A Multistandard Digital HD/SD Audio Multiplexer with Modular Ancillary Packet Substitution

Marco Crepaldi\textsuperscript{(1)}, \textit{Member, IEEE}, Daniele Franchino, Modesto Scavarda\textsuperscript{(2)}, and Mariagrazia Graziano\textsuperscript{(3)}, \textit{Member, IEEE}

\textbf{Abstract}—This work presents a multistandard digital HD/SD-SDI audio multiplexer (embedder) capable of inserting and replacing existing ancillary audio AES-EBU packets. The insertion of audio packets in SD-SDI video is achieved with an ad-hoc modular unit that shifts the existing audio packets in the video frame. This block enables the use of the entire multiplexer system in cascaded configuration along the same video line, a fundamental requirement in professional television studios. The embedder, implemented on a Virtex-5 FPGA, accepts 48 kHz AES3 synchronous audio, SMPTE 259M-C and SMPTE 295M full HD video, occupies a small area and it is installed on a system board comprising dedicated signal conditioning IC. The full system is tested for robust operation in audio/video production environments and for compatibility with older equipments.

\textbf{Index Terms}—Audio Mux, Packet shift, SD-SDI, HD-SDI.

\section{I. Introduction}

High-Definition Serial Digital Interface (HD-SDI or simply HD)\textsuperscript{(1)} and Standard-Definition Serial Digital Interface (SD-SDI or simply SD)\textsuperscript{(2)} are digital video standards commonly used in TV production environments regulated by the SMPTE (Society of Motion Picture and Television Engineers). These define the interface specifications for professional video transmission up to 3 Gbps video with different resolutions, picture rates and compression\textsuperscript{(3)}. The inclusion of additional data within a video frame is also possible through the use of specific data containers called ancillary packets. Ancillary packets, located in the vertical blanking and horizontal blanking space, can often contain AES3 audio\textsuperscript{(4)} [5] being transmitted synchronously or asynchronously with video images.

Audio is typically organized in four groups containing two stereo channels each, and transmitted in forms of normal and extended packets according to sample rate. In HD-SDI the audio packets have fixed length and their substitution can be easily achieved by replacing an entire ancillary packet, as specified in SMPTE 299M\textsuperscript{(6)} [6]. In SD-SDI video audio data is transmitted in variable length packets as described in the document SMPTE 291M [7]. Hence, SD standard SMPTE 272M [8] only specifies that variable audio AES sub-frames are inserted in the available ancillary data space provided that the necessary space for data and extended packets insertion is guaranteed. In addition, SMPTE 274M [9] imposes that all the ancillary packets are contiguous within the same vertical blanking line. A possible solution to solve the audio packet replacement problem consists of erasing a group's audio packets and shift all the other ancillary data in the same video line\textsuperscript{(8)}.

In fact, in TV production environments it is often necessary to efficiently control the multiplexing of AES3 frames into the ancillary data space, in many cases replacing audio ancillary data of a group with new samples, and sometimes including new audio streams among already present audio channels. This is not only useful during video and correlated audio production, but also when audio distribution is performed through the same video infrastructure to exploit the long cables interconnections allowed by SDI standard and avoid the installation of dedicated lines. This is one of the main motivations of our work.

Several products on the market can multiplex/de-multiplex audio packets in/from digital SDI video streams. In most of the cases they are based on dedicated chip-sets specifically designed and optimized for an application, thus maximizing performance and costs. Attempts to propose general solutions to multiple applications can be found in literature; for example in [10] a modular system based on boards that can be organized and programmed for several multiplexing and de-multiplexing applications is introduced. In [11] an end-to-end system for developing, broadcasting, downloading and executing digital video and audio is presented. Both solutions, though complete and complex, are based on specific hardware components, while system configuration and software allows flexibility and adaptability to manifold requirements. In [12] an IP for audio de-multiplexing based on SDI is proposed, implemented using a Field Programmable Gate Array (FPGA). None of these solutions can be used in production systems when the aim is distributing audio through video.

This paper thus describes a multistandard HD/SD-SDI audio multiplexer which is capable of substituting AES3 packets of a single group. For the most critical SD-SDI...
case we propose a specific modular control logic capable of shifting ancillary data and thus surgically erasing selected existing packets leaving untouched other audio data. The proposed system is flexible because based on an FPGA as suggested in [13] and because it is thought to support audio insertion in both SD and HD video standards, and as it is able to insert audio respecting, if necessary, audio streams already present in video frames. The design has been implemented on a Xilinx Virtex-V FPGA and extensively tested for robustness in production environments.

