-intensive skew compensation with the current byte-striped Hari definitions can become. The word-striping approach has the significant advantage of *not having to do any of this* and being able to accomplish the deskew function with a selected fixed sampling point within a word interval. In principle, the same can of course be done with byte striping, if the required skew compensation range is reduced to less than half a byte interval peak to peak. Our proposal *does not require any fractional bit* skew compensation. You mention that the deserializers are able to present either individual or multiple code groups for compensation. Presumably you will need latches to do this. Since we do not need this function, perhaps we should be credited with a compensating circuit-count offset under item 1. It is inappropriate to penalize our approach with an extra latency for skew since the skew is absorbed by the latency provided by the word granularity of word striping. I am not aware of anyone ever claiming that there is no need to deskew when word striping. We did state that there are no dedicated *deskew circuits* required. Your claim that deskew is independent of striping granularity is demonstrably false, granularity affects the range over which certain techniques are applicable. Your assumption that deserializers are similar to TBI does not apply to the word striping technique. Our proposal does not require cross correlation between lanes at the deserializer level all the way to a shared 40-Bit output register. So simpler implementations can be realized at that level. Also, I do not know what you refer to with `extraneous high-speed logic required to perform deskew function

past the SerDes'. There is simply nothing there except a four-way multiplexer feeding data into a shared 40-bit register which may interface directly with the MAC or be part of a FIFO for clock difference compensation. Apart from the multiplexer, this appears to be about the same as for your proposal. -- So with regard to skew compensation in the range advocated for Hari, I see plenty of facts favoring word striping and none for byte striping.

4. Train-Up Sequence: Continued proper deskewing for the current Hari solution depends on maintenance of fractional bit, bit and byte alignments. Lost misalignments will cause errors, but I am not sure whether you can easily monitor proper alignment and regain proper alignment in normal traffic patterns. With our word striped approach this is very easy since any isolated comma on any lane gives an indication of the alignment status of that lane. A specified majority of misaligned commas on a particular lane is reported as diagnostic information and enables automatic realignment in normal traffic.

5. Data Processing Rate: No comment.

6. SerDes Width: The interface between the 4 deserializers and the MUX shown on Page 8 of the Ritter presentation is certainly internal to a chip and the width is up to the designer. Since we deal with CMOS circuits, the power dissipation is proportional to the number of output transitions per time interval which remains constant regardless of the interface width. Assuming comparable line loading, the power dissipation will, to a first order, not depend on the width of this internal interface. The byte and word alignment to 40-bit boundaries does not dictate a wider deserializer. A wider sync field is obtained by sequential checks (the comma must appear in a certain location at a certain time) and the required deserialized width for comma detection is governed by the data rate and technology speed. Classical deserializers for 1 Gbaud links do comma detection at one fifth the baud rate after expanding the serial stream to a width of 5; at 3.125 Gbaud, it will be done at a width of either 5 or 10, and at 12.5 Gbaud at a width of 40. For all but the 12.5 Gbaud case, the data stream is expanded to the desired final width after byte and word alignment. For clocking convenience, each word-striped lane will probably dump the data, a byte at a time, into an intermediate 40-bit register (12.8 ns cycle) before merging the 4 lanes into a single 40-bit wide stream (3.2 ns cycle). So word-striping has extra latches here equivalent to the extra overhead at the serializer, but the extras operate at a relatively slow rate (12.8 ns) and there are no dedicated circuits to align the deserialized output with the other lanes or to recognize multiple 4-byte sequences as required for byte-striped skip operations. The bus width at the output of the SerDes and the width of the buffer used for merging the lanes is really dictated by the speed of the technology and future implementations may well reduce the width to two bytes. The word-striped formats are compatible with narrower (2 or 1 byte) interfaces.

7. Even/Odd Alignment: Since the MAC interface is a 4-byte word, and everything except the Ethernet End of Frame (before padding) is aligned with these boundaries, I think one would want to maintain 4-byte word synchronism throughout. As a consequence, any 2-byte interface which an implementer might want to use, has to be odd/even aligned. This has nothing to do with error checking, except that a synchronous system with fixed timing after synchronization is generally more reliable and supports more reliable error checking by a CRC which is heavily dependent on proper alignments. Again, as long as these interfaces remain on a chip, the effect of width on power is minimal. Actually, a wider interface can normally use slower, lower power circuits.

