RZ 3426 (# 93580) 06/24/02
Electrical Engineering   4 pages

Research Report

A Highly Versatile Architecture for SDH/SONET Framers

Rolf Clauberg and Peter Buchmann

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

IBM Research
Almaden · Austin · Beijing · Delhi · Haifa · T.J. Watson · Tokyo · Zurich

A Highly Versatile Architecture for SDH/SONET Framers

Rolf Clauberg and Peter Buchmann
IBM Research, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract. A new architecture for SDH/SONET components and entire framers is introduced. The architecture allows building highly configurable SDH/SONET components suitable as system-on-chip cores as well as single-chip data aggregation framers with add/drop, cross-connect, and multiplexing functionality for a large number of line ports with different data rates.

Introduction. Although synchronous digital hierarchy (SDH)/synchronous optical network (SONET) technology [1] is normally considered as a mature technology, a series of new standards is coming out just now to enable next-generation SDH/SONET with efficient data transport support [2], and efforts are taken to simplify framer designs for cost effectiveness. In this article we will show that the unique features of SDH/SONET allow a previously unexploited way of designing highly compact and efficient SDH/SONET components capable of operating not just at one specific data rate but at a large range of data rates. These units can be used to design highly configurable cores for system-on-chip libraries and enable the design of single-chip aggregation framers for an efficient aggregation of data from a large number of line ports into a single data stream for a switch fabric or a higher-rate SDH/SONET frame. An important point exploited in the new architectures is the highly regular structure of an STM-N (STS-3N) SDH/SONET frame of 9 rows of bytes, with each row starting with 9 × N section overhead (SOH) bytes (administrative pointer bytes in the fourth row) followed by 261 × N payload bytes. The rows are transmitted serially one after the other, forming a 2430 × N byte data block.

The following functions must be applied for receiving or transmitting SDH/SONET data through a line port:

a. Serializing and de-serializing data at the line port
b. Alignment of received data to the byte structure of the frame
c. B1 parity-byte calculation over the scrambled frame data
d. Scrambling and descrambling the frame data
e. B2 parity-byte calculation over a specific range of the unscrambled frame data
f. Generating and monitoring SOH bytes
g. B3 parity-byte calculation over each virtual container
h. Generating and monitoring path overhead (POH) bytes for each virtual container
i. Mapping and extracting the payload bytes from SDH/SONET virtual containers

Next we will show how all these functions can be implemented in a frame-independent way.

Analysis of framer functions: The serializing/de-serializing of frame data is independent of the specific frame and depends only on the width of the data path of the functional units. The other functions can be grouped into three categories: (1) functions that modify the data within a specified range (d); (2) functions that calculate a value from the data within a specified range (c, e, g), and (3) functions that create or monitor the frame structure (b, f, h, i).

Frame scrambling is frame synchronized based on the generator polynomial $1 + x^6 + x^7$ for all STM-N or STS-N frames. A reset to the same start value always occurs with the first byte of the second row of the frame. For frames from STM-1 to STM-64 and STS-1 to STS-192, scrambling starts with the second row and continues until the end of the frame. Hence, a frame-scrambling unit can be designed that starts scrambling at byte count value 270 × N, and stops scrambling at byte count value 2430 × N using the frame-independent scrambling function. Fig. 1 show a scrambling unit designed to use two register values for triggering the events and loading these register values (270 × N and 2430 × N) from a configuration register during system startup. This unit can be extended to STM-256 (STS-768) by hard-coding some additional triggers but without additional registers. As frame descrambling is the reverse function of frame scrambling, the same unit can be used.

The second category of functions also only needs to trigger on specific, frame-size-dependent byte counter values where the functions start and stop. The B1 byte is calculated over the entire STM-*N* frame, and the only trigger value needed is the byte count of 2430 × *N* which defines the end of the frame and the start of the calculation of the next B1 byte. For the B3 byte the data range is the entire virtual container (VC) to which this B3 byte belongs. A corresponding VC byte count triggers completion of a B3 calculation and start of a new B3 calculation whenever its value is equal to the length of the VC in bytes. Again this is one single value, which can be provided by a register during system startup. The B2 bytes are defined as bit-interleaved parity *N* × 24 code with even parity for an STM-*N* frame. Accordingly there are 3 × *N* B2 bytes, and the calculation is done over the entire STM-*N* frame except the first three SOH rows. Hence, a B2 byte-calculation unit needs the values *N*, 9 × *N*, 270 × *N*, 279 × *N*, 540 × *N*, and 549 × *N* for defining the number of B2 bytes and the triggers that start or stop B2 byte calculation. As a result, all functional units of the second category can be designed according to Fig. 1, with not more than six register values to be configured at startup.

The third category of functions needs no calculation or modification actions, and the functions are entirely based on event triggers for the byte counter of the frame or the corresponding byte counters for VCs of the frame. All functional units needed can be designed according to Fig. 1, with the appropriate event trigger values being loaded during system startup.

