

High Performance M2LC System

Michael E. Aiello
9/16/13

<mailto:michael.e.aiello@comcast.net>

Revision A
8/24/2014

Revision History:

Revision A (8/24/2014)	Added <i>Appendix A – Fault Redundancy and Uninterruptible Operation</i>
------------------------	--

Abstract

The discussion of a high speed communication medium for use in the control of a plurality of multi-level *modular multilevel converter* cells (M2LC cell). Each M2LC cell contains a field programmable gate array (FPGA) used for control and feedback. By taking advantage of two or more SERDES elements found in modern FPGA components, high speed communications between associated M2LC cells and a central controller module (CTRL module) can be created simply by linking these M2LC cells and controller module with glass optical fiber (GOF). Additional signaling using plastic optical fiber (POF) for controlling a battery and for providing optional M2LC bypass is also discussed. Also described is, a provision to allow the M2LC cell to be completely self-sustaining by using a lithium-ion battery to start operation and then rely on the power provided by an embedded current transformer attached to the underlining bus of the M2LC cell to sustain control power indefinitely. This allows the M2LC cells be stacked to a very high voltage potential. Further, a discussion on the encapsulation of the control and drive electronics of the M2LC cell to protect against catastrophic failure in the event of an uncontrolled IGBT short circuit insuring that upstream and downstream communications to adjacent M2LC cells can be maintained. Finally, a discussion on various connection topologies using one or more CTRL modules and M2LC cells that demonstrate systems that can be created with a high degree of fault tolerance.

Background

This application discloses an invention which is related, generally and in various embodiments, to a modular multilevel converter (M2LC) system employing high speed serial communications elements connected between associated M2LC cells and one or more central control modules (CTRL). This inventions exploits the fact that a typical M2LC cell employe at least one FPGA for control and feedback. Typically, the CTRL module would also contain at least one FPGA.

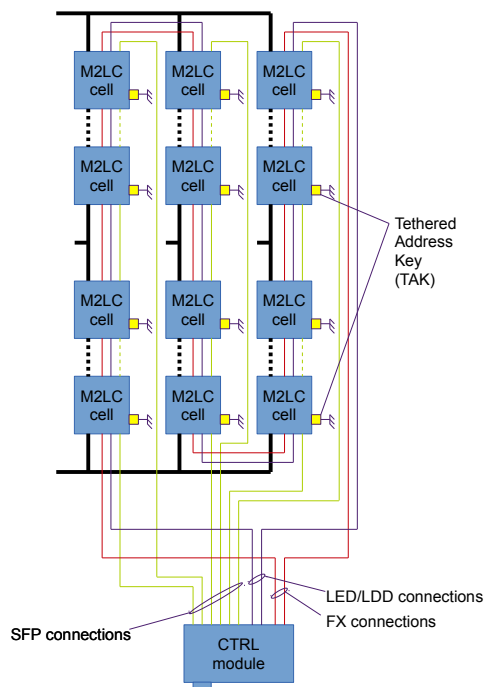


FIGURE 1 Block diagram of a three phase M2LC system.

Devices such as those provided by the Altera Cyclone IV GX and Xilinx Virtex 7 FPGA device family contain SERDES transceivers supplied as small ASIC cores embedded within the programmable logic. The serialization circuitry along with the physical interface are provided within the FPGA. This interface is designed to connect directly to copper media as well as fiber optical devices such as *small form-factor plug-gable* (SFP) modules. Typically, the minimum operating frequency is 1.25 GHz. For an M2LC application requiring electrical isolation between adjacent modules, only fiber optical GOF connections using SFP or similar devices would be applicable. SFP modules are engineered to be highly immune to external EM radiation. This along with the fact the SFP modules are designed to connect directly to the SERDES connections of the FPGA using simple PC board *strip-line* techniques, makes this circuit an ideal candidate as a communication media in power converter systems.

A block diagram of the proposed M2LC cell is shown in Figure 2. This cell represents a configuration for the generation of three levels of voltage. The discussion that follows can be applied to a more typical cell configuration that provides only two levels voltage.

Three optical fiber interfaces are shown in this figure. The SFP connections represents the SERDES GOF duplex interface which operates at a minimum of 1.25 GHz. This connection provides the control communications between the CTRL module and M2LC series-ed cells (see Figure 1). The connections labeled FX provides optional bypass control of the M2LC module in the event of loss in communications on a SFP link. This link utilizes POF operating at 100 MHz in full duplex mode. The connections labeled LDD/LED represent a single series-ed POF link that provides start-up sequencing of the battery circuit.

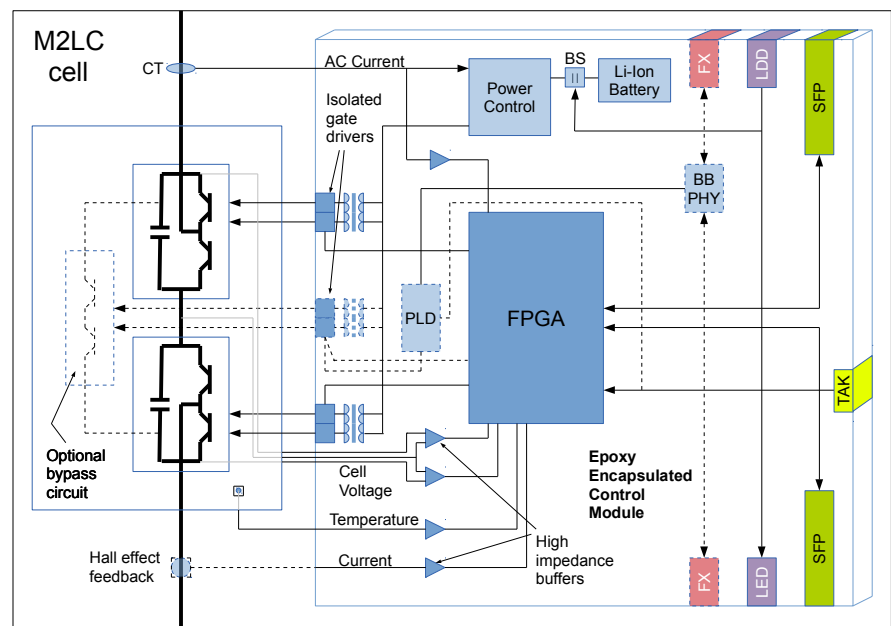


FIGURE 2 Functional block diagram of an M2LC cell.

The connection labeled TAK represents an address key socket. A mating plug specific to each M2LC cell is tethered to the cells mounting hardware. The typical techniques in modern communications of providing software enumeration to identify a specific M2LC cell would be considered much too dangerous to use under the conditions in which the system operates.

There are six isolated gate driver circuits, four for controlling the two multilevel half bridges and two for the control of the optional bypass switches. Each driver is supplied by an isolated transformer secondary and controlled with a high voltage opto-coupler. The output side of the opto-coupler is protected with a voltage clamp and fusing so as not to compromise the LED portion of the opto-coupler if high voltage is impressed on the coupler due to a fault condition in the power stage. Also, the primary side of the driver transformer is designed so that it can be disabled if a short circuit is detected on the secondary side of the transformer.

Cell voltage feedback is returned through high impedance, high voltage differential amplifiers which by their nature provide adequate fault isolation. However, the current transformer (CT) and hall effect feedback transducer are connected with low impedance buffering which requires they also be provided with fusing and voltage clamping to control ground.

In addition, the driver circuits and entire control circuitry and control power supply (including the battery) is encapsulated with an epoxy compound to protect from arcing conditions and inevitable plasma discharge caused by the short circuit fault of the capacitors in the multilevel bridges.

These protections described above are necessary so that any catastrophic fault conditions on the secondary side of the drivers and feedback elements cannot *take down* the FPGA, SFP transceivers and control power supply, and thus the communications to the upstream and downstream M2LC cells.

