POWER ELECTRONICS SYSTEM COMMUNICATIONS

By
Ivana Milosavljevic

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirement for
the degree of

Master of Science
in
Electrical Engineering

Dr. Dusan Borojevic, Chairman
Dr. Scott Midkiff
Dr. Walling Cyre

January 14th, 1999
Blacksburg, Virginia

Keywords: power electronics, three-phase, converters, control networks, industrial communications, protocol, serial, port, real-time, ring, delay, synchronization, high-speed, digital, design, PLD, FPGA, VHDL, RS-232

POWER ELECTRONICS SYSTEM COMMUNICATIONS

Ivana Milosavljevic

Abstract

This work investigates communication issues in high-frequency power converters. A novel control communication network (Power Electronics System Network or PES Net) is proposed for modular, medium and high-power, converters. The network protocol, hardware and software are designed and implemented. The PES Net runs at 125 Mb/s over plastic optical fiber allowing converter switching frequencies in excess of 100 kHz. Communication control is implemented in a field programmable gate array device. A novel synchronization method applicable to ring networks is proposed. The effect of the communication delay on the power converter operation is studied.

To my family

Acknowledgments

I would like to thank my advisor, Dr. Dusan Borojevic, for his patience and support, and for giving me the opportunity to learn at Virginia Power Electronics Center (VPEC), Virginia Tech (VT). I appreciate and admire his broad views and courage to investigate the interdisciplinary issues, such as the work described in this thesis.

I would like to thank Dr. Walling R. Cyre and Dr. Scott F. Midkiff for serving as my committee members. Dr. Midkiff's teaching and valuable comments contributed to the quality of this thesis. I am grateful to Dr. Cyre for entrusting me with his design laboratory package. This saved me a lot of time and made my design tasks easier.

I would like to recognize and give my gratitude to my co-workers at Fiber-Electro Optic Research Center, VT, especially Mr. Carvel Holton. I also thank center directors, Dr. Fred C. Lee (VPEC) and Dr. Richard O. Claus (FEORC), for taking an interest and being a part of this project.

I would also like to thank all of my PEBB team members, especially Mr. Kun Xing, Mr. Ivan Celanovic and Mr. Sam Ye, who have contributed their time and work to this project. I also thank my friends Mr. Nikola Celanovic, Mr. V. Himamshu Prasad, Mr. Dong Ho Lee, Mr. Carlos Cuadros and Mr. Sriram Chandrasekaran, from whom I learned many useful things.

Finally, I thank all of my friends and family in the USA and Yugoslavia for their encouragement and trust. The VPEC students and staff have made my life in Blacksburg better.

This work was supported by Office of Naval Research, the equipment was in part supplied by the Automatic Design Research Group, VT, and it made use of ERC Shared Facilities supported by the National Science Foundation under award Number EEC-9731677.

Contents

Chapter 1 INTRODUCTION

Power electronics converters are used today in many applications ranging in power from a few watts in power supplies for portable telecommunications and computer equipment, to hundreds of kilowatts in complex industrial motion control systems, and even tens of megawatts in power utility installations. Although specific implementations of the switching power converters vary as much as their applications, every converter consists of three different functional units, as shown in Figure 1.1.

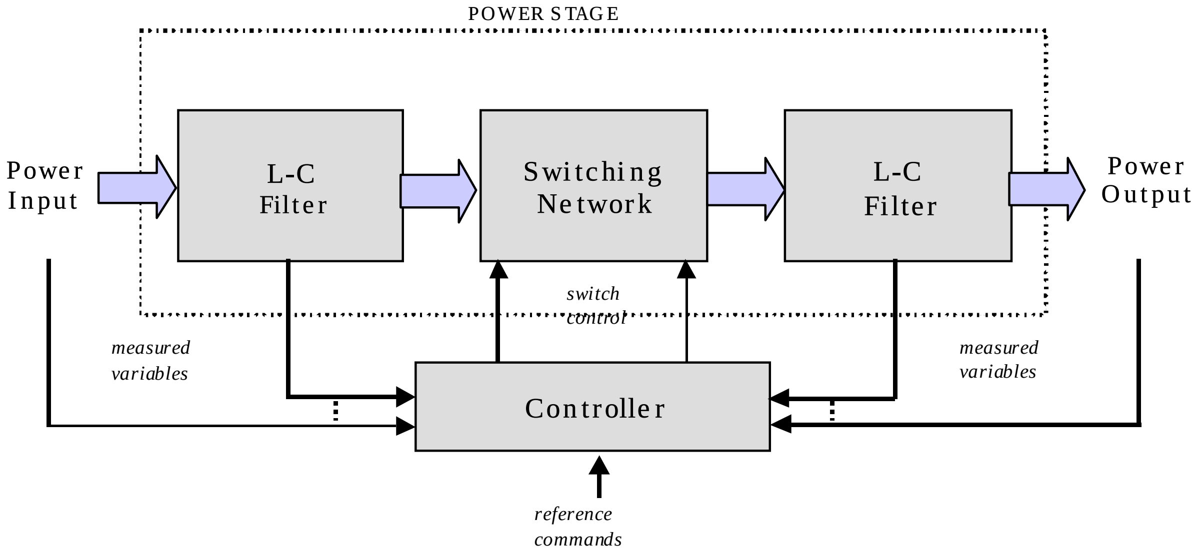


Figure 1.1 Principal functional blocks of a switching power converter

The core power processing is performed by the switching network composed of electronic switches such as MOSFETs, IGBTs, GTOs, and diodes. By using high switching frequencies, the desired low-frequency waveforms of output voltages and currents can be obtained from many different waveforms of the input voltages and currents. The high frequency components are filtered out by the L-C networks, which together with the switching network comprise the converter power stage. Operation of the power stage is controlled by a separate, signal-processing block, on the basis of the reference commands and measured system variables. Different power processing functions (such as DC-DC conversion, AC-DC rectification, DC-AC inversion, etc) are achieved by varying the circuit topology of the power stage and by different control algorithms [1], [2].

The switch control signals are “ON – OFF” commands which are pulse-width-modulated (PWM) in most of the new applications, although frequency modulation or combined frequency and PWM are also used. In most applications, the measured variables are continuous analog quantities, while the reference commands could be analog or digital. Although analog controller implementations are still widely used in industry, most new designs are fully digital, especially in medium– and high-power applications (a 10 kW and more) [3, 4, 5].

Very often, individual power converters are a part of more complex power processing systems. In those cases, the system control structure is usually hierarchical, as shown in Figure 1.2. Typical examples of such configurations are motion control systems (e.g.: in machine tools, paper and steel mills, robotics…) where the system controller generates coordinated reference commands for the power converters (called “motor-drives”), whose controllers are then responsible for achieving the commanded mechanical motion in only one coordinate. The high-level controllers in such power electronics systems are always digital and the data transmission between them and the individual controllers is also digital in most cases.

Because the high-level controllers and data transmission had been digital for over twenty years, standardized communication protocols have been developed for the links between the power converters and the system level control [17-23]. However, up to now, the connection between the controller and power stage within each converter, was not even considered to be a communication link; it was viewed more as a set of connection lines. In the low power converters this may still be true, since both the control and the switches are implemented close to each other, usually on the same board. The power stage is then directly linked to the controller. In medium- and high-power converters, where the controller is set at some distance from power switches this is not true. When the point-to-point link wires were substituted with optical fiber to improve the noise immunity, it became obvious that this is in fact a communication system.

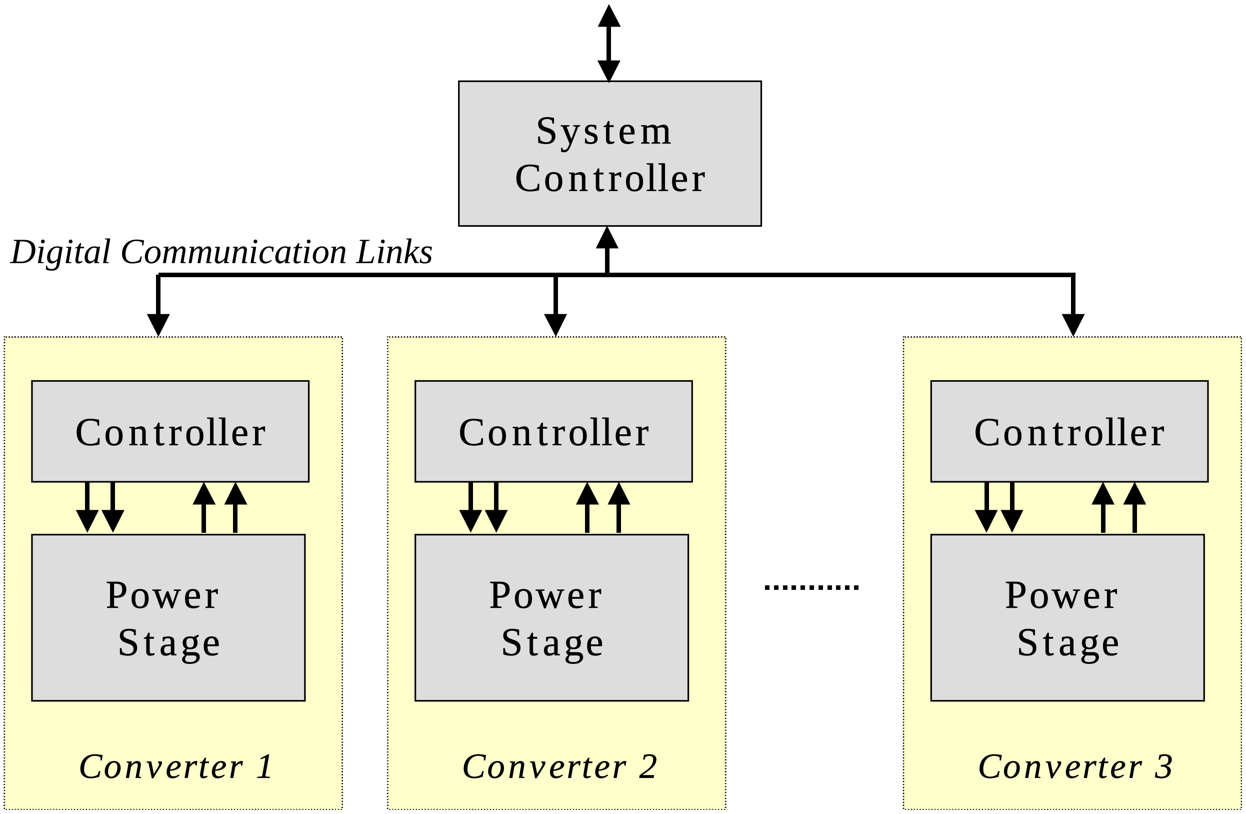


Figure 1.2 Principal control structure of a power electronics system

Today's digitally controlled power converters tend to imitate the structure of the analog ones. Although control processing is now in a digital form, control communication topology is not utilizing that. Information such as currents and voltages are sensed on the power stage and brought in analog form to the controller board. Control signals are sent in point-to-point line structure and sensed variables are received in the same manner. Structure such as this, just mimics the communication structure that is used in analog converters. In a digital system there is no need for this anymore; a structure that utilizes the digital nature of the control can be used.

The problems with the conventional communication structure will be illustrated on the example of a digitally controlled three-phase rectifier/inverter shown in Figure 1.3. Each switch receives an on/off command from the controller and sends back the fault information through an interface block called “gate-driver” (GD). Analog variables that could be sensed are phase currents i_a, i_b, i_c, phase voltages v_a, v_b, v_c, dc-link voltage v_{dc}, as well as temperatures of all switches. Not all of these have to be measured in every application because some of them may be redundant and some may be known apriori. However, it is safe to assume that in many applications the number of sensed variables is approximately equal to the number of active switches.