8. PMD Clock Tolerance: You acknowledge that the same skip definitions could apply for the word-striped version as for the byte-striped version. But what has been defined for the byte-striped version is anything but simple because of rules of the "can remove X bytes if and only if they are preceded by Y bytes and followed by Z bytes" variety. For this reason, we have proposed something simpler, a universal special 4-byte ordered set which can be unconditionally removed when recognized and the pacing requires removal, or inserted in the interframe gap, a status condition which is easily recognized. I have absolutely no idea why you think a PMD would have to recognize hundreds of ordered sets for removal purposes. It is easy to define a unique 4-byte ordered set and if you want to, you would not have to check all 4 bytes. We can also automatically control the ending disparity of the ordered set within the same word, that is why we would prefer to place the comma character into the last byte of a word, but this is not essential - it enables faster synchronization without having to look for both polarities of the comma.

9. Running Disparity Processing: There is no significant difference in the coding area for byte or word striping, but there is a slight performance advantage for a word based codec because disparity prediction can be applied more effectively. With current technologies available to anyone, just four codecs are required for either case, the only difference is how they are ganged together which is utterly trivial. As you correctly point out, at the receiving end of each lane, the bit pattern must be checked for invalid characters and disparity violations and these circuits are identical to what is needed for disparity adjustment which requires just about 50 additional gates and the whole thing is less than a decoder with checks. It is also obvious that either a byte striped or word striped Hari can be connected to a scrambled or a 64B/66B coded link with comparable ease or difficulty. An 8B/10B coded 12.5 Gbaud link is also a viable option and both Hari versions (byte and word striped) have to adjust the disparity, but the byte striped version as currently defined has a known disadvantage because its Idle structure is ill suited for byte and word alignment at very high speeds (It requires more complex pattern detection circuits with added delay in a critical area.) In addition, it has a known problem with normalized dc-offset as recognized and described in the Fibre Channel Jitter Specification T11.2/Project 1230, Annex B. If the present byte-striped Hari is chosen, it would be my recommendation to translate the current Hari Idle for

transmission over the 12.5 Gbaud single lane to an Idle word resembling the Fibre Channel Idle, a translation which would not be required for the word-striped scheme.

10. Preservation of ordered sets: Perhaps our presentation did not include sufficient detail on this point because of time constraints. Two points must be added: Our proposal is compatible with any protocol which defines Idles or Skips as 4- byte words and presents the comma character always in the same byte position of the word. So FC perfectly fits the model, and 10GbE as defined by Frazier requires only a trivial change in the Idle. We also suggested some changes which would improve certain features of FC because we feel it can easily be done and we should always be on the lookout for improvements if the opportunity arises to implement them. Please note, that the reversal of the of transmission of the bytes of a word is done in the SerDes and is not visible outside the transmission path, so there is no problem with compatibility: The FC mappings can remain as they are except for the Skip word which you feel is necessary and I think the Idle definition should be thoroughly studied to get the best possible compromises in the spectrum for easy clock acquisition, tolerable electromagnetic interference and zero normalized DC-offset for long strings of Idles.

11. Preservation of existing SerDes designs: No Comment. We are searching for the best technical solution as a priority, preservation of existing hardware is of much lesser importance.

12. Logic Complexity: The issues raised here have already been addressed under item 1, 8, and 9. There is no requirement for 16-octet granularity contemplated. I think it is not fair to refer to circuits operating at the word rate (12.8 ns cycles) as 'high-speed' logic, especially if they are not even there, while ignoring extra circuits operating at fractional baud intervals (less than 100 ps ?) for byte striped deskewing. Perhaps you can explain why a PMD would have to recognize anything more than a skip and perhaps an Idle.

13. Power Consumption: This item has also been addressed above. You can take any CMOS technology manual and see that power dissipation is proportional to $C \times V^2 \times f$, where C is the capacitive load, V the supply voltage and f the switching rate. Applied to this example, it is obvious that the larger number of latches is offset by a proportional decrease in the switching rate. The same thing applies to the deserializer width as long as the connections remain in the unterminated CMOS domain. Your deskew logic all operates at a comparatively high rate, contributing a significant power component and these circuits are simply not present in our approach, not even in a slower version because the expected maximum skew does not exceed the natural skew tolerance of a word interval. For equal configurations such as Hari to 64B/66B or 12.5 Gbaud 8B/10B, the required disparity recalculations are identical and as we pointed out before, the disparity adjustment circuits are just a small appendix to the error check circuits which presumably are

present in either case. So there are good reasons to expect a byte-striped approach to dissipate noticeably more power than the word-striped approach.

14. Patent Protection: This item is really out of place in a technical evaluation, it may be a business consideration. While the use of this patent is easily detectable, it should be assumed that all hardware suppliers play by the legal rules and it is hard to see how anyone could efficiently and quickly build a Hari macro in either version without having to get a license for at least some patents.

Albert Widmer Phone: 914 945-2047 email: widmer@us.ibm.com

IBM T.J. Watson Research Center

Yorktown Heights NY 10598-0218