A diagram of the CTRL module is depicted in Figure 3. This module along with the series connected M2LC cells (as shown in Figure 1) provide all the necessary control electronics for an M2LC system. As will be shown later, the six SFP transceivers can be used in a multitude of ways for connecting to the M2LC cells. Figure 1 depicts a scheme for providing an optional redundant loop to each of the three phases of an M2LC system. Note that the CTRL module is fabricated as a small board containing a PCIe edge connection..

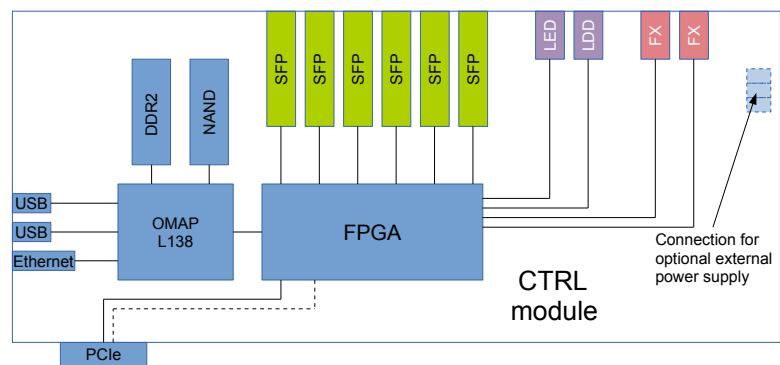


FIGURE 3 Functional block diagram of the CTRL module.

However, the typical setup would for a system where the CTRL module would *stand-alone* and utilize only the USB and Ethernet communication ports to the outside world. For this case, the PCIe connection would not be used. A small control power supply would be connected to this board to supply power to the control and communications ports¹. The OMAPL138 ARM/DSP is chosen as the controlling processor on this board. The ARM would typically run embedded Linux and would be used to communicate to the outside world using Ethernet or USB. The DSP would be responsible for running the power control algorithms².

For the discussion that follows, the focal point are the FPGA's and how they communicate between the M2LC cells and CTRL modules through the fiber optic interfaces. There are six SFP transceivers on the CTRL module that can be configured in a number of ways for communications to the M2LC modules. Figure 1 proposes one topology using a redundant loop for each of the three phases of the M2LC system.

A block diagram of the FPGA used on the CTRL module is shown in Figure 4. In this diagram only the connection to the SFP transceivers are shown. The other optical connections (labeled LED/LDD and FX) play an ancillary roll in the communications between the CTRL module and the

¹ Other configurations for the CTRL module are outlined as the end of this document.

² A detail discussion on the operational aspects of the OMAPL138 ARM/DSP processor is beyond the scope of this document.

M2LC cells and will be discussed in separate sections below. As shown in Figure 4, the FPGA provides modulation and control, and system time keeping along with one or more methods to construct packets that transfer this information back and forth between the CTRL module and M2LC cells.

The source and destination ports for the processing of these packets can be either the DSP processor on the OMAP, the PCIe port or both depending on the system configuration. The type of packet information is broken down into two groups, real and non-real time. Real time packets are those generated by the blocks SYSTEM TIME PACKET GENERATION and MODULATION LOGIC AND PACKET GENERATION. Non-real time packets are transferred directly to and from the OMAP DSP processor by way of the OMAP PACKET GENERATION AND RECEPTION INTERFACE block. The PCIe PACKET GENERATION AND RECEPTION INTERFACE block handles both real and non-real time packets. This will be explained in more detail later.

The SYSTEM TIME block is responsible for providing the cyclic rate in which the real time packets are to be sent to the M2LC cells. It is also responsible for updating the boundary stamp block (BS) which is explained in more detail below. All incoming and outgoing packets to the M2LC cells pass through FIFO buffers (TX FIFO and RX FIFO) and are routed to the various packet processing interfaces by way of the CROSS POINT ROUTER AND ARBITRATION LOGIC block³. Real time packet processing is initiated by the QUANTUM EVENT GENERATOR block.

In a similar fashion to that provided on the CTRL module, each M2LC contains an FPGA with integrated SERDES transceivers. Unlike the CTRL module, there is no *hard* processing engine like the OMAPL138 within the M2LC cells. However, they may or may not be a *soft* core within the FPGA⁴.

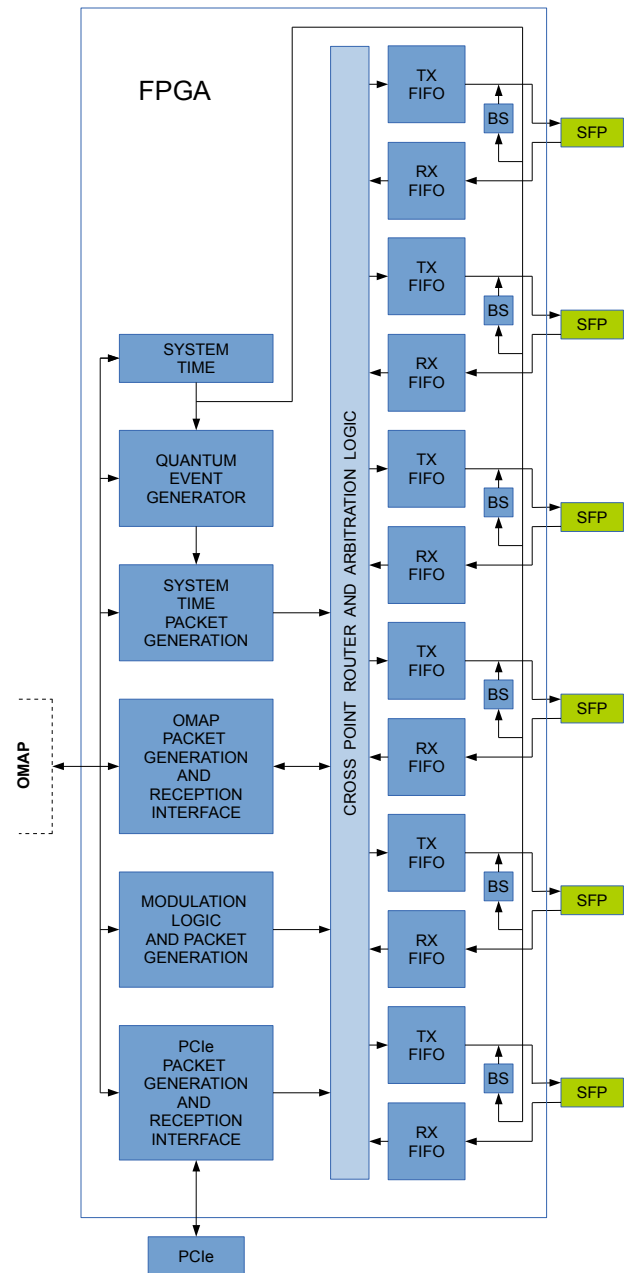


FIGURE 4 Block diagram of the FPGA within an CTRL module.

A block diagram of the FPGA contained on each of the M2LC cells is shown in Figure 5. Again, like the CTRL module FPGA, the fiber connections labeled LED/LDD and FX are of secondary importance to the discussion at hand and will be discussed in separate sections below. The block components shown in Figure 5 are a simplified depiction of the processing elements required for

³ A detailed discussion on how packets are processed by the CTRL module is beyond the scope of this document.

⁴ Candidates for this are Microblaze (Xilinx FPGA), Nios (Altera FPGA) or a third party core such as the ARM Cortex-M. The use of these cores are discretionary, and for the most part usually not necessary.

sending and receiving discrete packets between the CTRL module and the associated M2LC cell⁵. Like the CTRL module FPGA shown in Figure 4, FIFO elements and boundary stamp blocks (BS) are found in transmit and receive paths of the port connections. In addition boundary clock elements (BC) are also present in the FPGA of Figure 5.