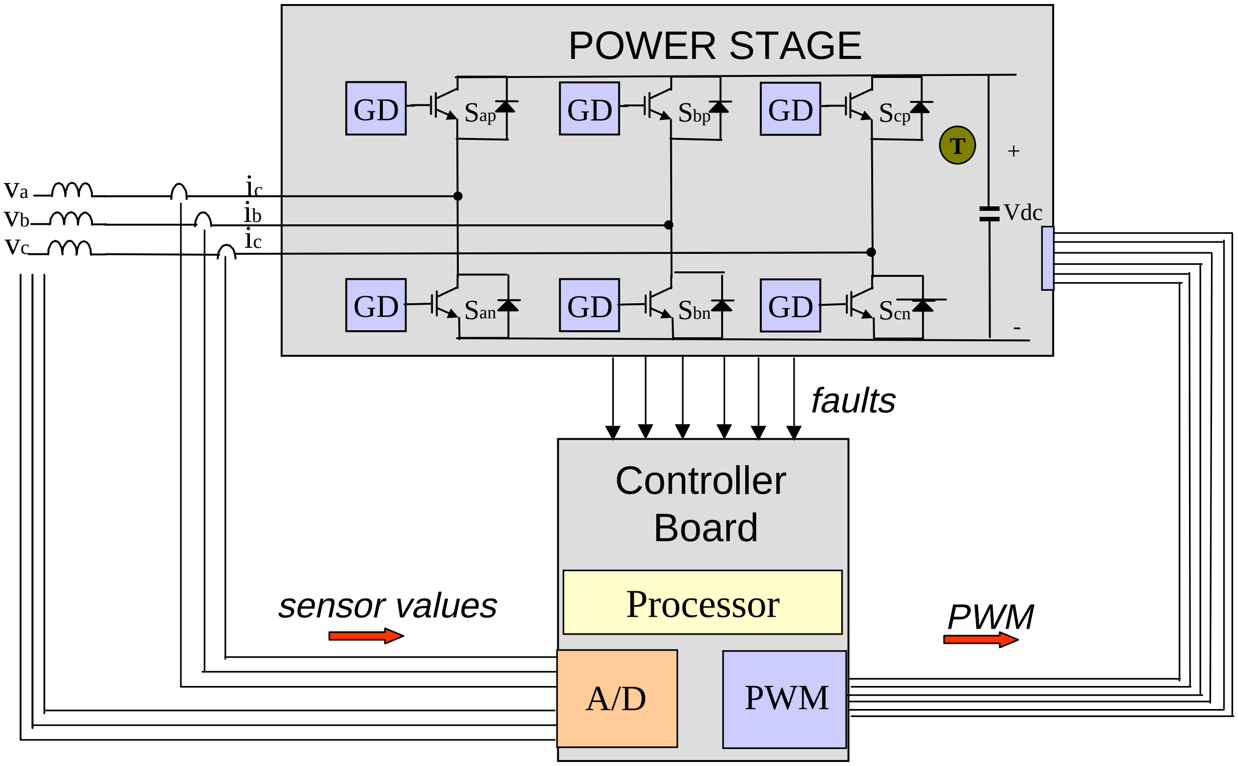


Figure 1.3 Conventional topology of a digitally controlled three-phase rectifier/inverter

In a soft-switching converters, additional auxiliary switches are used to relax the switching stress of the main switches. Each auxiliary switch also requires two commands (on/off and fault), but usually no additional variables are sensed. Therefore, the number of different variables that need to be communicated in a converter can be estimated as

$$N_{var} \approx 3 * N_{sw} + 2 * N_{aux}, \qquad (1.1)$$

where N_{sw} and N_{aux} are the numbers of active main and auxiliary switches, respectively. For the converter in Figure 1.3, (1.1) estimates 3* 6 = 18 variables, while the actual number of variables in some specific implementations of different three-phase converters is given in Table 1.1.

Table 1.1. Number of variables transmitted between controller and power stage in three-phase converters

TOPOLOGY	NUMBER OF SWITCHES		NUMBER OF LINES					TOTAL
			PWM	Fault	Sensing			
	Nsw	Naux			Iac	Vac	Vdc	
3-Phase rectifier	6	0	6	6	2	2	1	17
3-Phase rectifier with soft switching	6	6	12	12	2	2	1	29
3-Phase VSI[1] with neutral leg	8	0	8	8	3	3	1	23
3-Level NPC[2] VSI with soft switching	12	12	24	24	3	3	2	56

In the conventional converter communication architectures, one transmission line is used for every variable. The length of the communication links depends on the physical size of the converter, which in turn depends on its power level. For a 100 kW rectifier from Figure 1.3 the distance between the controller and the power stage is approximately $L_{link} \approx 3$ ft. Then, the total length of the communication links is

$$L_{comm} = N_{var} * L_{link} \approx 50 \text{ ft} , \qquad (1.2)$$

---

[1] VSI: Voltage Source Inverter

[2] NPC: Neutral-Point Clamped

In order to increase the noise immunity and safety, the optical fibers are often being used for the communication links, especially in higher-power applications. For every optical fiber link, an optical receiver and transmitter pair has to be used. Therefore, 17 optical receivers and 17 transmitters are required for the three-phase rectifier, in addition to the 50 ft of the optical cable. This amounts to a considerable quantity of hardware.

The capacity of such a communication system is hugely underutilized. The required communication capacity can be easily estimated by assuming that all the variables have to be transmitted with the same accuracy and the same sample rate equal to the converter switching frequency, f_{sw}. The capacity, without any data compression, is then

$$C_{req} = N_{var} * n_b * f_{sw}, \tag{1.3}$$

where n_b is the minimum number of bits required to code the variable with the highest required accuracy. For example, if a 16 bit accuracy (more than enough in most applications) and 20 kHz switching frequency are used in the three-phase rectifier/inverter, the required total capacity is $C_{req} \approx 5.5$ Mb/s. On the other hand, if the transmission medium is twisted-pair wire or plastic optical fiber (POF) with only 5 Mb/s capacity, the system with 17 parallel links has the total capacity of 85 Mb/s, yielding the utilization of under 7%. If standard commercial POF with 150 Mb/s is used, the communication channel utilization would be about 0.2%; obviously a gross waste of resources.

Finally, the power converter with a number of wires hanging around (Figure 1.4), is not only susceptible to noise and is costly, but can also lead to confusion, be a safety issue and an obstacle to converter modularization.

By using a more modularized converter topology, module boundaries can be defined so that the modules have functional and communicational independence. New topologies that better utilize and benefit from digital control concept, and cannot be applied to converters with analog control, are now emerging [6, 7]. A type of the modular control concept has been implemented in a converter, which is using a variable hysteresis band current loop [6].

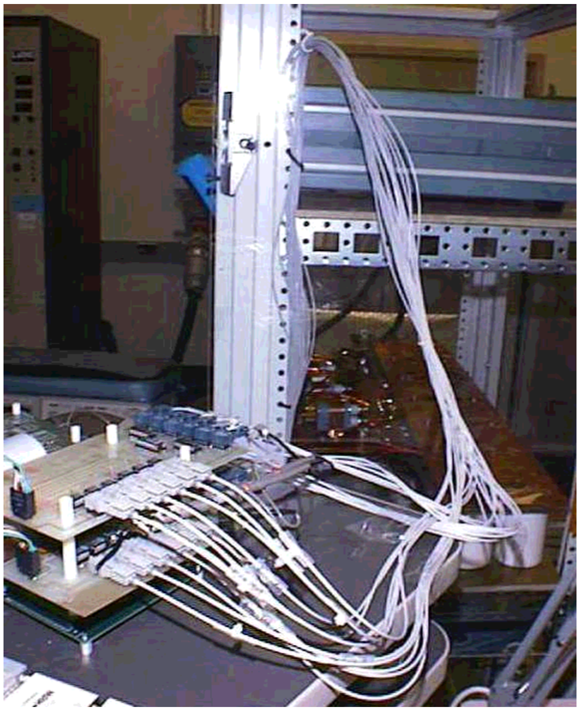


Figure 1.4 Fiber-optic interface in a digitally controlled three-phase converter

The objective of this thesis research was to devise a new approach to the communications between the controller and the power stage in medium- and high-power switching converters, by exploiting the digital controller implementation and modularized converter topologies. The approach should be able to provide significantly better utilization of hardware resources (i.e. potentially lower cost and complexity), without sacrificing and possibly improving electrical performance and reliability of the converters. The concept must be suitable for use in different converter implementations for a wide variety of applications, and should take into account expected future developments in power electronics technology.

The main result of this work is the development of a daisy-chained, ring communications architecture for data transmission inside power converters. The new communications protocol suitable for these applications, named “Power Electronics System Network – PES Net”, is proposed. A novel solution for the critical issue of the power stage switch synchronization has been developed. The architecture and the protocol have been implemented in hardware and software. Operation of the proposed communication system has been evaluated and verified.

The following chapter describes the concept of modularization of the power converter topologies and analyzes the communication system requirements in detail. Existing communications standards, which are mostly used for high-level communications in power electronics systems, are surveyed in Chapter 3. The issue of switching synchronization is addressed in Chapter 4, and possible solutions are evaluated. The PES Net architecture and protocol are defined in Chapter 5, while its implementation and measurement results are presented in Chapter 6. Evaluation of the proposed system and suggestions for future developments are given in the last chapter. Appendixes contain technical documentation for the implemented experimental system.

Chapter 2 COMMUNICATION REQUIREMENTS IN MODULAR POWER CONVERTER SYSTEMS

2.1 MODULAR POWER CONVERTER SYSTEMS

Recently, the concept of modular power converter systems became more popular. This concept requires intelligent power modules, which can replace complex power electronics circuits with a single device [8]. The efforts are made to develop universal, intelligent, scaleable power control devices, which will become standardized units, easy to manufacture, replace and use. A high volume production of such devices would contribute to its decreasing cost and, consequently, the price reduction of power electronics systems [9] enabling their application in many new areas. This principle also minimizes the layout and packaging parasitic, because all of the power semiconductor device control circuits and interconnections will be integrated together as a large intelligent power device [10].

U.S. Navy has considerable efforts in developing such systems through the Power Electronics Building Blocks (PEBB) program. This work is a part of that undertaking.

2.2 POWER CONVERTER SYSTEM

A conventional digitally controlled three-phase converter is shown in Figure 1.3. The controller calculates duty cycles based on control algorithms and the measured values of currents and voltages on the power stage. The controller contains all of the 'intelligence' in this system. The processor, analog-to-digital (A/D) converters and field programmable gate array (FPGA) are all located in that same place [11]. All of the data acquisition, processing and gate drive signal formation is performed on this board. Power stage contains just the components related to power-processing.