Data-multiplexing architecture for SDH/SONET aggregation framers: So far we have considered the case that the configuration for the SDH/SONET framer is read at system startup and does not change during operation. The main advantages of such a framer would be its configurability at system startup and the suitability of its specific functions for system-on-chip cores with an enhanced range of applicability. However, with current fast on-chip memory it is possible to read the functional unit configuration, perform the necessary operations, and store all intermediate results in memory within a single clock cycle. This opens the way to aggregation framers, which are able to aggregate data from a large series of different frames into a single data stream or a higher-data-rate frame. The basic idea is to multiplex data received from a series of line ports onto a single data path with context switching of the data arriving at a functional unit from one clock cycle to the next. Fig. 2 shows a corresponding architecture. While data is being forwarded on the data path from one functional unit to another, the corresponding port address encoding is forwarded on an address bus in parallel to the data bus. The port encoding on the address bus determines the memory bank from which configuration and previous intermediate results will be read at the beginning of the current clock cycle and to which the intermediate results of the current operation will be written at the beginning of the next clock cycle. As configuration information and all intermediate results from the preceding cycle, in which the unit worked on the same frame, must be read within a single clock cycle, certain units will require parallel access to multiple memory banks per frame. An important point in Fig. 2 is that the line ports P1 to PL may be receiving and transmitting at different SDH/SONET data rates. Another point not visible in the figure is that for building the frame on the transmit (Tx) side of the framers, there must be a FIFO buffer per VC to interleave the VC data in the required way into the STM-*N* frames. This buffer is needed in each SDH/SONET framer, whether it uses the architecture described here or a conventional one. On the receive (Rx) side in Fig. 2 there is a dis-interleaving of the VCs in a frame just before payload extraction. At this point it is possible to write the data of a specific VC into the corresponding FIFO buffer on the Tx side rather than a buffer on the Rx side. This offers a very easy implementation of a SDH/SONET add/drop function in the framer. The same implementation also offers cross-connect functionality between the VCs from ports P1 to PL and SDH/SONET multiplexing functionality because VCs received on the Rx side can be interleaved into a higher-data-rate frame on the Tx side. We have performed detailed feasibility analyses of this architecture, including complete logic design of timing critical units, for two specific aggregation framers, an STM-64 (STS-192) framer with 16 STM-4, 4 STM-16 and a single STM-64 port, and an STM-256 (STS-768) framer with 16 STM-16, 4 STM-64 and a single STM-256 port. In principle, the STM-64 framer can also support 64 STM-1 ports and the STM-256 framer 64 STM-4 ports, but this would require a much larger chip package than needed for the chip owing to the number of required I/Os. These analyses show that e.g. the STM-256 framer is feasible with an approximate chip size of 150 mm^2 in IBM's Cu-11 technology if a channelization down to STS-1 SPE or VC-3 is supported. A chip size of about 220 mm^2 would also support the new virtual concatenation feature [2] with an embedded DRAM for storing the payload of 16 STM-256 frames as required by the standard. A comparison of the data-multiplexing architecture with the conventional parallel architecture will be given elsewhere [3]. Here, we only want to mention that the use of a single functional unit instead of multiple parallel units leads to savings in chip size that far exceed the small increase in size caused by the memory for configuration information and intermediate results in the new architecture.

In conclusion, the architecture described above offer new solutions for highly configurable SDH/SONET framers and for large, efficient aggregation framers with cross-connect and multiplexing functionality.

References

1 SEXTON, M., and REID, A. Reid: ‘Transmission Nntworking: SONET and the synchronous digital hierarchy’ (Artec House, Norwood, MA, 1992)

2 BONENFANT, P., and RODRIGUEZ-MORAL, A.: ‘Framing techniques for IP over fiber’, IEEE Network, July/August 2001, pp. 12–18.

3 CLAUBERG, R.: ‘Data aggregation architectures for single-chip SDH/SONET framers’, to be published in the IBM J. Res. Develop.

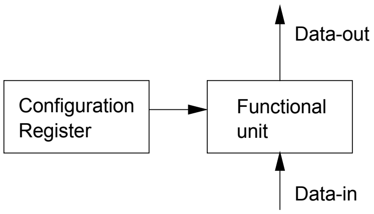

Fig. 1. Simple block diagram of functional unit that loads frame-dependent configuration information from a register

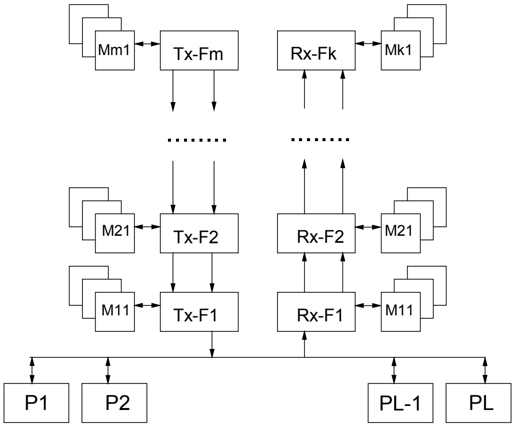

Fig. 2. Data multiplexing architecture with line ports P1 to PL, and functional units (F) on the transmit (Tx) and receive (Rx) side of the framer each connected to multiple memory banks (M)