The boundary stamp and boundary clock elements together perform time synchronization between the CTRL module and M2LC cells. The *boundary clock* extracts time information from an incoming packet and updates the system timer using a technique that does not cause an abrupt change in the control logic state-machines. The *boundary stamp* extracts time information from the system timer and inserts into an outgoing packet. Note that this insertion/extraction of time into and out of the packet always occurs as the packet arrives or departs the transceiver elements.

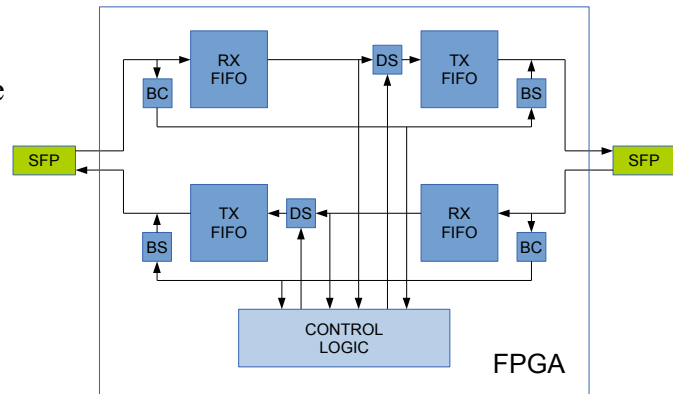


FIGURE 5 Block diagram of the FPGA within an M2LC cell (Option A).

The rate in which the packets arrive, are processed and then resent does not effect the time synchronization mechanism⁶. This form of adaptive time synchronization is known as *IEEE-1588 clock synchronization*.

The elements labeled DS represent data pipe selectors (multiplexers) where one of two input paths are *steered* to a single output path. In Figure 5, incoming packet can be processed and not sent through, or sent through and not processed, or both processed and sent through. This form of packet processing (label as Option B, *discrete packets* in Figure 7) is ideal for high bandwidth, peer-to-peer communications, and non-synchronized redundant communication sources⁷. The FIFO's depicted in Figure 5 (and Figure 4) act only to *hold* for a short time, a packet that cannot be processed due to a conflict within the internal pipeline.

As an alternative, a packet processing scheme can be derived that reduces the complexity of the processing elements on both the CTRL module and M2LC cell FPGA's. This however is at the expensive of reduced bandwidth and the increased complexity in the implementation of redundant CTRL module backup. A diagram of the processing elements on the M2LC cell FPGA is shown in Figure 6 (the equivalent representation of components required in CTRL module FPGA to implement this scheme is not presented here).

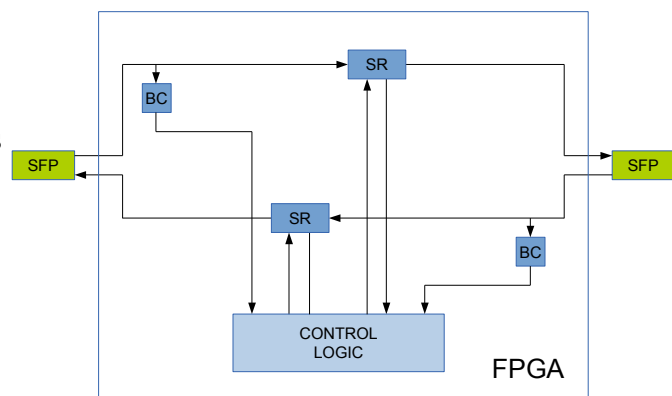


FIGURE 6 Block diagram of the FPGA within an M2LC cell (Option B).

This approach is based on the generation by the CTRL module of one large packet with fixed size that passes though each M2LC cell using the upstream portion of the SFP link. The last M2LC cell

⁵ Refer to Figure 6, Option B.

⁶ Timing accuracy depends on how well the clock crystals on the CTRL and M2LC modules are matched.

⁷ There is no need to alter the methods in which packets are processed by the M2LC cells when adding an additional CTRL module.

(usually the last physical cell in the chain) must then return this packet back to the CTRL module through the downstream portion of this link. As the data stream progresses upstream, information is extracted from the stream at elements predefined within the packet. As the packet returns downstream, information is added to the stream at elements predefined within the packet⁸. In a redundant loop configuration, the circulation of the packet reverses after each cycle. If a M2LC cell fails, the packet is circulated on each side concurrently, stopping at the cell that is disabled or faulted.

As shown in Figure 6, this protocol (label as Option A, *single circulating packet* in Figure 7) requires only a small shift registers to insert and extract data as the packet progresses through each of the M2LC cell FPGA's. Also, since the packet has a fixed time relationship as it enters and exists the FPGA, only the *boundary clock* (BC) elements are need.

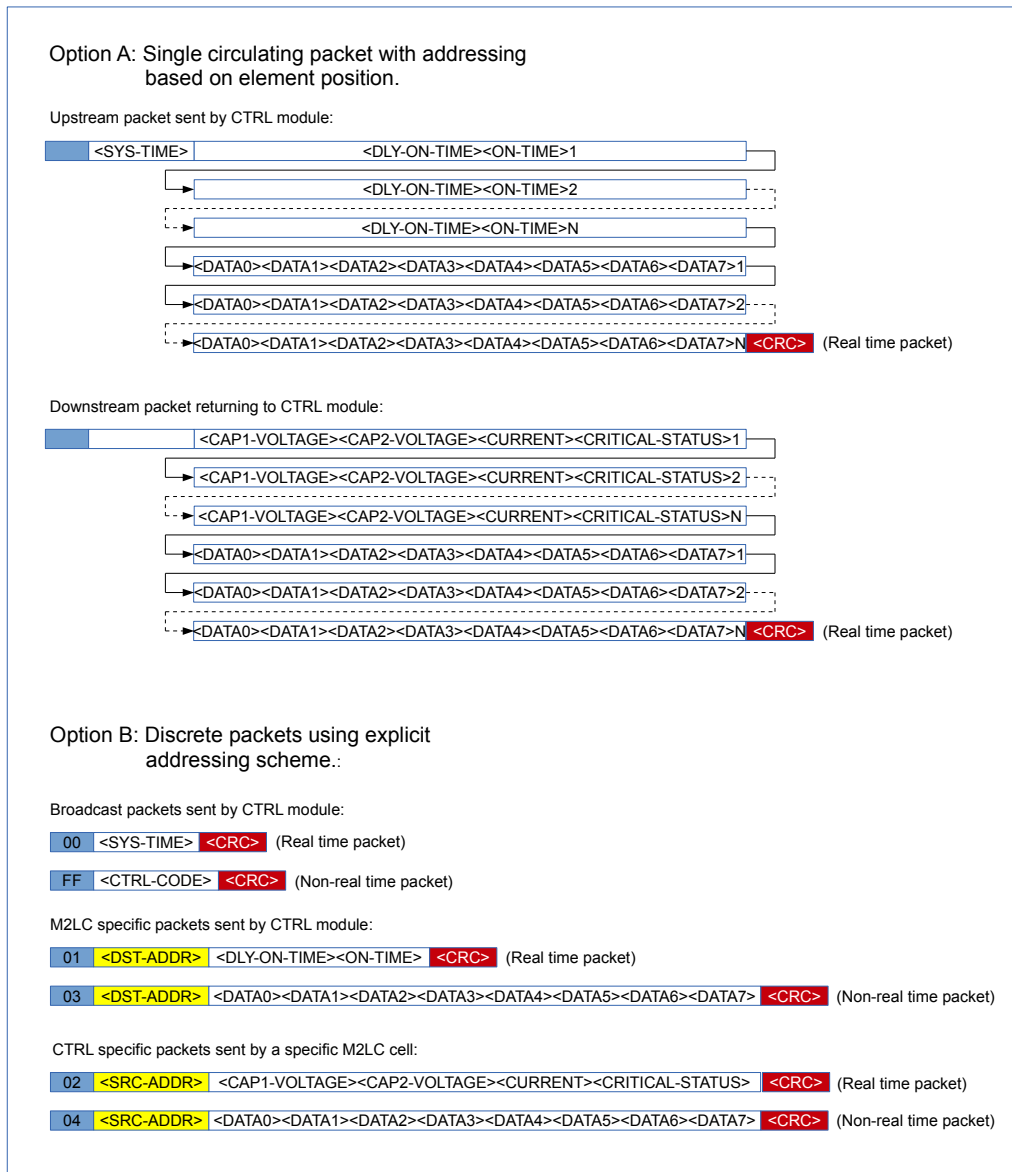


FIGURE 7 Packets exchanged between a CTRL module an M2LC cells.