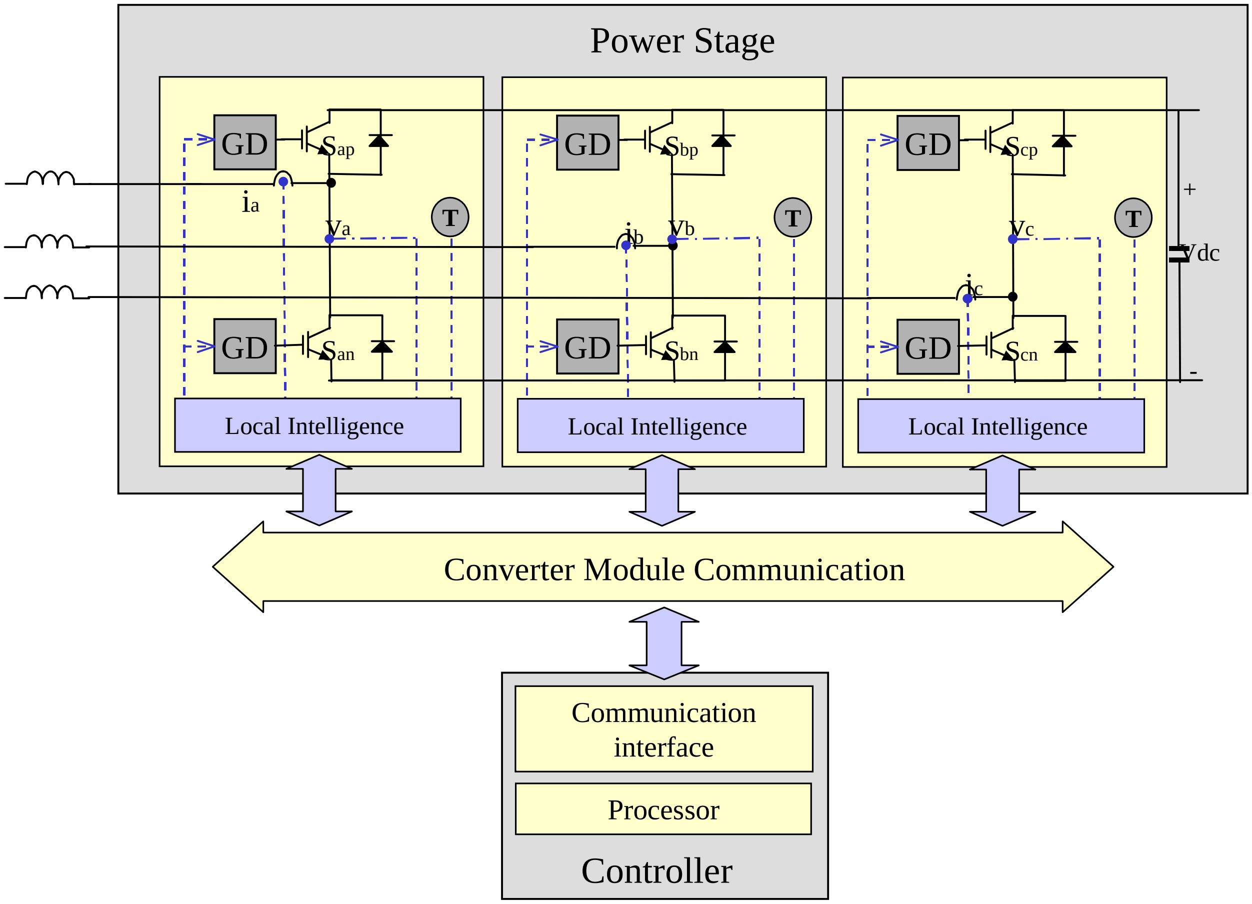


Figure 2.1 Modularization of a power converter

In this work a modularized approach is used. Instead of having just one smart module, all of the modules have some local intelligence (Figure 2.1). Such 'smarter' modules can execute the low-level control functions: local A/D control, gate-drive signal generation, fault control and communication functions (Figure 2.2). The controller keeps the supervisory and converter control functions.

Control algorithm of a power converter is based on the values of two currents, two AC voltages and a DC voltage. In Figure 2.1 analog variables i_a, i_b, V_a, V_b and V_{dc} are measured and converted into digital form, within a power module. Based on them, in a closed loop operation controller calculates the PWM variables, two-level analog signals, for all of the switches: Sap, San, Sbp, Sbn, Scp and Scn. The PWM signals are sent to switch gate drives, which then handle switch operation. Their turn-on and turn-off times are assigned so that the output voltage would follow the desired line-to-line AC voltage waveforms.

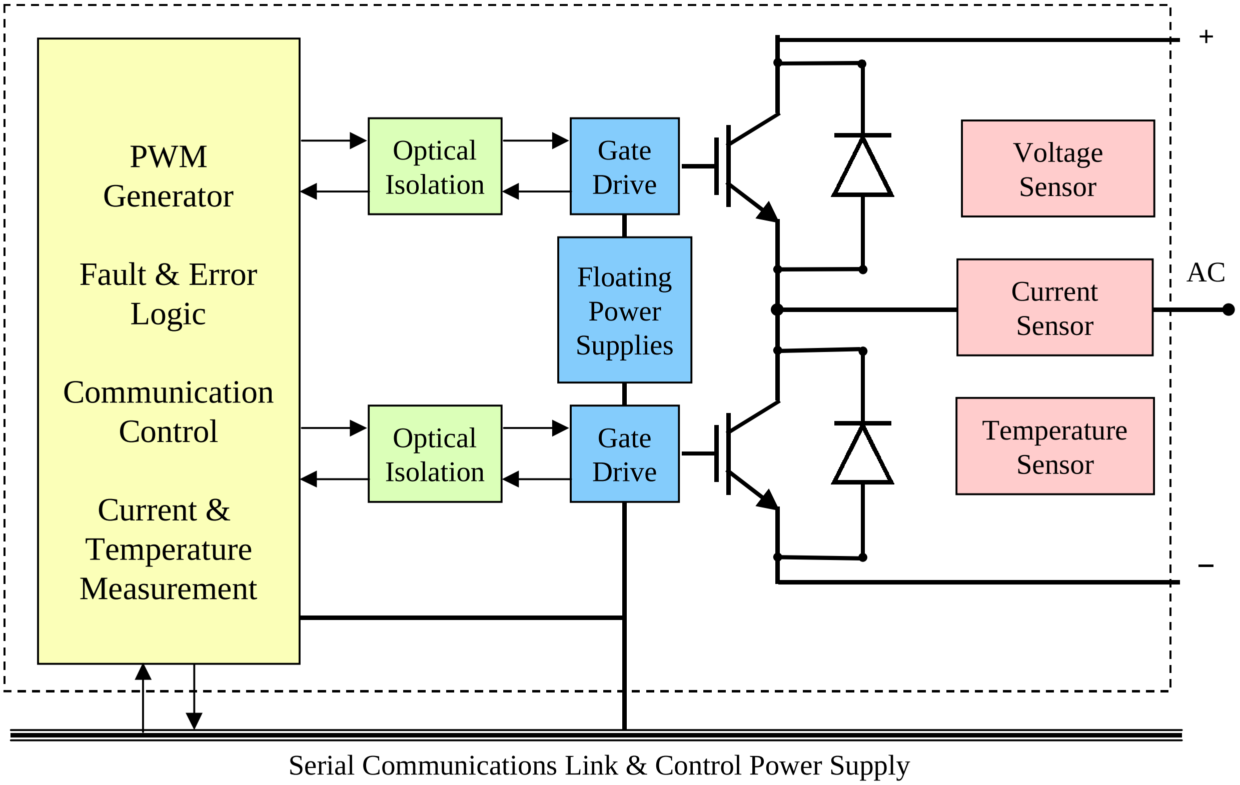


Figure 2.2 PEBB module

Not all of the switching combinations are allowed during the converter operation. Often switches of one power module operate in complementary manner (Figure 2.3). To prevent the shorting of the DC link voltage a ‘dead-time’ has to be implemented. This is the time during which both of the switches have to be turned off, after either of them has been conducting. The dead time between two transitions may be variable and should be programmable.

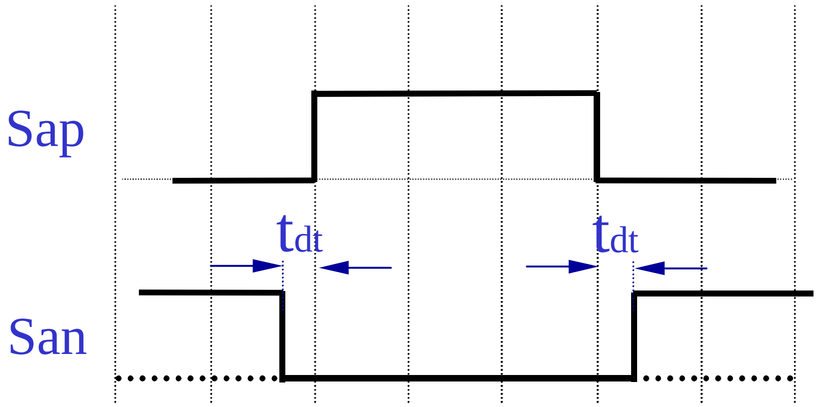


Figure 2.3 The PWM sequences of two complementary switches: Sap and San; t_{dt} is a switch 'dead-time''

The power modules in Figure 2.1 and 2.2 are, what we call, PEBB modules. In each of them, current, voltage (AC or DC) and temperature can be measured. A fault signal is generated in a PEBB module in case of an over-current. For each of the switches a PWM signal is needed, but when the switches operate in complementary manner, only one PWM signal needs to be sent. The 'dead-time' can be pre-programmed into the PEBB module, along with the converter switching frequency and a sampling frequency (the last two are usually the same). The communication between PEBB modules and a controller is the communication system that this work is most focused on.

In this part, the implementation and design of a power electronics communication system is discussed in terms of a three-phase converter. However, it is also applicable to more complex topologies: when the system has more than three PEBBs (such as a three-phase converter with a neutral leg [12]) or when there are more than two switches within a module (soft-switching case).

2.3 CONTROL COMMUNICATION

The communication in the power electronics system should be hierarchical in nature. Control of the power electronics system can be defined on three hierarchical levels (Figure 2.4). The highest hierarchical level is an Executive Level. The middle one is Zonal level, and the lowest is the converter level [13],[14].

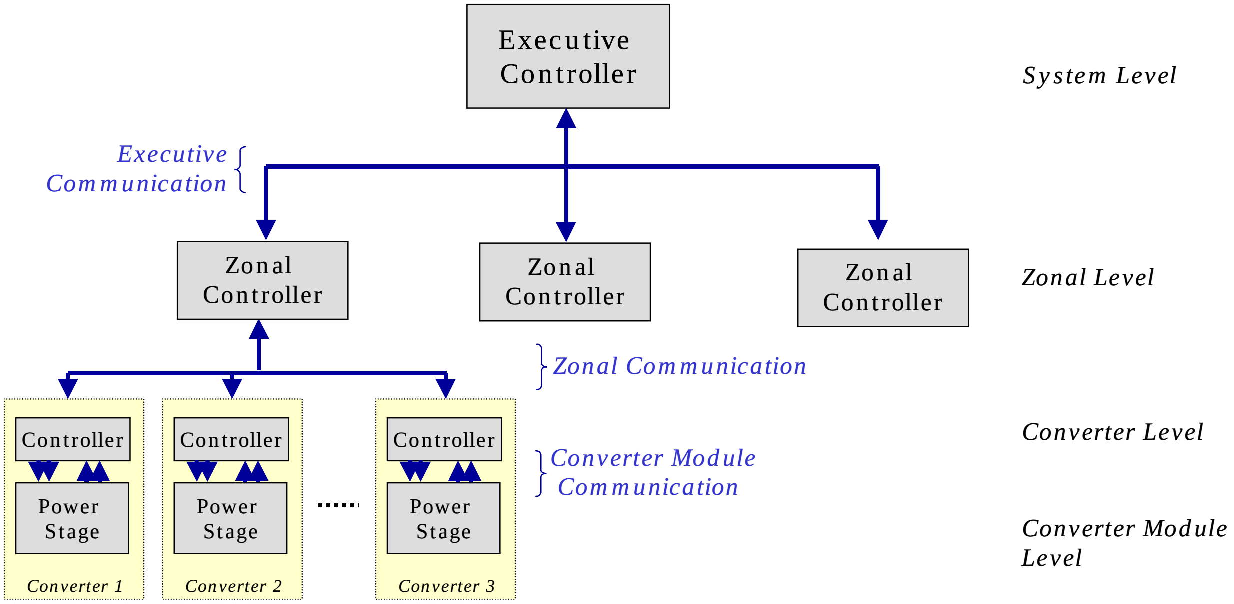


Figure 2.4 Structure of a communication system

An option is to have just one communication network for all levels. That way all zonal controllers, converter modules and executive controller would communicate over the same medium. In this case we would have fewer cables and probably a cheaper network. Also, it would be easier to reconfigure the network (zonal controller could reassign working converter moudle in the place of the one that just broke down). A disadvantage would be that heavier traffic would cause the higher delay over such networks and possible network congestion problems. Probably, a better option is to have separate networks. This would enable us to pick one or two protocols and topologies that could better suit the particular needs of each hierarchy level. It would also allow less complex network control and interface hardware.