The paper is organized as follows. In section II audio and video standards are described emphasizing ancillary packets essential characteristics. In section III the system board architecture and the embedder core architecture is presented. Section IV shows the packet substitution unit useful to rearrange data in the SD ancillary data space. Section VI deals with robustness test measurements of the entire architecture and section VI concludes the paper.

II. VIDEO AND AUDIO STANDARDS

HD-SDI is now a widespread standard in video broadcasting and production studios. Similar is SD-SDI, the main difference being the higher sample rate and wider bandwidth of the former. The Society of Motion Picture and Television Engineers (SMPTE) formally defines the two standards HD (HDTV) and SD (SDTV) in SMPTE-292M [1] and SMPTE-259M [2] respectively. These two documents contain the interface specifications allowed through physical layer for HD and SD video on a 75Ω coaxial cable. The HD and SD standards are containers that define the serial video interface, but video data format itself is separately normed. Indeed, the possible picture and refresh rates are specified in other documents (called variants in SDTV), indicating also related parameters such as chroma subsampling compression. After HD or SD video is de-serialized by a dedicated receiver, data is parallelized in words, 10 bit long. In the following we are going to detail the essential points of these data.

Video words are organized in frames similar to those in Fig. 1, containing a total number of lines and a number of active lines. For HD frame rates can be 24, 25, 30, 50 and 60 Hz and the total number of lines can be 1125, 1250 and 750. Additionally, HDTV supports also the full frame 1080p at 3 Gb/s. For SD the total number of lines are 525 or 625 at 50 or 60 Hz interlaced video with a maximum rate of 360 Mb/s.

Only a part of the transmitted video frame contains image information, the active video field, which size again differs for HD (i.e. full HD, H=1920 and V=1080) and SD (i.e. PAL, 50i, H=720 and V=576). To identify the video starting and ending points two groups of reserved words are used: SAV (start of active video) and EAV (end of active video). By convention a line starts with an EAV, followed by a horizontal blanking interval, then the SAV and the active video as last.

EAV, as shown in Fig. 1 top left detail, consists of a sequence of eight words. Initially, a sequence of all ones (3FF) and two all zeros words (000, 000) is followed by word XYZ, containing information on the stream characteristics (not reported as not in the focus of this work); then two words specify the video line number, and finally two words contain the cyclic redundancy code (CRC) of the active video. SAV identifier is identical to the first four words of EAV. Except from sample rate and capacity, HD and SD lines are identical, even if correction code calculation in EAV is differently implemented. In addition, SD-SDI includes a special packet called Error Handler Identifier (EDH) that summarizes all the error information on the complete video frame including optional ancillary packets.

SMPTE 291M [7] defines how ancillary data (ANC), i.e. audio channels or control error packets, can be embedded in the blanking zones (vertical VANC and horizontal HANC), while dedicated standards specify their internal organization: for HD and SD video streams refer to SMPTE 299M [6] and SMPTE 272M [8], respectively. In Fig. 1 a detail (bottom left) on the ancillary data (ANC) organization is shown. An ancillary data flag (ADF) group of three 10 bit words identifies the packet beginning. A group of data identifiers words (DID) is then inserted, followed by a data count (DC) word indicating the number of used data words present in the
next payload (from 0 to 255). After the ancillary payload, a check-sum (CS) closes the ANC packet.

In case of ancillary audio data, the allowed format is AES3 [4], where samples are organized in sub-frames as depicted in Fig. 3. To each sub-frame (top detail in Fig. 3) an audio sample (24 or 20 bit) for each channel is assigned with a few preamble, control and information bit. Two sub-frames, related each to one of the two audio channels, define a frame. Frames are grouped in blocks of 192 to form the AES3 audio block.

SMPTE 272M [8] allows the insertion, for each SD video stream, of up to 16 AES3 audio channels (both 20 or 24 bit) organized in 4 ordered groups, numbered 1 to 4. In each group a further two-by-two association holds for stereo channels of the same audio source (typically 48 kHz sample rate, but 96 kHz, 44.1 kHz and 32 kHz are also allowed). Fig. 2 shows how an AES3 packed is inserted in the SD ancillary data payload identified in Fig. 1. Each AES3 sub-frame is organized in three samples, each depicted in the bottom detail of Fig. 3, where audio bit, control and information bit, are distributed (in particular two bits in the first sample specify the channel). When audio samples have 24 bit, it is necessary to insert an extended data packet immediately after the audio data packets. Extended packets, similar to data packets, contain the additional 4 bit not included in formers that can carry up to 20 bit samples.