Figure 7 shows the layout for both communications protocols described as *Option A* and *Option B* above.

⁸ The type of communication protocol described here is similar to EtherCAT except both upstream and downstream links are utilized for data exchange with *overlays* used to reduce the size of the packet.

The discrete packets used in Option B are sent and returned at different rates based on the importance of the information exchanged. For example the packet labeled “01” for controlling the switch of the IGBT's on each of the M2LC cell is sent by the CTRL module at a high cyclical rate (typically every 50 uSec). The packet labeled “00” sent by the CTRL module and used for correcting system time on each of the M2LC modules is also sent on a cyclical basis but at a much lower rate. The packet label “04” which is used for general purpose data transfer is sent on an asynchronous basis by a M2LC cell in response to a command sent by the CTRL module. Also packet “01” that controls IGBT switching can be sent two or more times within the update quantum. If the CRC fails on the first packet sent, the update is ignored. The second, if successful, updates the switching information.

On the other hand, the single circulating packet used in Option A relies on one large packet to transfer the same informational components contained in the discrete packet approach. To improve efficiency, *overlays* are used. For example, for a given M2LC cell, the data elements <DLY-ON-TIME><ON-TIME> of the upstream packet uses the same position and size as the returning <CAP1-VOLTAGE><CAP1-VOLTAGE><CURRENT><CRITICAL-STATUS> element of the downstream packet. Also, sequencing of elements can be done to reduce the size of the packet. For example instead of <CAP1-VOLTAGE><CAP2-VOLTAGE><CURRENT><CRITICAL-STATUS> being passed from a given M2LC cell at 50 uSec, the data can be broken up over multiple 50 uSec periods. Say <CAP1-VOLTAGE> would be passed in the first 50 uSec period, <CAP2-VOLTAGE> in the second, and so forth. <DLY-ON-TIME><ON-TIME> would always be passed every 50 uSec.

Providing redundancy to the system by adding a *loop(s)* in the communications link is simpler using discrete packets because the protocol of the packets are not dependent on number of M2LC cells in the loop. This is not true however for the single circulating packet where proper data exchange relies on the position of its data elements relative to the M2LC cells being accessed.

An important concept in both Option A and Option B above is the use of the fields <DLY-ON-TIME><ON-TIME> which is used for turning on and off the IGBT switches in the M2LC cells. In this design, a simple *ON/OFF* control scheme is not practical because of the potential for connecting together a large number of M2LC cells on a single communications link⁹. The time skew between each cell would cause inaccuracies in the switching waveforms.

Instead, *delay-on* and *time-on* information is passed to each of the M2LC cells. Typically, an update quantum of 50 uSec would be used. Using the synchronization timing of the IEEE 1588 method described above (the discussion on boundary stamps and boundary clocks), the update to all IGBT's of all M2LC cells would occur on the next 50 uSec quantum concurrently, adding virtually no error to the intended switching waveform.

⁹ In practice, this number of series-ed M2LC cells could be in the hundreds.

In some application, an M2LC system can be configured to have one or more backup cells in each of its phases that can be switched into operation in the event of a failure in one or more of the currently operating cells. A detailed discussion on how this mechanism operates is beyond the scope of this document. However, it is clear that whatever mechanism is used, there is a requirement to isolate the failed M2LC cell by *shorting* its connections within the link. Shorting of the link is accomplished with an optional bypass circuit as shown in Figure 8 (see also Figure 2).

In order to insure maximum reliability for the operation of this circuit, simplicity of design is paramount. On the other hand, the requirement for the circuit to communicate with multiple cells requires some form of complexity in the communication protocol¹⁰. Both of these factors can be addressed by using two Ethernet FX PHY's connected back-to-back and *tapping* the communications connections (in this case the MII interface with a small Programmable Logic Device (PLD)).

Even though both the RX and TX channels of the each PHY are used, the communications is uni-directional only (from CTRL module to M2LC cell). Like the SFP communications described above, this interface is also capable of operation using a redundant loop. Also like the SFP communications, this circuit connects to multiple M2LC cells in series. A two channel plastic optical fiber (POF) connects the *seriesed* M2LC cells together, and to the CTRL module.

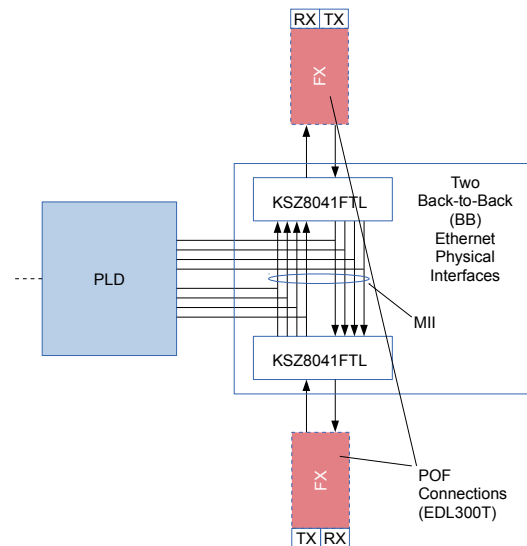


FIGURE 8 Block diagram of the optional bypass control circuit within an M2LC cell.

A cell to be bypassed is addressed using a small packet that is decoded by the PLD using the TAK (tethered address key) described above. Each packet is verified using a CRC. The PLD controls a separate set of isolated gate drivers (see Figure 2) that ultimately drive the bypass IGBT's.

As shown in Figure 2, this bypass control circuit is optional. Instead, the bypass isolated gate driver can be controlled directly by the FPGA through the SFP communications link. The encapsulation of the entire control and drive circuit in combination with the redundant loop applied to the SFP connection provides the required reliability for controlling the bypass switches.

¹⁰ For example, since multiple cells must be addressed, some sort of CRC must be used within the protocol to insure reliability.

The SFP and optional FX communications ports, FPGA, PLD, IGBT driver circuits, and feedback transducers outlined in Figure 2 require a control power supply. Typical approaches to the design of this supply can not be applied here because of the requirement for high isolation between the primary and secondary circuits. One approach taken to solve the isolation problem is to derive the primary power from the link capacitors of the M2LC cell.

There are three problems with this approach. First, the high potential of the link capacitors (1000 to 1500 VDC) adds complexity to the front end switching circuit. Secondly, start-up becomes tricky because voltage must first be applied to the link capacitors. Finally, but most importantly is the issue of failure mode operation. In the event of failure in the M2LC cell, it is not possible to control the bypass IGBT circuit (FX communications) unless this circuit is powered with a separate supply. Also, it is not possible to sustain the SFP communications link to the upstream or downstream M2LC cells even if the fault does not breach the SFP ports or the FPGA control circuit. The epoxy potting of the entire M2LC control circuit as outlined in Figure 2 above dramatically increases the likelihood that this circuit will survive a catastrophic failure caused by the a short in the IGBT's and link capacitors.