Communication in Higher Levels

The purpose of an executable level communication is to monitor the work of the zonal controllers. Zonal controllers monitor operation of the systems located in close proximity. Some of those systems are the power converters.

Examples of information passed on executive and zonal level are: shutdowns, power-ups, downloads of the power converter controller's code, changing the proportional gain, etc. A higher level of communication does not have very strict timing requirements, because the human or a computer issues the commands. Data messages could be between 1 bit (command "turn on") and 2KB long (usual size of a converter controller's executable code).

The system should be reliable, flexible, re-configurable and possibly zonal. A possible network configuration at this hierarchical level could be from 3 to 10 higher level controllers in charge of 50 to 500 converter controllers. An optical fiber network would also be beneficial for this application, since the lines would be in the same environment.

Converter Module Communication Requirements

The communication system, within a power converter, carries messages with different information about PEBB modules (values of current, voltage, temperature, and duty-cycle ratios). Messages at this level are short. They are sampled at the converter's switching frequency. Data that is exchanged at this level is isochronous. This is similar in nature to audio data delivery requirements: all of the packets are the same size and are sent at a constant rate [17]. The difference is that audio data is not affected by processing delay, as long as it is fairly constant or smaller than a certain value. This is not true in the case of converter communication.

At converter module communication level, there is a much harsher time delay constraint since one switching cycle of a converter lasts only few tens of microseconds. Communication concerns on this level are high speed, implementation delay and synchronization of operation of PEBB modules. Standardization is a key issue here, since today this is mostly custom designed and topology dependant.

The requirements of this kind of system are discussed based on necessary communication transmission medium, bandwidth, number of nodes, synchronous operation capability and desirable features.

Bus Arbitration

The controller is processing measured data once in every switching cycle. Based on calculation results, it generates control signals. Both measurement and calculation information has to be delivered in every switching cycle. The system is controlled from one point, making it a master-slave structure (controller acts as a master).

The largest amount of data (except status information) is isochronous. Consequently, an appropriate network for this system does not need a sophisticated bus arbitration scheme, because all of the nodes, except the master node, are the same, and messages are occurring regularly.

Transmission Medium

In power electronics systems, there are numerous transitions between high and low energy levels. That creates a very noisy environment. Since copper cables pick up electromagnetic radiation and act as an antenna, non-conductive optical fibers are a much better choice.

In this system, plastic optical fiber (POF), Hard Clad Silica® (HCS) and glass fiber can be used. Plastic fiber is the cheapest, lightest and easiest to work with, but glass optical fiber has lower attenuation. Some of the relevant fiber characteristics are given in Table 2.1.

The medium of choice for this system is plastic optical fiber because of its low cost and because it is very easy to work with. No special training is required for its termination. Plastic fiber can be cut with a wire cutter and then polished with a grid paper. POFs are not as fragile as glass fiber. Their bend ratio is much higher.

Table 2.1. Characteristics of different types of optical fibers

PROPERTY	POF[15] (AT 650 nm)	HCS®[15] (AT 650 nm)	MULTIMODE GLASS FIBER[16] (AT 1300 nm)
Typical Cable Attenuation	22 dB/m	7 dB/m	0.0011 dB/m
Diameter (core and cladding)	1 mm	0.43 mm	3 mm
Mass	5.3 g/m	6.1 g/m	9 g/m
Short term bend radius	25 mm	9 mm	45 mm
Long term bend radius	35 mm	15 mm	30 mm
Cost	lowest	medium	Highest
Cable Termination	Easy	Easy	Requires Training

The Link Length

The length of physical medium in this system is not a critical requirement. Phase legs are very close to each other (tens of centimeters), while the controller is usually in the same room (several meters). The POF easily can support a 100 m long link. That is more than is needed for this kind of a system.

Number of Network Nodes

The number of nodes in the system is small compared to some other systems where control networks are used (automotive, appliance control, etc.). In a three-phase system, there is a main node (controller) and three more nodes – one for each of the phase legs. It is possible to have one controller supervising more than one converter, to have more than three-phase legs in the system or to have a load as a network node. It is reasonable to assume that it will be less than 32 nodes in this type of network.

Bandwidth

In a power converter system, only short messages are exchanged. Resolution of PWM signals sent from the controller to the PEBB modules is usually between 8 and 12 bits. The number of different PWM variables that need to be sent to each PEBB module vary from 2 (in the case of single switch or two complementary switches) to 8 or more (in the case of complex, soft-switching assemblies). The resolution of used A/D converters determines the length of digital representation of converter's currents and voltages. It is usually set from 8 to 12 bits. Control algorithms usually require the measurement of one current and one voltage to be received from every PEBB module in each switching cycle. Often, other variables (e.g. module temperature) also need to be measured and transmitter to the controller.

The necessary channel capacity can then be calculated as:

$$C = N_{var/node} * n_n * n_b * f_{sw} * (1 + k_{oh}) , \qquad (2.1)$$

where $N_{var/node}$ is the number of variables per node and n_n is the number of nodes in the system (not counting the controller). The k_{oh} is defined as a percentage of data overhead, which needs to be transmitted along with the data. The minimum channel capacity, for the simplest case, would be: 4 variables per node, 3 nodes, 8-bit resolution and switching frequency of 10 kHz. Assuming 50% data overhead and based on (2.1) the minimum channel capacity is about 1.5 Mb/s. For the more complex case, where there are 10 variables per node, 4 nodes, 12 bit resolution is used and switching frequency is 50 kHz, with the same overhead, a capacity of 36 Mb/s is required.

From the equation (2.1) it obvious that a channel capacity is directly proportional to switching frequency of a converter. For a two times higher switching frequency, we would need twice as much bandwidth.

Furthermore, within a switching cycle, besides the time necessary for transmission and reception of data (equivalent to channel capacity), the time has to be allotted for the data propagation delay and for the converter controller to process the received data and recalculate new duty cycles. This can be represented in a following equation:

$$f_{swmax} < (T_{DSP} + T_{TX} + T_{RX} + T_{p})^{-1} , \qquad (2.2)$$

where f_{swmax} is the maximum converter's switching frequency; T_{DSP} is the time that a converter controller needs to recalculate converter parameters based on newly received data; T_{TX} is the time needed for data transmission; T_{RX} is the time needed for data reception and T_{p} is time delay due to data propagation.

Based on this discussion, it can be safely assumed that the needed bandwidth should be at least 1.5 Mb/s, but than this communication system would be applicable in very few cases. Desired bandwidth is more that 100 Mb/s. A higher speed would just be to a benefit, as it would allow more flexibility in a system design. It would also provide means to use this structure in more complex systems.

Synchronization

For this type of system it is not only important to deliver data in every switching cycle, but also to do it in such way that it would not disrupt a normal converter operation. Since the network nodes are, in fact, phase legs of a converter system, it is necessary to ensure their synchronous operation. If this were not accomplished, the converter's operation would not follow the intended control algorithm.

Beneficial features

Modularity is desirable for this kind of system. Due to the fact that many in the power electronics field are working on 'building blocks', it would be good if communication is based on the same principle. Modularized nodes with an identical interface would make the network easier to redesign in case a node needs to be added or simply replaced. It is especially good if all hardware interfaces remain the same for different types of systems.

Engineers building a power converter should not have to use time to work on a communication interface, which is topology dependent. A standardized interface does not need to be changed each time the converter topology or control changes. The more a communication system is independent of these, the better.

The desired characteristics of a control network within PEBB-based power converters are summarized in Table 2.2.

Table 2.2 Desired control network characteristics

Aspect	*Optimal*
Transmission Medium	Plastic fiber
Arbitration	Isochronous data, Master-Slave type
Link Length	Max. 30 m
Network Nodes	Min. 4, Max. 32
Bandwidth	> 100 Mb/s
Synchronization	Necessary

Chapter 3 COMMUNICATION BACKGROUND

3.1 COMMUNICATION TOPOLOGY

When designing a power electronics system, one of the choices that a designer needs to make is which communication topology should be used. Two of the appropriate ones are described below.

Star Structure

A star structure [17] (Figure 3.1.a) is also called a Star Bus. It is a geometric description of the network topology in which each of remote terminals or nodes is connected directly to a central node. The bus lines run from the center of the star structure to the peripheral nodes.

In conventional converter topologies shown in Figure 1.3, this is the structure that is used. All lines are point-to-point and are used for sending one signal, in one direction, over one line. As the controller generates the PWM signals, it sends them over the transmission lines to each of the controlled switches. The sensed values, whether they are in analog or digital form, are also sent one per line, but in other direction.

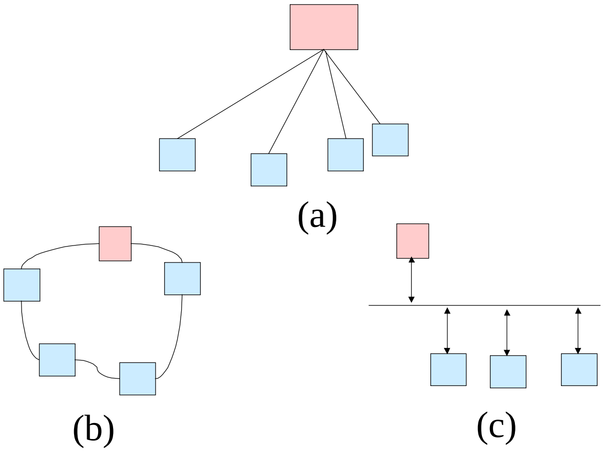


Figure 3.1 Communication topologies: (a) Star structure (b) Ring structure (c) Multitapped bus

It is tempting to use this topology for a new redesigned communication system. There are few possible ways to utilize this topology. One is to keep the point-to-point links between the controller and modules while multiplexing all the data that needs to be delivered. The number of lines would, in that case, equal the number of modules. In the case that communication in a line is uni-directional, the number of lines would be equal to two times the number of power modules.

A drawback is that this approach makes the controller interface dependant on converter topology. Since there is a link between each of the modules, there must be a port (optical transmitter or receiver) that serves as an interface to each of the modules. In the case that the number of network nodes needs to be increased, a new controller interface would have to be made. This is undesirable because the modularity is lost.

Another way to implement a star topology is to use the ‘T’ connectors of a ‘star’. For electrical signals, it is easy to implement either of these. Unfortunately, in the case of optical fibers, this technology is very expansive and developed primarily for glass fibers, while the preferred medium for power electronics system is plastic fiber. Both the fiber splice (equivalent of ‘T’) and ‘star’ are expensive products, fragile or big in size if mechanically shielded.