For each channel, each audio AES sub-frame is split in four words for each of the four channels of the same audio group (details at the bottom of Fig. 4).

In both HD and SD, audio control packets containing information on the encapsulated audio are also transmitted in specific positions of the video stream. For example, in SD this information packet is mandatory if the sample rate is not the default synchronous 48 kHz AES3 audio. For example, the ancillary data space available in the PAL 4:2:2 13.5 MHz SD stream can be written in all lines except from the synchronous switching lines 5-7 and 318-320. Control packets are inserted only two lines after the synchronous switching point in the C channel.

III. System Architecture

The system is implemented on a single Printed Circuit Board (PCB) module with on-board power supply and dedicated integrated circuits for FPGA [14] inclusive of on-chip Rocket-IOs [15] for high speed serial-to-parallel and parallel-to-serial data conversion. For HD, the system handles videos conforming to the standards SMPTE 295M/274M, Full HD 1920x1080 and SMPTE 296M 1280x720, PAL. For SD, the SMPTE 259M-C 720x576, interleaved 50 Hz, 270 Mb/s, and the corresponding 50 Hz progressive are implemented. From the video viewpoint, the system works as a pass-through while embedding (erasing and/or selectively replacing) audio data in the ancillary space.

A system block scheme is given in Fig. 5. The Cable Equalizer and a Reclocker process the incoming digital video for conditioning the dynamic range of the on-chip Rocket-IO in the Virtex-5 [16]. The cable equalizer eliminates the distortions due to the transmission cable and adapts the signal dynamic range to the correct values for the FPGA Rocket-IO [17]. These two input/output units are used in the design, as receiver and as transmitter. The receiver chain is clocked with a dedicated off-chip low jitter oscillator GEN lock GS4911B that generates the video clock, a lock signal and initializes the operation of the RX Rocket-IO once video is detected. The GEN Lock is interfaced with the FPGA through SPI and generates also the 24.576 MHz audio clock synchronized with clock for AES3 multiplexing. The AES receiver (AES RX), off-FPGA with a dedicated analog-to-digital converter, requires a 12.288 MHz clock. This is
generated by the FPGA which divides the GEN Lock audio clock by a factor 2 with a dedicated asynchronous divider. The board accepts synchronous analog AES3 audio at 48 kHz, the sample rate commonly used in TV studios. The board powers all the IC, by converting a general 12 V supply to 3.3 and 2.5 V with dedicated step-down regulators. The embedding is controlled with specific external Switches that select the audio group; two Status LED indicate the locked standard, SD or HD. At the FPGA output a Cable Driver amplifies the power of the transmitted video signal.

The system must meet some robustness constraints to operate in critical conditions. Normally, in these kinds of environments the maximum performance and robustness are required with no upper bound. These features highly impact on the overall architecture which must be robustly conceived since the first design steps.

In particular the system shall be capable of:
1) **Synchronous switching:** (for both HD/SD) the embedder shall not lose synchronization and generate the minimum flicker within 1 video frame.
2) **Asynchronous switching:** (for both HD/SD) the embedder shall recover the synchronization in less than 100 ms.
3) **Power-on run:** if a signal is present at the input of the embedder both HD or SD video, or AES audio, the system shall not freeze.
4) **Forced-panic:** the system shall lock and be robust to any sudden cable disconnection and misuse.
5) **Video-IN to Video-OUT delay:** No more than 100 µs in all operating conditions.
6) **Automatic Video recognition:** SD or HD video is automatically recognized and locked.
7) **Fast HD/SD lock during video commutation.**

![Fig. 6. Multistandard FPGA embedder architecture.](image)

Fig. 6 shows the FPGA multiplexer architecture. After signal RXV is received by the built-in RX Rocket-IO clocked with the external GEN lock, a Data Recovery Unit (DRU) generates a clock enable signal that is used across the whole chip to validate each video word. The GEN lock recovers the 148.5 MHz clock from the incoming video that is used in all the video FPGA logic for HD video. For SD, a 297 MHz clock is used by the Rocket-IO and the 10 bit parallel words of SD video (running at 27 MHz, 1/10 of the serial data rate of 270 Mb/s) are oversampled by a factor 11×. For HD the enable signal is a half the clock frequency. For SD, the enable signal that lasts 1/(297 MHz) because video is downsampled by 11 times by the Data Recovery Unit (DRU). The RX unit with an internal descrambler decodes the input video and automatically detects the format, SD or HD.