A better approach is to design a power supply that relies on a current source input. The outline in Figure 9 depicts such a circuit. Here, the AC current source can be supplied from two possible sources depending on the M2LC system is to be used.

For applications involving high AC voltage power conversion (a system bus greater than 25,000 VDC), the current source would be provided by the M2LC cell link bus connection itself (see also Figure 2). As long as this connection is outside the bypass IGBT circuit, AC current would be provided by the other functioning M2LC cells within the link even if the M2LC cell in question is disabled or shorted.

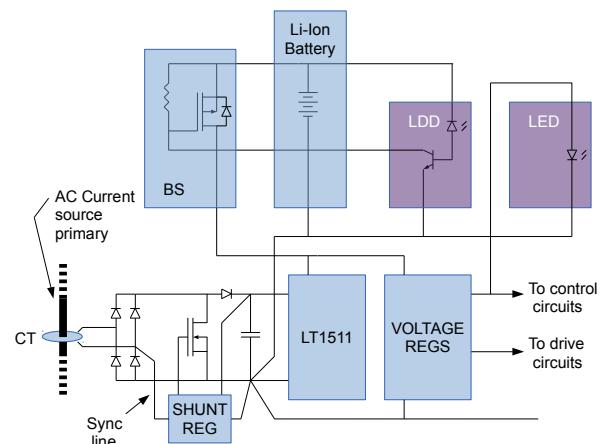


FIGURE 9 Block diagram of control power supply within an M2LC cell.

For applications that use a lower DC bus voltage and require that the M2LC system be able to sustain DC current to a load (aka, motor control applications), the AC current source primary would be provided by a single and separate AC current source¹¹ amplifier. The output of this amplifier would connect in series, the current transformers of all M2LC cells¹².

No matter which source is used, power from the current transformer (CT) into the cells power supply is regulated by a current shunt. The switching frequency of the shunt is synchronized to the current source by monitoring the leading edge of the AC voltage before it is rectified (Sync line). This is important for two reasons. First, to eliminate audible *beating* between the source frequency and the shunt regulator frequency. Secondly, in the case where a solid state current source is used as the supply, different phase shifts can be added to each shunt regulator of each M2LC cell in order to control the maximum voltage swing on the output of the solid state current source.

When the power source is derived from the M2LC cell link bus, additional circuitry must be

-
- 11 The current source can be derived from passive components (transformer and inductor) or from a PWM current source amplifier. Both these methods are currently covered under a US Patent owned by Curtiss-Wright Corporation.
- 12 An added benefit of a current source supply is it's immunity to failure of the power supply of a specific M2LC cell. Operation of the power supplies of the other cells would continue to operate.

added to the power supply to be able to start-up the circuit when power is first applied to the system. This is done by allowing a lithium-ion battery to be switched into the circuit of each M2LC cell on start-up.

Referring again to Figure 9 a single POF element is driven starting at the CTRL module (Figure 3) using a simple light emitting diode (LED). The light detecting diode¹³ (LDD) in the first M2LC cell (Figure 2) of the link can detect this light through a single POF. The LDD when energized switches in the Li-Ion battery using a P-channel MOSFET (BS). This connects the battery to the voltage regulators through a battery management circuit (ex., a Linear Technology LT1511 IC) and energizes the M2LC control circuit. When energized, the LED on M2LC cell turns on it's LED which feeds to the next M2LC upstream. When the last M2LC cell energizes, it's LED signals back to the LDD of the CTRL module indicating all cells are energized.

The M2LC system is then switched into operation by the CTRL module. AC current begins to flow in the link connections of each M2LC cell. The shunt regulator on each cell is then engaged taking over the duty of supplying power to the voltage regulators and maintaining the charge on the Li-Ion battery.

The requirement that a separate LED/LDD POF circuit be used as a means to connect the battery to the control power supplies is born out of the necessity to keep the quiescent current draw on the battery to an absolute minimum. One may be tempted to use the RX portion of the circuits associated with the optional FX and SFP transceivers as means to enable the battery. This cannot be done because of the high current draw these circuits would have on the battery during storage or long periods when the system is disabled.

However, it may be possible to *tap* the light source of the receiver portion of the FX using the LDD and thus eliminate the need for the separate POF cable bridging the LED/LDD circuits on each M2LC cell.

13 The LDD is a simple a high gain opto-coupler.

The CTRL module is designed to provide flexibility in regards to various interconnection methodologies between itself and the M2LC cells. For example the M2LC system depicted in Figure 1 provides for three pole paths each with a redundant loop. The block diagrams shown in Figure 10 depict three alternative configurations. For simplicity, each of the configurations in Figure 10 show only the SFP connections are shown connected to each cell.

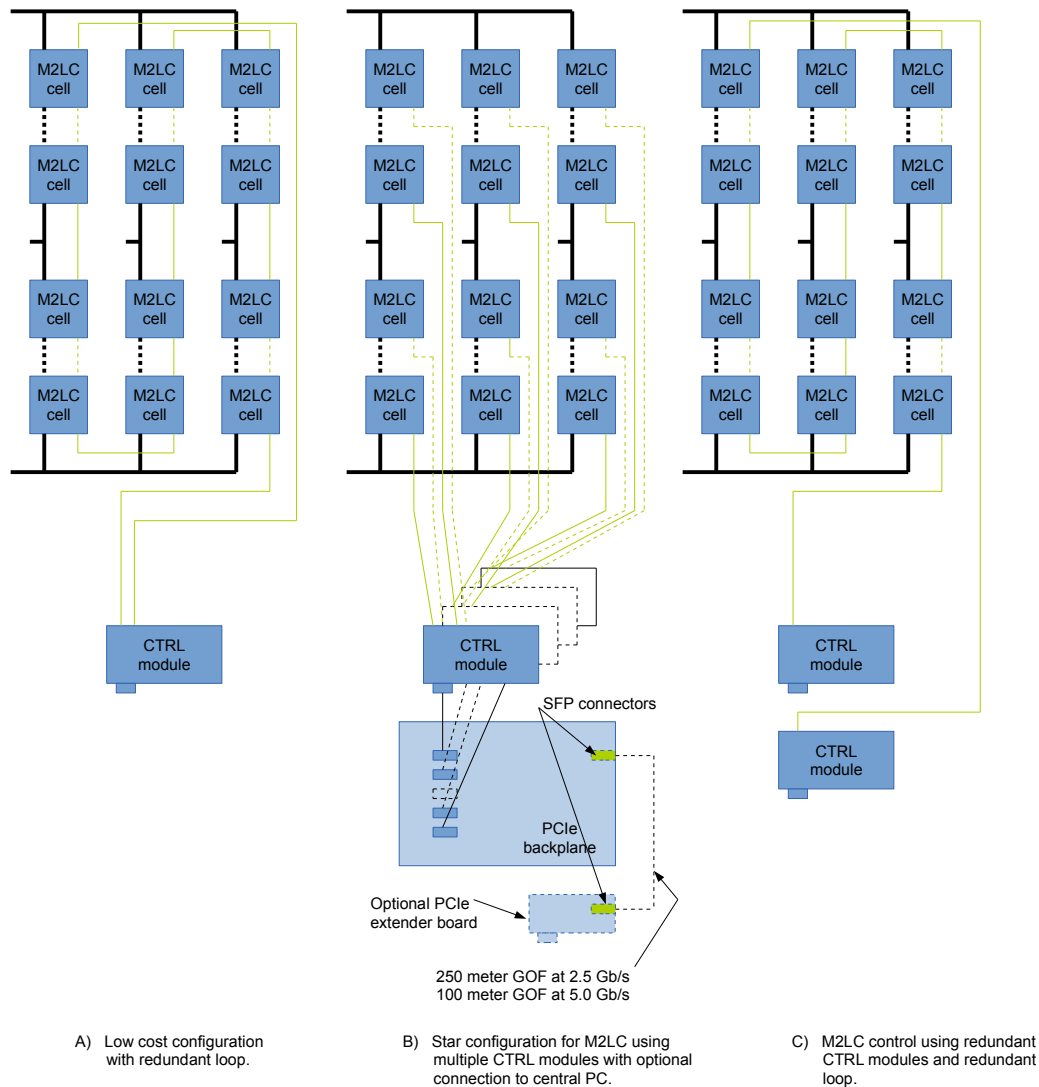


FIGURE 10 Alternative configurations for M2LC cells and one or more CTRL modules.