Again, there is a problem with modularity. If the ‘star’ coupler is used, the number of output ports has to be specified. If all of the ports are not used, a proportional fraction of optical power is wasted. On the other hand, if the system is tightly dimensioned, another ‘star’ would have to be used if, the number of nodes increases.

Similarly, if a splice is used, there are problems with dimensioning the system in terms of optical power. When a splice is designed it is pre-specified in what ratio it divides optical power among fused fibers. In the network environment they would be daisy-chained. In the case that the optical power is divided evenly, the optical power that reaches the last module would be much smaller than the original signal. This may cause problems. If care is taken and splices with different characteristics are used, the system would not lose optical power easily, but the system itself becomes more complex in a hardware sense.

An effort was made to produce an inexpensive, less that 4 cm^2 in area, ‘T’ connector for plastic optical fiber [18]. The problem of uneven optical power distribution also stands in this case.

Ring Structure

The ring structure [17] shown in Figure 3.1.b is a network topology in which terminals are connected in a point-to-point serial fashion forming an unbroken circle. A ring structure is, because of its simplicity and low cost, a very attractive topology.

The benefit of this topology is that only the optical fiber and the transmitter-receiver pairs need to be used. No other passive or active optical component is necessary. This, consequently, lowers the cost of the system. In the case that system needs to be expanded, the interface to the other nodes remains the same. Still, only one fiber goes into the module, and one goes out. This is also true for the controller module. The modularity is constituted.

The problems of ring networks are that they are susceptible to failure. When just one communication link or node ceases to operate, the whole ring fails. The other issue is inherent to ring structure: a propagation delay is created as data propagates through the ring.

In a ring structure that would be implemented with ‘T’ connectors, propagation delay would not be a problem since it would be reduced dramatically. Most of the delay in the previous structure is created by reception and re-transmission of information through the node. However, the same discussion about optical power distribution applies here. The difference is that, in this case, a double path would be provided (optical signals would propagate in both directions) to the controller board.

3.2 DATA MULTIPLEXING

Data over optical fiber can be multiplexed in three ways:

- Frequency multiplex
- Time multiplex
- Wavelength multiplex.

In frequency multiplex, channels are transmitted using different frequencies. Information has to be shifted from its original frequency range to the carrier frequency. In time division multiplex (TDM), channels are sent each in a predetermined time slot in a frame.

When wavelength multiplex is used, each channel is sent using a different wavelength (color of the light carrier). The problem is that a different type of transmitter/receivers has to be used (they operate on different wavelengths), and it is difficult to implement if more than two channels should be multiplexed. At the reception point, filters have to be used before receiving the signal.

3.3 ISO/OSI

Data network layering is a principle in which network operations are grouped into hierarchical functional blocks. The module within hierarchy performs a part of the overall functions. Data messages can be repackaged or analyzed at each of the layers.

One of the best known reference models is the model of Open System Interconnection (OSI) developed as an international standard for data networks by the International Standards Organization (ISO). This model defines seven network layers:

- Physical interface
- Data link control
- Network
- Transport
- Session
- Presentation, and
- Application layer.

The physical layer (PL) provides a link for transmitting a sequence of bits between any pair of nodes joined by the physical communication channel. The physical interface module is on each side of the communication channel whose function is to map the incoming bits from the next higher layer (Data link control).

The data link control layer (DLC) data is encoded for transmission over the physical layer. It provides some error detection and puts data into data frames that will be recognized at the reception point.

The network layer checks the data header for the purpose of routing and flow control. Packet address field is checked at this layer.

At application layer a use of specific fields, within a data packet, is defined.

3.4 CONTROL NETWORKS

In standard voice and data communication networks over which multimedia, audio or other types of data are transmitted, data frames can be several kilobytes long. In contrast, in the discussed modularized, PEBB-type system, only short messages are exchanged. Networks that are designed for this type of communication are control networks. While the major concern of data networks is throughput, in control networks, it is reliability, responsiveness and predictability [19]. The development of these networks started around 1980 when smart sensors began to be developed and to be used in digital control [20]. Today, there are many standards on the market; they were developed for different purposes (within cars, buildings, for motion control, etc) and by different companies [21].

Typically, control networks operate on low-cost, twisted pair cables. Today, for many of these networks, optical fiber is considered a new medium. There are also some types of control networks that developed from opposite directions - they started with optical fiber, and then added to their standard the use of copper cable, such as Fibre Channel.

A summation of characteristics of some of the popular control networks is given in Table 3.1. Most of the networks have an arbitration scheme, which would be excessive in a power converter communication system, while a speed requirement is reached only by a few. Standards that are intended for motion control are closer in nature to power converter systems, as those are also synchronization critical applications. The popular SERCOS [22], [23] standard does not have enough throughput. The MACRO protocol [24] comes closest to the needs of power converter communication system, so the solution proposed here is based on modification of MACRO with an improved synchronization scheme. The most relevant control networks are briefly presented below.

Table 3.1 Control Network Characteristics

Network	Speed	Max # of nodes	Arbitration	Cable type	Primary application
BITBUS	375Kb/s	32 or 250	Master/ Slave	Twisted pair	Intelligent I/O modules, Process Control
World FIP	31.2Kb/s, 1Mb/s, 2.5Mb/s	64 or 256	Bus Arbiter	Twisted pair, copper wire, optical fiber	Real-time Control, Process/ machine
Profibus	9.6,.., 1500kb/s	32 or 127	Hybrid medium access	Twisted pair, preparing for fiber	Inter-PLC communication, Factory automation
CAN	1 Mb/s	30	CSMA/CD enhanced	Twisted pair	Sensor/ actuators, automotive
Lon Works	Up to 1.25 Mb/s	32000	Predictive CSMA Collision Avoidance	Twisted pair, coaxial cable, fiber optic cable,..	Appliance control
SERCOS	2-4 Mb/s	256	Ring management	Plastic Optical Fiber	Motion Control
MACRO	100 Mb/s	256	Ring management	Glass Optical fiber, twisted pair	Motion control

LonWorks Networks

LonWork [25] technology is used today for different applications. It is intended to provide solutions to problems of designing, building, installing and maintaining control networks. Intelligent nodes communicate with each other using a common protocol. The LonTalk protocol implements all seven layers of the OSI model. Its main characteristics are summarized in Table 3.2.

Table 3.2 The LonWork network characteristics

Characteristics	***LonWork***
Transmission Medium	twisted pair, power line, RF, coaxial cable, infrared, fiber optic
Arbitration	Predictive CSMA Collision Avoidance
Link Length	Max. 30 m
Speed	up to 1.25Mbps
Network Nodes	Max. 32000
Synchronization	No
Features	supports large number of nodes, networks hierarchy, incorporates a full set of network management
Applications	appliance control

The LonWork protocol is worth considering in executable level communication. It supports a large number of nodes on a network and is hierarchical in nature. The main benefit is that it is defined for fiber optic interface and is gaining a larger user base.

The speed of the network is not very high (1.25 Mb/s), but at executive and zonal level that is not a critical factor. Executable level commands are not time-critical in nature. Furthermore, if an operator at a computer is issuing commands, the limiting factor is not the network bandwidth, but a human reaction time.

The advantage of choosing this network is that LonTalk is the open protocol. The company that has designed LonTalk protocol, Echelon, has allowed companies to port LonTalk to the processor of their choice. Besides the expected features, such as media access, acknowledgments, and peer-to-peer communication, it also includes more advanced services (sender authentication, priority transmission, mixed data rates, foreign frame transmission, etc.).

Because of its advanced services, it would be easy to implement more than just PEBB communication at this communication level. It is possible to send other data that is not related to power converter systems, but intended to the same executable level nodes, through this network. The limiting factor is bandwidth. If other applications are identified and network speed becomes an issue, it would be advisable to consider other protocols, such as an Internet-like computer data network.

Controller Area Network (CAN)

CAN [21] is one of the more popular Fieldbus [20] protocols. CAN is a high integrity serial data communication bus for real-time applications. It was developed originally for use in cars, but today it is used in other industrial automation and control applications. It uses carrier sensing multiple access/carrier detection scheme (CSMA/CD) [17] with the enhanced capability of non-destructive arbitration to provide collision resolution. The priority of a message is determined by the value of its identifier. Data messages do not contain addresses; instead, the identifier labels the content of the message. The characteristics of the CAN network are shown in Table 3.3.

Table 3.3 The CAN network characteristics

Characteristics	***CAN***
Transmission Medium	twisted pair
Arbitration	CSMA/CD enhanced
Link Length	40m-1Mb/s, 1km-20 kb/s
Speed	up to 1 Mb/s
Network Nodes	Max. 30
Synchronization	No
Features	Low cost, efficient for short messages
Applications	In sensors/actuators, automotive

The implementation problem of this scheme is the low network bandwidth. It is simply too slow to carry the necessary data for a power converter system. The non-destructive bus arbitration is very interesting, but unnecessary in this application. This network is more appropriate for the status information oriented systems, where data is transmitted at some random time. The requirements of a power converter system put more emphasis on the systems data, which are intended for retrieval of real-time, isochronous data. The ABB Power Systems, a company involved in a power electronics filed, has included the CAN communication logic on their controller board for communication to the higher level control [26].

SERCOS

SERCOS (**Se**rial **R**eal-time **Co**mmunications **S**ystem) is designed primarily as a multi-axis motion/machine control standard that provides open controller-to-intelligent digital drive interface specifications. It was accepted as an international standard, IEC 1491, in 1995. Its main characteristics are given in Table 3.4.

As a multi-axis motion standard it is capable of synchronizing communication nodes. It can support up to 254 digital drives connected to the higher level controller. The network operates at 4 Mb/s speed. Communication cycle time is set during initialization. Predetermined cycle times are: 0.062ms, 0.125 ms, 0.25 ms, 0.5ms, 1ms or any multiple of 1ms. It utilizes plastic optical fibers.

It has a ring master-slave structure. The master initiates a communication cycle by sending a Master Synchronization Telegram (MST). Each slave node uses the MST to resynchronize its clock, so that it can calculate the exact time it could reply. The one-byte data that is a part of a MST indicates if the ring is in initialization or communications phase.

Table 3.4 Characteristics of the SERCOS network

Characteristics	***SERCOS***
Transmission Medium	Plastic fiber
Arbitration	Master – Slave
Link Length	Max. 30 m
Speed	4 Mb/s speed
Network Nodes	Max. 254
Synchronization	Yes
Features	Designed for similar application
Applications	Motion control

The fastest cycle occurrence for SERCOS is 1/0.62 ms = 1.6 kHz. However, data needs to be delivered to a converter in every switching cycle, which would mean that a converter utilizing a SERCOS interface could operate at a maximum switching frequency of 1.6 kHz, which is too low for today's applications.

A good characteristic is its physical layer, since it is desirable to use plastic optical fiber because of its EMI immunity and lower cost. Synchronization is necessary for a power converter system, as well as for a motion control system, because phase legs need to operate synchronously.

MACRO

MACRO (**M**otion **A**nd **C**ontrol **R**ing **O**ptical) is a non-proprietary digital interface developed by Delta Tau Systems for the connection of multi-axis motion controllers, amplifiers and I/O on a glass optical fiber or twisted pair ring [24]. It is a master-slave, ring type, motion control standard. It operates at a considerably higher speed than SERCOS. It utilizes a physical layer of the Fiber Distribution Data Interface (FDDI), (ANSI X3T9.5) [17] network. It uses 4B/5B Non-Return to Zero, Invert on Ones (NRZI) [28] data encoding and transmits data at a speed of 125 MHz. Such high-speed capabilities are achieved by implementing most of the communication protocol in hardware.