The asynchronous FIFO synchronizes the clocks rx_side_clk, generated by the GEN Lock, and tx_side_clk, which lie on two separate domains. The width of this FIFO is very important for controlling the latency, the delay introduced by the system. Since the TX and the RX domains are supposed phase uncorrelated, the amount of data that fills the FIFO once video is locked depends on the operating conditions and on the whole system initial start-up. In particular it is related on how fast the GEN Lock acquires the signal and on eventual errors in the video stream generated during commutation. A general rule is to keep the width of the memory low but high enough to avoid flicker on the output video due to imperfect commutations on start-up.

In the TX clock domain the Packet Shifter adjusts the ancillary data for substitution for SD, while it is skipped in the HD case. The core of the embedder is the Audio MUX which multiplexes the audio samples from AES3 in the VANC and HANC space of the video frames. According to the locked video standard HD or SD, it generates the control and data packets, formats the AES3 stream, calculates CRC and inserts this payload in the available space. The AES RX decodes and parallelizes the AES frames from the stereo inputs AES3-in 1 and AES3-in 2: other asynchronous FIFOs are also needed to synchronize audio clk domain. Their width of 1024 samples is sensibly higher than those of the video due to the big difference between audio and video sample rates: 33 vs. 78.5 MHz (audio vs. video throughput must be controlled by definition). Many other signals are passed to the MUX from the SDI receiver chain such as locked, input standard (hd,sd), chip enable and controls.

In SD mode, the multiplexer output is processed by the EDH processor, (present also in the RX) that recalculates and inserts EDH with the new frame inclusive of audio. For HD-SDI a specific transmitter HD-TX with internal scrambler is used. After the Encoder, output video TXV is generated with the TX Rocket-IO.

**IV. Packet Shift**

The allocation of contiguous space for new ancillary data insertion is carried out by the Packet Shifter in Fig. 7. The shift mechanism is based on the use of two FIFO memories, one that buffers the input video stream and another that saves the ancillary data that must not be deleted. The unit has two dedicated Finite State Machines (FSM) that check the incoming and outgoing video streams. In addition, to permit packet deletion marking, a specific unit called Marker FSM is inserted as first block.
When an audio group has to be deleted, the Marker FSM unit substitutes in audio_groups all the DID with the codeword 180h used to mark the packets for deletion. The incoming FSM, with the aid of the EAV detector, that indicates the incoming EAV, checks whether the ancillary data space is present. Next, the FSM checks if a packet with deletion marker occurs. If so, the FSM saves it in STREAM FIFO; otherwise, if the packet has a valid ID (belonging to another audio group), the FSM saves the ancillary packets in the STREAM FIFO and in the ANC FIFO. This memory is activated only when a new data is inserted in such a way as its internal circuitry counts the exact number of stored words. Signal save blocks the FIFO Chip Enable signal ce that remains low as long as save is low. This process is repeated for each ancillary data packet present in the HANC space and also for each contiguous packets marked for deletion. This is fundamental to permit the processing of the extended audio packets which are inserted contiguously after the normal audio packets. The FSM stalls when a SAV is received and waits for the next line whenever an EAV occurs.

After the two FIFOs, the block ANC Cleaner removes any of the ancillary data present in the HANC space. The output of this unit is thus the video stream without any ancillary data packet embedded. At end of the chain the ANC cleaner inserts a black marker in the ancillary data space which is assumed to be the sequence 200h-040h. This avoids successive errors when EDH is calculated before packet retransmission. Whenever an EAV is present at the output of the STREAM FIFO, the Outgoing FSM, immediately after XYZ, substitutes the 020h-040h words with the output of the ANC FIFO. Its data is inserted contiguously by definition, since the Incoming FSM activates the wr_en signal only when the store of a new ancillary data is necessary. This dump process ends when the ANC FIFO is empty and the Outgoing FSM toggles the MUX which selects which of the EAV detector or ANC FIFO output is correct.

To avoid that the ANC FIFO is written during the ready phase of the Outgoing FSM, it is necessary that the STREAM FIFO is filled by not more than one video line length. To avoid this condition when a video stream is not present at the input, the Loader block gets the first EAV since a video lock event has occurred and blocks the read of the FIFO by a given number of clock cycles. Then it releases the block_read signal and freezes, waiting for a different video locked.