The diagram in Figure 10a provides for the lowest cost configuration where only one of the six SFP ports on the CTRL module is active. This single SFP connection and redundant loop is responsible for controlling all the M2LC cells within the system.

For a system that requires a high level of failure redundancy that cannot be satisfied using the simple redundant loop technique, a configuration like the one shown in Figure 10b is constructed. Here, the PCIe bus of the CTRL module comes into play (refer to Figure 3 and the accompanying text). In this example the system requires 24 SFP connections for the direct control of each of the 24 M2LC cells.

To accomplish this a PCIe back plane board is required to join together four CTRL modules. PCIe back plane boards like the Adnaco R1BP1B can be used. In the case of the R1BP1B board, an option is provided to extend the PCIe bus to a remote PC using an extender board and a 5 GHz GOF fiber cable¹⁴. For systems that require more than four CTRL modules, a custom PCIe backplane board can be designed. The control entity on the Adnaco R1BP1B is a PLX Technology PCIe switch IC. This design can be easily expanded to 16 or more PCIe connections.

Finally, in Figure 10c a configuration can be devised that incorporates two CTRL modules at each end of a redundant loop. Instead of this loop connecting back to a single CTRL module, it instead connects back to a second (backup) CTRL module. This approach can also be expanded to provide CTRL module backup for a system like the one depicted in Figure 1. Here, the return loops of each of the three phases of M2LC cells can instead return back to a second CTRL backup module.

14 The details of such a system are beyond the scope of this document.

Appendix A – Fault Redundancy and Uninterruptible Operation

The number of variables that can contribute to failure in an M2LC converter is quite large due to the complexity of the system. However, there are some design methodologies that can be built into the design to minimize the effects of certain types of failures.

Here we will concentrate on failures associated with an M2LC Cell (Figure 2) and assume for the time being that the control (CTRL Module in Figures 3 and 4) is viable at all times.

Also to be assumed are the following:

An M2LC system of the configuration outlined in Figure 10-A. (One CTRL module controlling an arrangement of M2LC cells configured as a three phase converter with a redundant loop connection.

An M2LC cell as outlined in Figure 2 with two SFP ports only. No FX ports or PLD for controlling the *bypass circuit*. Instead, the *bypass circuit* is configured to be controlled by logic in the FPGA. We assume the Li-Ion battery start-up circuit and the associated LED/LDD interface to *kick-start* the power supply fed by the current transformer (CT).

Being that the CT is used to supply power to the control power supply, the assumed application is for continuous AC operation (e.g., not a motor drive).

Each cell has the ability to measure current flowing in its bus. So the difference between current measured in any cell in the upper leg and any cell in the lower leg of a given phase can be interpreted as the output current for that phase.

The control circuit of a given cell is protected from catastrophic failure (as described for Figure 2) such that the bypass circuit if required, can be engaged at any time by logic in the FPGA.

Using the conditions above we need only to add at least one *floating M2LC cell* to the top and bottom leg of each phase of the system shown in Figure 10-A. Each floating cell assumes some portion of the operating time. For example if there are four cells plus one redundant cell per top and bottom leg of given phase (a total of six redundant cells for the three phase system), each cell of a given leg assumes 4/5 the total duty.

If the CTRL module detects a failure in one of the cells, this cell is put into bypass mode and taken out of operation. The duty for all cells in the leg that experienced the failure is now 4/4 or 100 percent. The CT in the faulted cell continues to supply power to the control power supply of that cell. Thus, the cell in question remains viable as far as a conduit for maintaining the communication link.

With this arrangement we can handle one M2LC cell failure per leg. However, it should be obvious that by adding more floating cells per leg, we can handle more failures per leg.

The CTRL module outlined in Figure 3 and 4 is responsible for providing the coordination required to switch in and out, the redundant cells described above. A portion of this control from Figure 4 is shown in Figure 11 below, which would be the minimum required interface to satisfy the system described in Figure 10-A.

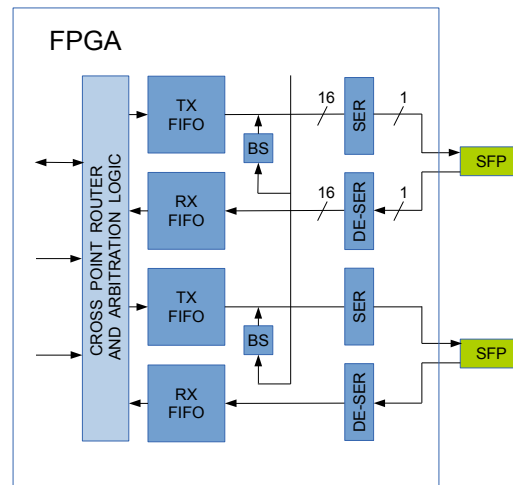


FIGURE 11 Portion of interface circuit required to control the system in Figure 10-A

As for the SFP communications provided on each of the M2LC cells, a more detailed block diagram of the interface described for Figure 6 is shown in Figure 12 below.

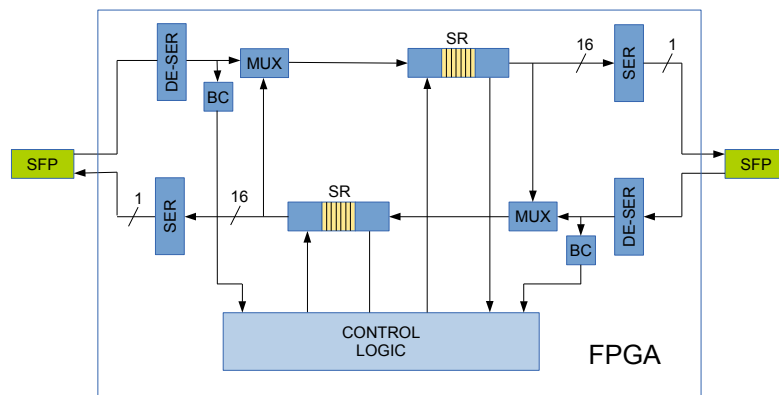


FIGURE 12 Detailed block diagram of the SFP interface shown in Figure 6.

The interfaces outlined in Figure 11 and Figure 12 above connected together form the representation of the entire control interface for the system described for Figure 10-A. This is shown in Figure 13 below.