MACRO protocol allows multiple master nodes, as well as a slave node becoming a master node in case of a ring break. Nodes on a ring can be passive or active. Only the active ones accept data, while the passive ones just pass it on.

A master node initiates communication and sends data down the line. The node to which the data frame is addressed, accepts the frame and substitutes received packages with its own data. That data goes back to the master node. If the data frame is not addressed to the node, those frames are just passed on. Communication is unidirectional.

The data frame starts with a "command byte". An address byte follows. An address consists of 8 bits, where the higher four represent a master number and the lower four a slave address. The third through eleventh data fields are data bytes. Their format is specified in a MACRO application layer. The twelfth byte is a checksum byte.

There is only one master synchronization station on the ring. It is the one that initiates the ring. Stations on the ring are synchronized on a pre-specified node-number data packet. The purpose of the synchronization packet is to ensure tight coordination between axes and to keep the clocks of different stations from ‘walking’ from each other and producing a ‘beat’ frequency.

Data packets are divided into four fields: there are three two-byte fields and one three-byte field. Their use is defined by the MACRO standard. It does not define these fields for use with a power stage.

Table 3.5 Characteristics of the MACRO network

Characteristics	***MACRO***
Transmission Medium	Glass fiber, twisted pair
Arbitration	Master – Slave
Link Length	Max. 100s of meters
Speed	125 Mb/s
Network Nodes	Max. 254
Synchronization	Yes
Features	Small user base, designed for similar applications
Applications	Motion control

Major features of the MACRO network are shown in Table 3.5. This standard has all of the bandwidth needed for power converter applications. It even allows their operation at higher frequencies, which are used seldomly. The optical fiber is defined as a physical layer, but the recent development in plastic fiber transmitters and receivers allows us to use them instead, at the same speed as specified by this protocol.

By comparing the Table 2.2, which summarized the requirements needed for the power converter communications network, with the characteristics of existing standards in Tables 3.2 – 3.5, the decision was made to use the MACRO standard as a starting point for the design of Power Electronics System Network (PES Net).

Chapter 4 SYNCHRONIZATION OF CONVERTER MODULES

4.1 PROPAGATION DELAY PROBLEM

When a ring communication topology is applied to a power converter system, a problem of inherent propagation delay is introduced. The commands arrive to the nodes with a delay that accumulates from node to node. This would cause the nodes located farther away from the controller to have a larger latency than the ones that are closer. Both SERCOS and MACRO protocols are based on a ring structure. When a ring-type communication structure is applied to a power converter system its phase-legs become daisy-chained in the communication sense.

The delay created by a ring structure becomes an important issue for power electronics system and motion control systems. These kinds of systems need to operate in synchronous way. In the case of a motion system it is obvious that coordination of movement is important. In power converter systems, the asynchronous behavior of phase legs can lead to larger ripple of currents and voltages. Actually, in such a case, the converter does not operate according to the original control algorithm, and in some cases a catastrophic failure may result.

For example, in the ring structure, shown in Figure 4.1, communication is uni-directional. Each node has two fibers connected to it. One is for receiving information, and the other is for transmission.

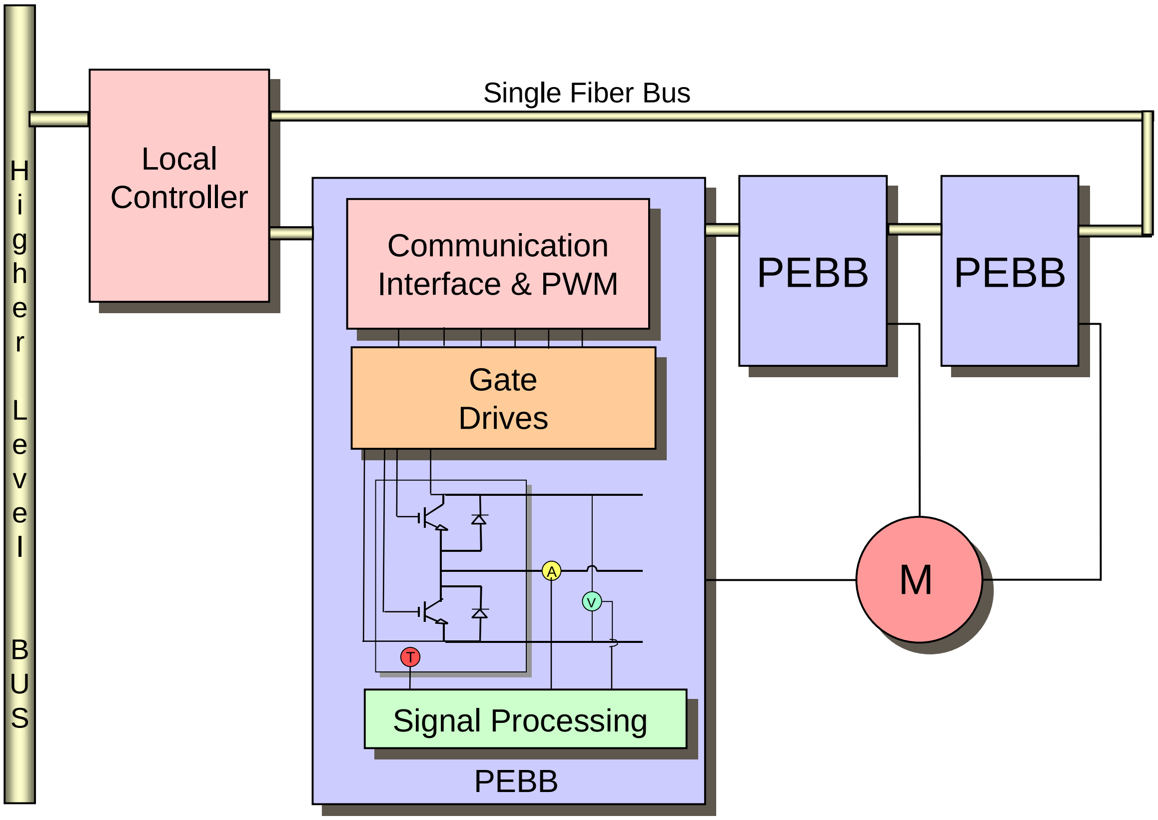


Figure 4.1 Power converter system with daisy chained phase – legs

When the controller sends a synchronization command, we can assume that it has reached the first phase leg in a daisy-chain almost immediately (Figure 4.2), because the delay through optical fiber is minimal. The first node passes that synchronization signal to the next node, but now that signal is sent with a delay, which is equal to the propagation delay through the first node. When the second node transmits the signal, it has an accumulated delay of propagation through two nodes, and so on.

The effect that a ring-type or daisy-chained control/communication structure has on the behavior of power converters is investigated in [27]. Prior to this, there was no study in the literature on how to approach this problem and what is a “short enough,” or acceptable, delay. This is a big problem for a person designing such a system.

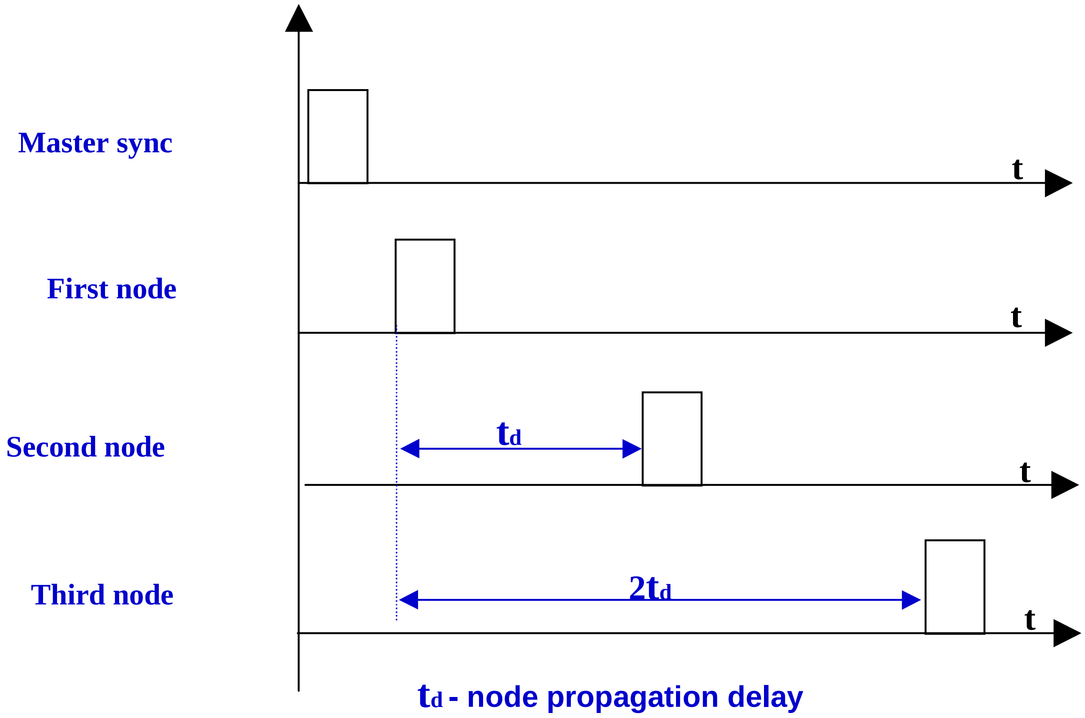


Figure 4.2 Data propagation through ring type structure

The possible ways to mitigate the problem are proposed in [27]. Using simulation in Saber, the critical delay at which the operation of the power converter still is acceptable is determined. Some design guidelines for this type of system also are given. The results form [27] are briefly summarized below.

Simulation of Converter Operation with Propagation Delay

The three-phase boost rectifier topology was used to investigate the problem of propagation delay in daisy-chained systems. Figure 4.3 shows a discrete switching model of the digitally controlled boost rectifier simulated using SABER. The phase legs are modeled as ideal switches with anti-parallel diodes. Space vector modulation (SVM) is used in generating the switch duty cycles [2].