A graphical sketch of the operation of the unit is given in Fig. 8. In this example, the packet marked for deletion is A, while the packet B must be shifted and placed immediately after EAV. Data is first processed by the Incoming FSM, ancillary data stored in the ANC FIFO, video data is cleaned and Outgoing FSM replaces the black words 200-040h with the buffered ancillary data. This packet shift mechanism permits another important function which can be very useful for interfacing old equipments. These indeed, depending on their internal firmware, can generate video patterns with embedded AES3 audio. Unfortunately, at their design time the ancillary data space was not fully normed, hence the ANC packet generation is not fully conform to the current standard. For example, in some devices, the first ANC data in a video line is inserted starting 1 or more words after the XYZ word of EAV. This issue can be problematic in many newer equipment that can completely skip the processing of the embedded audio. With our solution, ANC packets are always realigned for being transmitted immediately after the XYZ word of EAV.

The shifter unit itself can be used in daisy chain configurations on the same FPGA design. Also, thanks to its modular...
SISO architecture it enables the whole embedder system use in Daisy Chain, allowing cascaded audio multiplexing and multiple audio data overwriting on the same video line. This fully scalable architecture permits the synthesis of designs with different Video IN to Video OUT latencies by changing the value of the terminal count of the Loader unit. For HD, the packet shift block is deactivated and the substitution job, much easier, is carried out by the Audio MUX.

V. ROBUSTNESS TESTS AND MEASUREMENTS

The architecture, described using VHDL, is synthesized on a Virtex 5 VLX50T FPGA and supported by a specifically developed PCB shown in Fig. 9. Synthesis results are shown in table I. The overall area ratio of the design is 23% thus allowing the integration of additional extra modules. The embedder is mostly composed of sequential logic (registers and LUT). The number of BlockRAM is not consistently high (21 out of 60). Also, the overall memory required is 720 Kbit out of 2160 Kbit. In [12] an audio de-multiplexer for SD-SDI proposed occupies 20% of the total logic elements and 30% of the available memory of 90 Kbit on an Altera Cyclone EPIC6 FPGA. The implemented embedder requires larger memory and logic occupation because it handles multistandard videos and it provides flexible embedding functions. However, the effort to multiplex data is higher than that necessary for de-multiplexing and this is translated into a higher logic utilization.

To verify the correct operation of the system multiple tests are run. These are divided in two different typologies, Video Robustness and Audio measurements. Also, other specific performance measurements are run.

A. Video Robustness Tests

The whole system has been extensively tested in presence of two video sources generating color patterns. The joint robustness and correct operation tests can be divided in three categories:

- SD-SDI switching (synchronous and asynchronous).
- HD-SDI switching (synchronous and asynchronous).
- HD/SD switching.

The first two series aim at determining any malfunction while switching the input video of the embedder within a single set of standard, SD or HD. These tests are important for checking the robustness of the logic with the two different video clocks. The last check consists of switching randomly between HD and SD to test the lock performance of the system. This is fundamental to validate the multistandard architecture performance.

Fig. 10 shows a summary of the test setup. In every test set-up a switching matrix controls the video inputs of the system, SD source and HD source. The Video synch. block generates a sync signal so video can be synchronized on demand by the Video matrix. The sources transmit the video patterns defined by EIA-189-A 75% and SMPTE RP 219-2002 for SD and HD respectively. Two different cables of length 5 and >100m are used for both testing synchronous and asynchronous switching channels. The two multiple output of the embedder board (Out1 and Out2) are connected to two HD and SD video analyzers and to a de-embedder, to verify both quantitatively and qualitatively the packet encapsulation process.

During the tests different problems have been encountered and solved. While synchronous switching is correctly ensured without flicker in equal length cables, with different lengths the delay mismatch causes image flicker that depends on the system startup. This issue can be corrected by increasing the width of the VIDEO FIFO (Async FIFO) used to synchronize the receiver and the transmitter clock domains. However, the maximum memory width is bounded (256 elements maximum) because the use of higher widths increases the random lock delay introduced by the system since VIDEO FIFO is unpredictably filled. For HD, since the system runs at 148.5 MHz against twice the speed for SD (297 MHz), all the FPGA logic locks faster because delay margins are larger and internal logic settle faster.