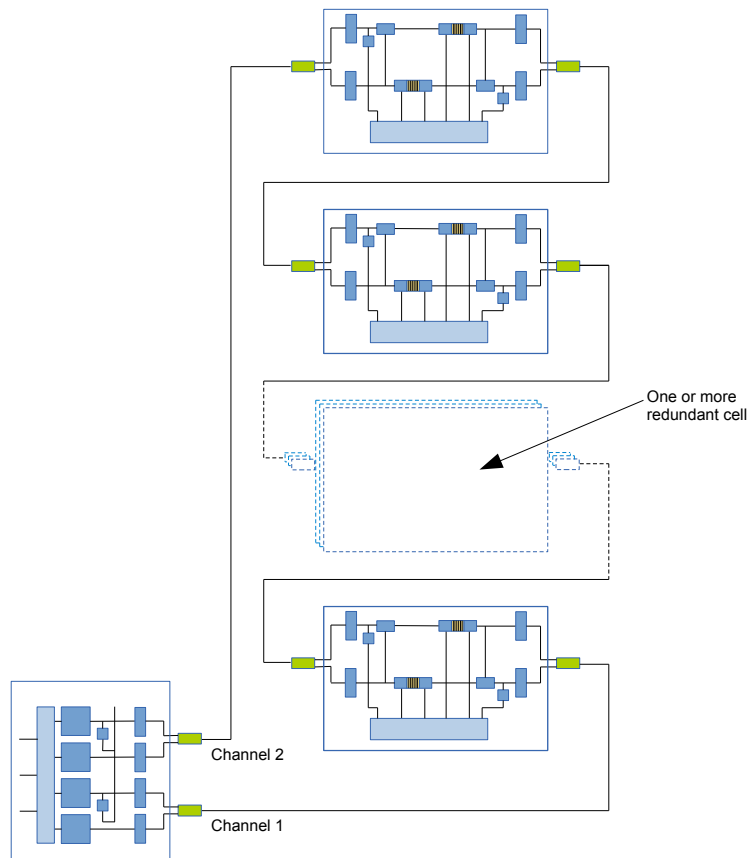


FIGURE 13 Control interface for the system outlined in Figure 10-A (Phase connections omitted).

For simplicity, the motor phase connections are not shown in Figure 13, making the system appear as one array of interconnected cells. Also, as described at the begin of this section, at least six redundant cells are distributed in the system of Figure 13.

Figure 11 and 12 provides additional detail as to the components required for the SFP interface within an M2LC cell. Serialization and de-serialization (SER and DE-SER) blocks embedded within the FPGA utilizing 8b10b decoders/encoders and clock recovery circuitry are used to transfer data between the successive interfaces.

Note in Figure 12 the components labeled SR (shift register) and MUX (2-to-1 multiplexer). Data is de-serialized, decoded and then shifted into the SR. Based on the outcome of the decoded header information, $n\text{-states}^{15}$ later one of three operations are performed.

- Data is inserted from the Control Logic block and passed to the SER.
- Data is extracted from the SER and passed to the Control Logic block.
- Data is extracted *and* inserted to and from the Control Logic block and SER.

¹⁵ “n-states” to decode the header.

This technique of transferring extracting and inserting data from a serial interface is similar to the methodology used in EtherCat™ communications.

As described previously for Figure 5 and Figure 6, the BC (boundary clock) and BS (boundary stamp) provide IEEE-1588 clock synchronization. This will be described in more detail below.

The MUX blocks in Figure 12 serve as mechanism to return data to the sourcing SFP interface on the CTRL module. In normal operation (no communication faults) the M2LC cells at the *ends opposite of the sourcing SFP interface* on the CTRL module use the MUX to steer the return data back to this sourcing SFP. In figure 12 this would be the M2LC MUX's closest to to Channel 1 and Chanel 2 connections of the CTRL module.

Thus, if all the communications links are in tact, there are two communications loops operating concurrently. It is the decision by control logic in the CTRL module which of the two communications loops are operating at any given time¹⁶. The M2LC cells respond accordingly to currently active loop.

The use of IEEE-1588 synchronization plays an important rule in this communications mechanism. As discussed previously, the control of IGBT turn-on/turn-off in each M2LC cell is accomplished by sending two pieces of information, a *delay-on-time* and *on-time* every update interval (see Figure 7, Option B). Synchronized timers on each of the M2LC cells uses these two fields to control the IGBT switching. This information sent in the current update cycle is acted upon at the beginning of the next update cycle provided there are no data errors (CRC) in the current cycle.

This approach provides two benefits.

- IGBT switching on all M2LC cells is aligned precisely.
- Under most usage scenarios, the last valid on/off information can be reused in subsequent cycles if one or more CRC errors are detected.

The second condition provides for uninterrupted operation in the presents of one or more cycles with a detected CRC error. This will be explained in more detail below.

16 A usual method for selecting the interface is by *round-robin* scheduling.

Using a 1 Gb¹⁷ link, we will assume the update rate for delivering a data packet to each M2LC cell in the chain to be 50 uSec. Deviating from the packet definition outlined in Figure 7 – Option B, we define a new protocol comprised of four packet types shown in Figure 14¹⁸.

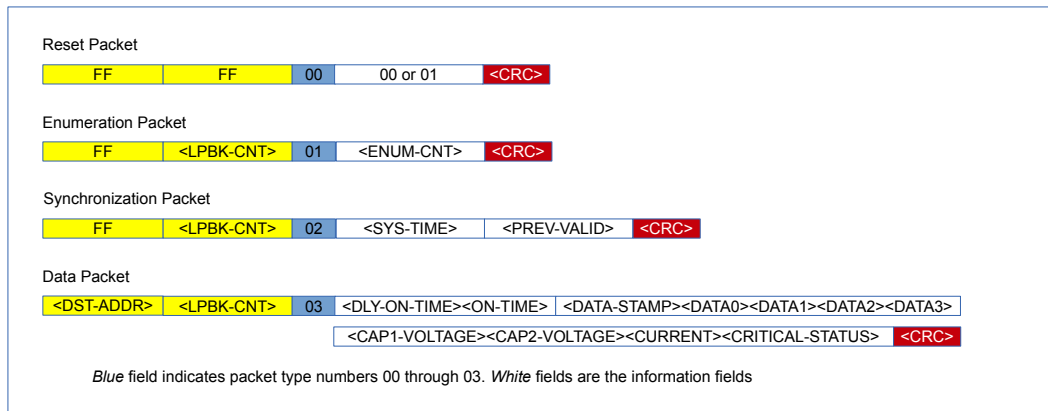


FIGURE 14 Packet types.

Some changes have been made to this protocol relative to that described in Figure 7 – Option B. This new definition is as follows (reference Figure 14).

- 1) - Each packet begins with two *steering* fields (in yellow), <DST-ADDR> the destination address and <LPBK-CNT> the loopback count field which is decremented each time the given packet passes through a M2LC cell interface. The number FF indicates broadcast for the <DST-ADDR> and *ignore loopback* count for the field <LPBK-CNT>. The Reset, Enumeration and Synchronization packets are always broadcast packets. The Reset has no context in regard to the address and loopback mechanisms and as such has a value of FF for both <DST-ADDR> and <LPBK-CNT>.
- 2) - Each packet type has an information field (in white). For the Reset and Synchronization packets, the information field is *read-only* by the M2LC cells and *write-only* by the CTRL module only. The Reset packet can be generated by an M2LC cell or the CTRL module with its information field *read-only* by the consumer. The information field of the Enumeration and the <DATA-STAMP><DATAx> fields of the Data packet is *read-write* by both the CTRL module and the M2LC cell acting on this data.
- 3) - The Reset packet is acted on by all M2LC cells and passed on to the next cell when sent from the CTRL module. This is indicated by the setting of it's information field to 00 by the CTRL module. If the Reset packet is sent by a given M2LC cell it's information field is set to 01. It is also passed through all subsequent M2LC cell interfaces but is acted on only when received by the CTRL module.
- 4) - The Enumeration, Synchronization and Data packets are sent only by the CTRL module. If the <LPBK-CNT> of any of these packet types is non-zero it is decremented¹⁹ by the receiving M2LC cell and passed on. If this field is zero when received by a subsequent cell, the MUX

¹⁷ 1.25 Gb before 8b10b decoding.

¹⁸ All packets start with a special *frame* character followed by one or more *clock correction* characters, a detailed explanation of this is beyond the context of this document.