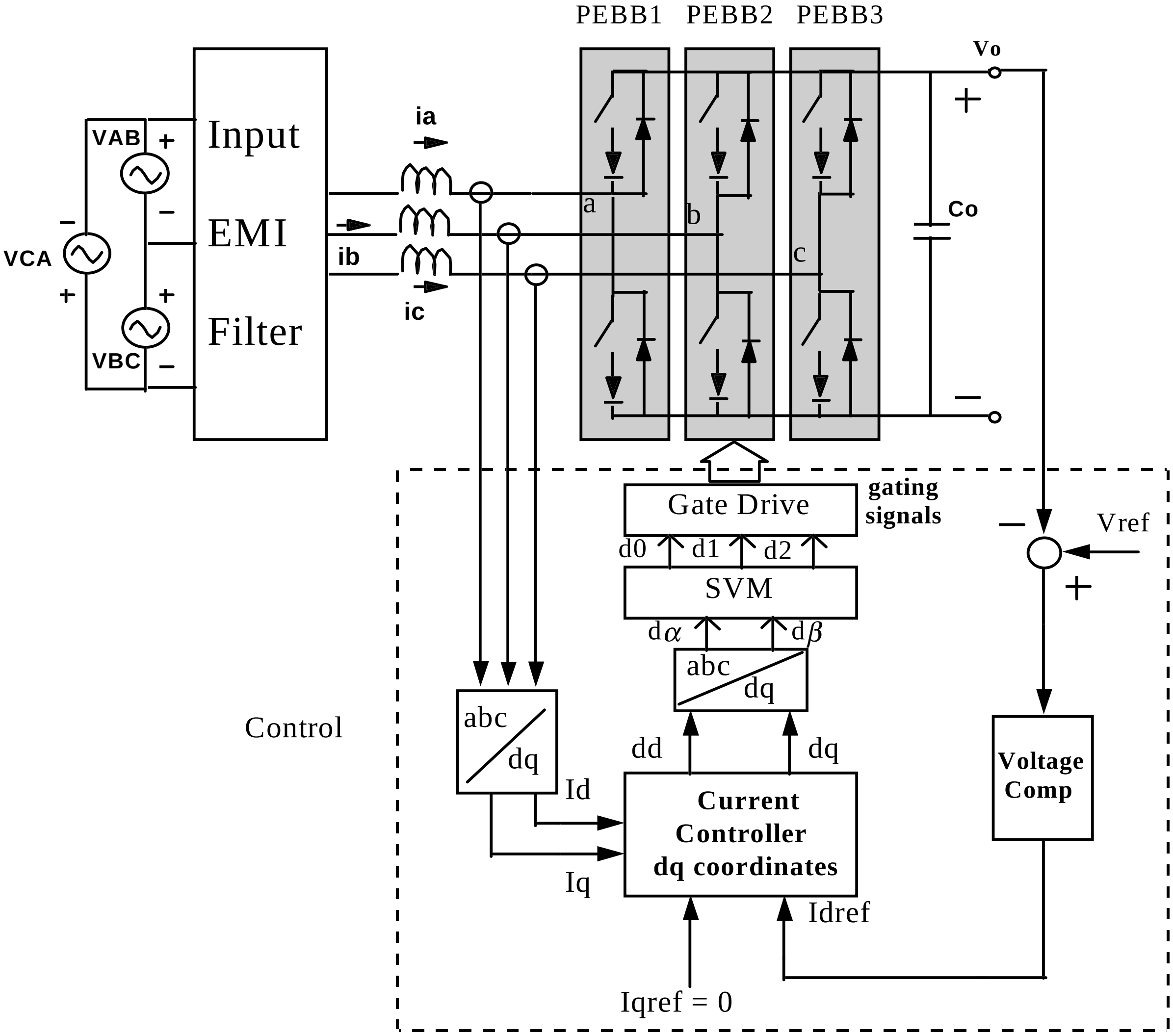


Figure 4.3 Discrete switching model of a digitally controlled boost rectifier

Three cases were simulated:

a) All phase legs are synchronized,

b) The communication delay between two adjacent phase legs is 1% of a switching period, and

c) The communication delay is 5% of a switching period.

The switching frequency of the simulated converter is 20 kHz. Figure 4.4 shows the three-phase current and dc-bus voltage waveforms for all three cases.

These simulation results show that an uneven delay between the modules can cause waveform distortion as the delay becomes comparable to the operating switching period of the converter. The simulation shows that a propagation delay of 1% of a switching period causes minimal difference in waveforms compared to the case without delay. However, a delay of 5% causes significant change in converter operation and its waveforms.

It is noticeable from Figure 4.4c that the input current distortion is not the same in all three phases. In the simulations, the converter power stage and control were balanced fully except that the delay accumulates from one phase to the next. This uneven distribution of the delay among the three phase legs caused the current imbalance and, consequently, the waveform distortion and significant oscillations of the dc-bus voltage.

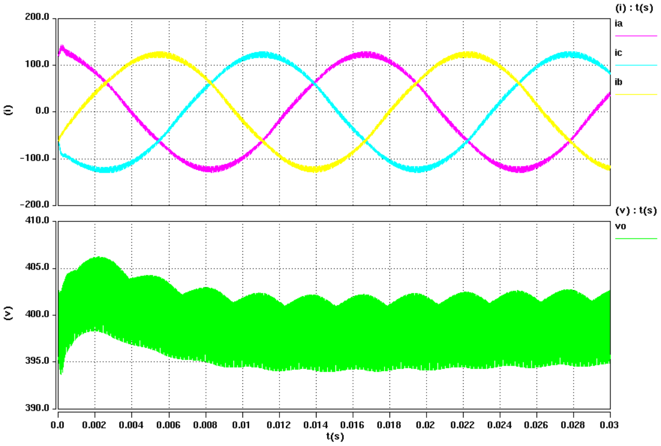
Switching Frequency Limitations

A qualitative conclusion from the simulations is that the performance of a converter with daisy-chained control architecture is not affected significantly by the accumulating communication delay as long as the delay between two phase legs is on the order of 1-2 % of the switching period. This conclusion has been derived from the example of the converter with three phase legs. However, it is applicable to more complex conversion systems.

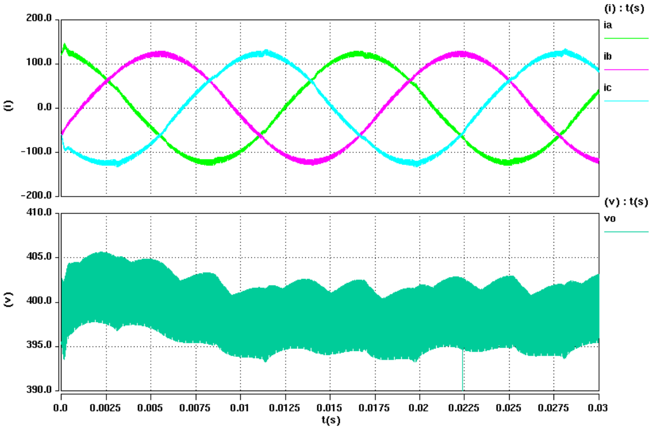
The largest amount of the propagation delay is accumulated at the last node. Assuming that the total allowed propagation delay is 4% of the switching period (i.e. the delay between two phase legs is 2%) the maximum switching frequency, can be written as:

$$f_{sw} < \frac{1}{25 \cdot (n_n - 1) \cdot \tau_d} \quad , \tag{4.1}$$

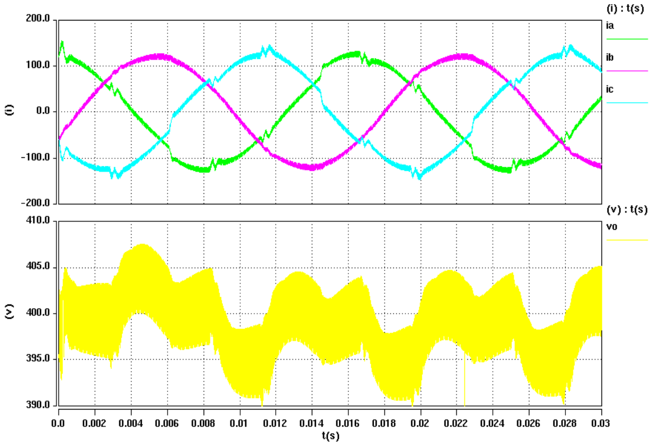
where n_n is the number of nodes (phase legs) which need to be synchronized, and τ_d is the node-to-node delay time.



(a) Case 1.



(b) Case 2.



(c) Case 3.

Figure 4.4 Three-phase currents and dc bus voltage with different propagation delay: (a) no delay, (b) delay is 1%, (c) delay is 5% of a switching frequency

Problem Solutions

Classically, nodes in a communication structure are synchronized using one synchronization command. This is appropriate when the system has a star structure. For a ring topology, there are few more appropriate synchronization methods that compensate the ring propagation delay. This delay can be compensated for in the communication algorithm, hardware, or could be avoided by using a more star-like topology (Figure 4.5). Furthermore, a control design can be modified in such a way that it takes into account the propagation of the ring structure.

The assumption of these algorithms is that the delay through the slave nodes is known. This is true, since the hardware introduces a dominant, fixed delay, which can be measured. Some smaller, variable delay is introduced by the propagation time of a signal through optical fiber. This delay depends on the fiber length, and it is negligible for shorter distances.

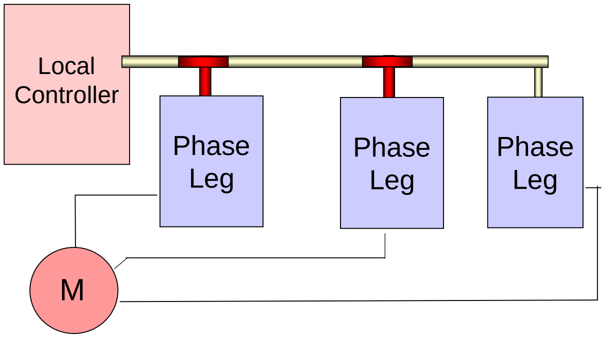


Figure 4.5 Converter topology with a single fiber bus

Ring Synchronization Sequence

The idea of this novel synchronization method is to send a synchronization sequence, instead of a synchronization command. This sequence would initiate synchronization in each of the slave nodes at approximately the same time (see Figure 4.6). The assumption of this algorithm is that we know the delay through the slave nodes.

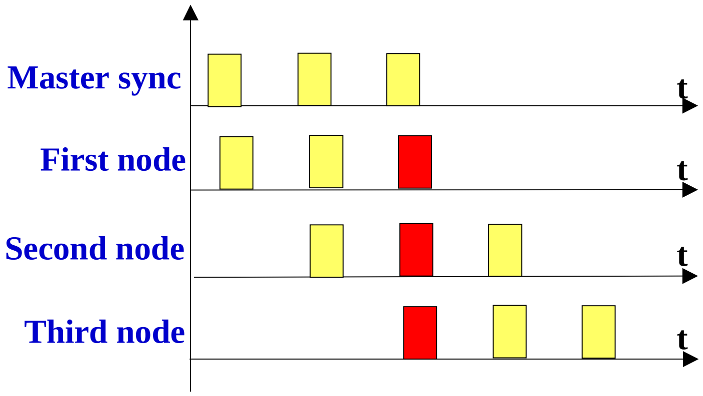


Figure 4.6 Synchronization sequence

The sequence consists of (n+1) fields (Figure 4.7.a), where ‘n’ is the number of nodes that need to be synchronized. The first field is a synchronization command that alerts the nodes to wait for their time to synchronize. Following this are the address fields of nodes being synchronized. After the synchronization command has passed, the node awaits its address field. When it is received (Figure 4.7.b), the node generates an internal synchronization signal.

The importance of the new synchronization method increases at a slower speed of communication, because in that case propagation delay also increases. This synchronization technique can also be applied in motion control networks, such as SERCOS and MACRO.

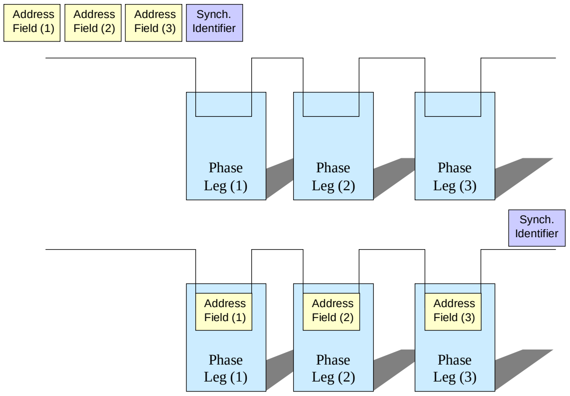


Figure 4.7 Illustration of the ring synchronization sequence in three-phase converters

Hardware Compensation

The hardware compensation of the propagation delay can be done in a similar way. Master node sends synchronization command; after this command is received, each slave node waits for a specified amount of time, before it issues a local synchronization pulse. The wait-time is implemented in hardware, through use of counters.