<table>
<thead>
<tr>
<th>Logic</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slice Registers</td>
<td>4369</td>
</tr>
<tr>
<td>Slice LUT</td>
<td>4720</td>
</tr>
<tr>
<td>Route thrs</td>
<td>482</td>
</tr>
<tr>
<td>Number of occupied slices</td>
<td>22300</td>
</tr>
<tr>
<td>Number of bonded IOs</td>
<td>39</td>
</tr>
<tr>
<td>Number of BlockRAM/FIFO</td>
<td>21</td>
</tr>
<tr>
<td>Number of 32k BlockRAM used</td>
<td>17</td>
</tr>
<tr>
<td>Number of 18k BlockRAM used</td>
<td>6</td>
</tr>
</tbody>
</table>

**Fig. 9.** Photograph of the embedder board.
after RX rocket IO and GEN Lock are resynchronized. No countermeasures are taken in the case of HD as VIDEO FIFO length 8 is suitable without locking performance compromises.

Finally, the hardest test, consisting of random asynchronous switching between any HD or SD source, implies other system-level optimizations; in particular, it leads to the necessity of making the receiver framer resynchronize, and to the requirement of increasing counters of modules for video standard detection (triple autorotate). System operation is verified also in presence of 4 multiple cascaded multiplexers embedding a separate audio source each.

B. Audio Measurements

By using the same test conditions for Video Robustness, functional and quality verification of the embedded audio is investigated during synchronous and asynchronous switching. To verify the presence of high quality audio in the ancillary data space the output of the de-embedder is analyzed.

The embedder does not generate distortions on the embedded audio data since all source AES packets are correctly embedded in the video stream. The absence of transmission errors is observed by checking the correctness of video fields and EDH in the HANC space with the SD and HD analyzers. The complete multiplexer board, including power supply, the digital embedder and the off-FPGA AES RX with dedicated analog-to-digital converter, has been benchmarked using a 1 kHz audio tone applied to the analog AES input with an SD and HD video in. Fig. 11 shows the FFT at the output of the de-embedder. The overall 2nd harmonic distortion of the full embedder/de-embedder chain is below -80 dBu thus ensuring high audio quality. This measurement validates the audio features of the complete board, comprising the presented FPGA design and the external AES audio input subsystems clocked by the on-FPGA clock divider.

C. Other Measurements

Another fundamental measurement regards the video delay introduced by the embedder. System latency shall be as much controlled as possible since it is considered by professionals before broadcasting. The overall delay shall not exceed a single video line. For example in a news channel, where the multiplexer is only a part of a more complex infrastructure, audio and video are realigned by delaying the audio at the source thus ensuring lip sync [8]. If the delay introduced by a unit is uncertain, its elimination is impossible because it depends on the system operating conditions. Also, the compensation of an overall infrastructure delay lasting more than a video line is technically hard.

To run this measurement a digital-to-analog SD video converter is connected at the output of the embedder. This transforms the digital video into analog domain with a fixed 10 µs delay and enables the measurement of the TXV-to-RXV system latency. The measurement involves the analog converted TXV video and the corresponding analog RXV source. Fig. 12 shows the measured TXV-to-RXV delay occurring at line 1 for a 720x576, 50 Hz, PAL video in: with the SD substitution unit, overall latency is 14.3 µs, i.e. 4.3 µs due to the embedder and a constant 10 µs offset due to the video converter latency. Multiple measurements showed a variation of 4 µs, a value that is tolerable in TV production environments. The multiplexer operation is verified in SD mode by recording the audio embedded videos with a Sony PDW-HD1500 professional disc recorder.

VI. CONCLUSIONS

This work describes a commercial multistandard HD-SDI/SD-SDI audio embedder used in professional TV production systems. It allows a modular substitution of existing multiplexed synchronous audio groups sampled at 48 kHz. A specific modular unit enables ancillary data shift and erase in the SD-SDI video frames thus enabling the use of multiple cascaded multiplexer along the same video line. The system accepts synchronous 48 kHz AES3 audio and handles up to
1920x1080 Full HD video. The system has been extensively tested in different operating conditions for robustness and reliability. The multiplexer system occupies an area of 23% when synthesized on a Virtex-5 VLX50T FPGA.

VII. ACKNOWLEDGMENTS

The authors thank all the staff at S.E.TEL Telecomunicazioni S.p.A. for all the help and support for the development of the project. The authors also thank Paolo Gatti [18] for his help and effort especially during system tests.

REFERENCES