¹⁹ Read, decreased by 1 and written back into the <LPBK-CNT> field.

associated with the channel in the containing M2LC cell is switched such that the packet is passed on in the reverse direction using the channel opposite the sending channel (see Figure 12). The packet is passed on without any further processing just as it would normally be the case if being passed on to an upstream interface.

5) - All packet types (Reset, Enumeration, Synchronization and Data), sent by the CTRL module return to the CTRL module either on the same channel or opposing channel if a MUX is one of the M2LC cell is active. Packets that do not return indicate a communications failure and are handled accordingly as described below. A Reset packet (the only packet that can be initiated by an M2LC cell) is passed through all other M2LC cell interfaces and consumed by the CTRL module (assuming the connection is viable).

6) - All packet types are appended with a <CRC> (cyclic redundancy check) field. All fields of a packet that passes through and/or are destined for a particular M2LC cell are validated against the CRC field. If the current M2LC cell detects an error, it is flagged for the current update cycle. If the packet is an Enumeration or Data packet (where the information fields are updated by the cell, the generated outgoing CRC is *poisoned*²⁰ and sent out with the outgoing packet. For all other packets that do not have their information field modified, an invalid CRC is sent out with the invalid packet unmodified.

Two conditions can exist at system start-up. All M2LC modules (including if any, the redundant modules) can be intact without any fault conditions. Or, one or more M2LC modules can be disabled due to a communication fault.

Assume first, a healthy system with no communications faults. If this is the case, system start-up proceeds as follows.

7) - First a basis must be derived as to number of M2LC cells connected in the system. This is done by referring to the *Tethered Address Key* (TAK) mechanism described for Figure 1 and Figure 2 above.

8) - One or more Reset packets are sent by the CTRL module to all M2LC cells on both channels, but not concurrently (see Figure 13). With no MUX switches enabled (see Figure 12) on any of the M2LC cells, the Reset packets are received on the complement SFP channel of the CTRL module (see Figure 11). If the packets are successful received on the opposite channel (Channel 1 TX sends a Reset and Channel 2 RX sees the propagated Reset), an assumption is made that the integrity of all M2LC are not compromised and the cells are set to receive the Enumeration packet. If the opposing channel does not detect the propagated Reset, an assumption is made that one or more M2LC interfaces are not functioning and an auxiliary step must be performed which is described below.

9) - Assuming a complete non-compromised system, an Enumeration Packet is sent next. Starting with Channel 1 of the CTRL module, this packet is sent with <ENUM-CNT> set to zero and <LPBK-CNT> set to FF. As the packet passes through each M2LC cell the value of <ENUM-CNT> is incremented and the packet is passed on to the next cell. Also, the M2LC cell records the number (before being incremented) and uses this value as the cell address. Channel

²⁰ A *poisoned* packet has its generated outgoing CRC inverted (one's complemented).

2 of the CTRL module eventually receives this Enumeration packet. The number contained in <ENUM-CNT> must match the known TAK count. If not, a mis-configured system is flagged and no further actions are performed. Assuming a valid enumeration, for each channel the MUX of the M2LC cell *closest to the channel opposite* of the sending channel is chosen to be the loopback MUX for all subsequent packet traffic (Synchronization Packets and Data Packets).

10) - For the condition above where the Reset packet was not detected on the opposing channel, an intermediate step must be performed to determine if the system is capable of operation using Channel 1 and Channel 2 of the CTRL module concurrently²¹. First, a Synchronization packet is set by both Channel 1 and Channel 2 starting with the <LPBK-CNT> of zero. If both are returned to their respective receive ports of the sending channel, the <LPBK-CNT> is increment by one and sent again on both channels. This is continued until one or both channels cease in the reception of the returned packet²². Here, only two conditions can allow the system to continue with enumeration. An M2LC cell has one broken SFP connection, or an M2LC cell is completely non-responsive on both channels. Assuming a valid state, for each channel the MUX of the M2LC cell that has been determined to be the *furthestmost loopback MUX opposite* the sending channel is chosen to be the *loopback* MUX for all subsequent packet traffic (Synchronization Packets and Data Packets) on that channel. Note that if the condition was that of a broken SFP, we maintain the capability of communications with all M2LC cells.

11) - An Enumeration packet can now be sent on each channel. However, unlike the that described for 9) above, the <LPBK-CNT> of the Enumeration packet must now be set immediately to the *furthestmost loopback MUX* determined for each channel by the Synchronization packet described in 10). Enumeration for each M2LC cell is now determined in a similar fashion to that described in 9).

At this point we have one of two condition. A completely intact communication interface as described by 9), or an impeded communication interface described by 10). For 9), the choice of which channel (1 or 2) is to be used is arbitrary. For 10) both channels must be used in order to be able to communicate with all cells. The selection and control of *redundant* cells as described at the beginning of this section is still valid for both conditions.

After enumeration, it is expected that the communication cycle begins at a predefined period (typically 50 uSec). A Synchronization packet is sent followed by a series of Data packets (reference Figure 14) of an amount equal to the number M2LC cell in the system, in an alternating fashion if condition 9) is in effect. If condition 10) is in effect, the Data packets sequence is broken up into two sets²³ and sent on each of the two channels concurrently. For both of the conditions 9) and 10), the <LPBK-CNT> is set in all packets to select which MUX of a given M2LC cell is to *loopback* the series of data packets. The Synchronization packet is sent to keep the IEEE 1588 *boundary clock* BC (the M2LC cell reference clock) synchronized to the *boundary stamp* clock BS of the CTRL module²⁴ (see Figures 11 and 12).

Next, let us assume that the system is operating with no error conditions as described by 9) above. Because of the operation of the IEEE 1588 synchronization mechanism it is possible to react to

²¹ In our example configuration of Figure 10-A, we can tolerate only one communication break in the chain.

²² A MUX or MUX's of a given M2LC cell failed to *loop-back* the Synchronization packet.

²³ The set sizes dictated by the enumeration process discussed in 10).

²⁴ The use of IEEE 1588 to establish an accurate reference for the data packet fields <DLY-ON-TIME><ON-TIME> was discussed earlier in this document.

a cell communication failure condition as described in 6) above in *realtime* without shutting down the system and re-enumerating the cell communications start-up protocol. Two types of failure conditions can be handled as described as follows.

First, in regards to *soft errors* (CRC errors) as described in 6), the field <PREV-VALID> in the Synchronization packet that precedes the series of Data packets sent to the cells is used as flag to indicate to all cells that the <DLY-ON-TIME><ON-TIME> fields sent in the *previous* update cycle can be used to adjust the on-off control commands to the IGBT's in the *current* cycle. <PREV-VALID> is set TRUE if on the return of the Data packets to the CTRL module in the *previous* cycle, there were no *poisoned* CRC's (see 6) above). If the M2LC cells detect <PREV-VALID> as FALSE on the *current* update cycle, the *last correct* values of <DLY-ON-TIME><ON-TIME> should be used for the current cycle. This mechanism, provides for *un-interruptable* operation of the M2LC system in that the update cycle time of 50 uSec is usually must faster than required for the operating frequency of the system load²⁵.

Secondly, in the case of a *real time* condition regarding the loss of communications on one of the M2LC cells as described in 10), the IEEE-1588 synchronization protocol allows the system to re-enumerate "*on the fly*". Here again two condition come into play like that discussed in 10). First, loss of communications can be a result of a broken SFP connection. In this case, the entire sequence described in 10) and 11) can be re-administered within one or two update cycles while each of the M2LC cells maintain the use of the last valid <DLY-ON-TIME><ON-TIME> fields.

The second of these two conditions (the complete loss of the operation of an M2LC cell) becomes problematic. Here, for the duration of the re-enumeration process, we lose the operation of the faulted M2LC cell until one of the redundant cells is selected to take it's place.

25 A typical load max. operating frequency is 50/60 Hz.