Depending on the position of the node in a network chain, the implemented wait-time differs, but is closely related to the node's position in a chain. When the node addresses (n_a) are assigned sequentially, so that the node which is first to receive the master node frames is assigned the address '1', and each following node is assigned address incremented by one, the node's address can be utilized for propagation delay compensation. The slave node wait time can be implemented in hardware as:

$$t_w = (n - n_a) * t_d \,, \qquad (4.2)$$

For example, the first node would wait for $t_d * (n - 1)$ after the reception of a synchronization command to issue local synchronization pulse.

This would achieve the same quality of synchronization as the previously proposed synchronization sequence. Benefit over the previous method, is that the time is not wasted to transmit the synchronization sequence. Drawback is that the node position, in reference to a master node and the ring communication direction, has to be known.

In a ring structure, the auto-address assignment procedure in the initialization phase, can be utilized to assign sequential node addresses. More about this procedure can be found in Chapter 5.3.

Topology Modification

One of the solutions to the delay problem of daisy chaining phase legs of the converter is the application of a different communication topology. A multi-tapped bus (Figure 3.1c) is a modification of a star topology. The difference is that it no longer has point-to-point links, but it is a simplex multiaccess system.

The bus, in this topology, is a single fiber (Figure 4.5). Communication signals are sent using two wavelengths, one for each direction of the data exchange. The controller transmits data to the bus using one wavelength, and the power module nodes receive information sent on that wavelength. The modules use the other wavelength to transmit information to the controller. All of the nodes receive information simultaneously, but only the node that data is sent to, can transmit an ‘answer’. If the controller transmits a synchronization pulse, no delay is introduced.

To implement this topology, a “T” connector for POF is needed. The system described in Chapter 5, requires minimal modifications to use this kind of topology. The only change in protocol described in [5], is that a node would not retransmit data intended for other nodes. It would only send its own data when it is addressed.

The solution to the delay problem is not the only benefit of this system. With such a structure, the nodes have just one fiber connector, instead of two. Reliability is also improved since a malfunction of one slave node does not mean the failure of the complete ring. A drawback is that the necessary hardware has not matured. An algorithm of self-assignment of node addresses cannot be implemented. Instead, they have to be programmed into the device or assigned by hardware.

Chapter 5 POWER ELECTRONICS SYSTEM NETWORK - PES NET

PES Net is a control network tailored to the particular needs of power electronics systems. It is intended for use in power-converter systems for control data exchange between a power stage and a digital controller.

The task in designing this protocol was to come up with a smarter way to utilize the optical fiber link between the digital signal processor that controls the system and each of the phase legs or loads (nodes) of the converter system. This consequently would improve the reliability of the system, ease the hardware debugging process, standardize the interface and reduce the cost of building the system. Fewer fibers would be used and the number of transmitters, receivers, and engineers' time to implement control communication on such system would be brought down.

PES Net is, in fact, a more sophisticated way of using TDM. Information is sent using data packages, which have their destination address. Since asynchronous communication is implemented, the beginning of each packet is marked by pre-specified command.

A choice had to be made in respect to network topology. Because of the many benefits that the ring structure provides, and specifically because just the basic fiber-optic components are needed, the decision was made to implement this structure. It is very important to use a simple technology, which does not require extensive user training. It is much easier to deal just with optical fibers, transmitters and receivers, than also with splices and optical stars.

The control network design for converter module communication is based on a MACRO and FDDI protocol.

5.1 PROTOCOL FUNCTIONING

In the communication sense, this is a ring structure, as shown in Figure 5.1. One fiber leaves each of the nodes, and one comes in. Information is sent in only one direction. For the sake of redundancy, a double ring can be implemented.

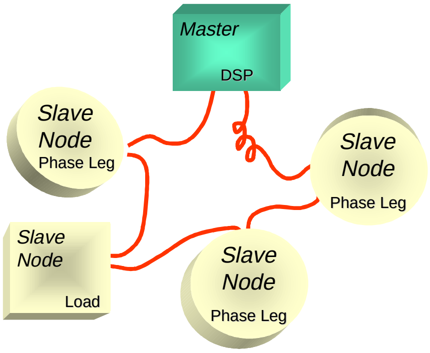


Figure 5.1 Ring nature of proposed topology.

This is a master-slave structure, where converter controller is a master and power modules are slaves. The master first transmits data packets for each of the slave nodes and then follows with a synchronization sequence. One communication cycle corresponds to one switching cycle of the converter.

In the passive state, slaves are just passing data to the next node. When the node recognizes its own address, instead of passing data to the next node, it starts sending its own data, while it stores the received data, as illustrated in Figure 5.2. When the packet is received fully and transmitted, node goes back to the passive state. Data measured on a power stage and status information are delivered to the controller.

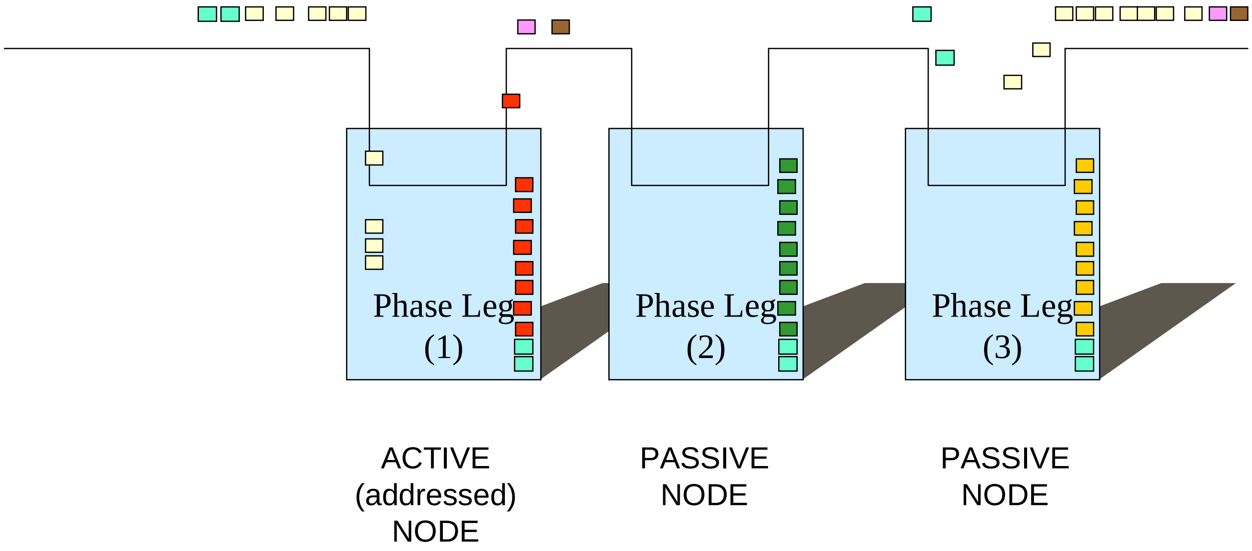


Figure 5.2 Ring Structure Functioning: Node (1) recognizes its address, transmits local data, and stores received data. Other nodes are just passing data down the line.

5.2 PROTOCOL LAYERS

The PES Net protocol does not have all of the proposed ISO/OSI layers. The implemented layers are described below.

Physical Layer

In the designed system, the PL is plastic optical fiber link. Data that needs to be sent is transmitted in the light form over the link. The Hewlett Packard's HFBR-0507 [29] series optical transmitter and receiver are used. The transmitter is a 650nm LED (HBR-1527) [29]. The receiver is a PIN receiver HFBR-2526 [29]. They allow data transmission of 125 Mb/s over 100 m. These components can be used with POF or HCS® fiber [15]. Operational wavelength is

650 nm (red light). The fact that visible light is used is beneficial for debugging and safety purposes.

Data Link Control Layer

In the FDDI standard a 4-bit data is coded into 5-bit nibbles (4B/5B), making use of the most efficient combination for transmission with a NRZI scheme. In this scheme, ones are represented by transition, while on zeroes the signal remains the same. Use of NRZI coding makes clock extraction from the data stream possible. The used AMD’s communication chip, TAXIchip^{TM} [28], receiver automatically does this.

The output of the optical receiver and input to the optical transmitter is a pseudo-ECL [29] data sequence. Complementary pseudo-ECL signals are used. A transition between two levels represents logical ‘one’, while the lack of transition represents logical zero. This representation of the signal is particularly good for sending data over long distances or in noisy environments. A two-line complementary input/output of the fast signals is very appropriate for printed circuit board (PCB) routing, because it picks up less EMI noise.

TAXIchip™ transmitter and receiver perform functions in this layer. Data is supplied to it on an 8 bit wide bus. TAXIchip™ adds two more bits to every 8 bit word, based on 4B/5B coding. This coding uses combinations of ones and zeroes, which are best for clock extraction and resynchronization on the receiver side. This coding is regulated in the FDDI standard, [28].

Network Layer

Data Formats

There are three types of data frames defined in the PES Net. They are: command frame, data frame and synchronization sequence frame, as shown in Figure 5.3.

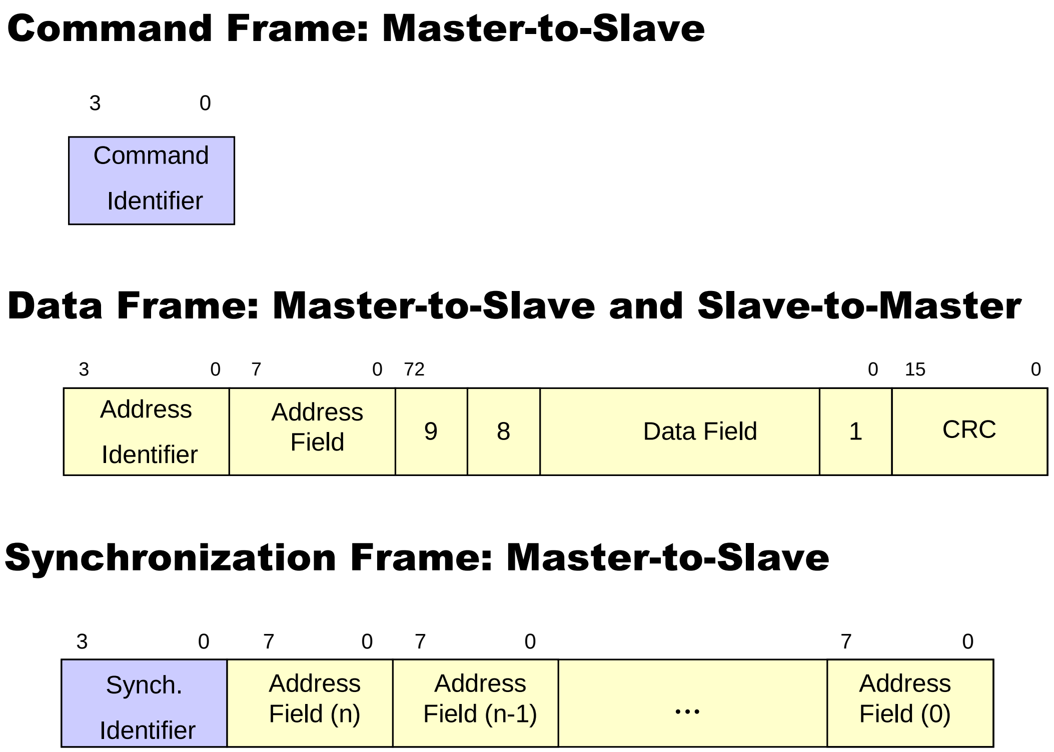


Figure 5.3 Data formats

Data Frame

Data frame is 13 words long. Nine bytes are for data, one is the address field, two are CRC and one word is the address identifier, as shown in Figure 5.3.

The address identifier command marks the beginning of the data frame. Following is the node address and data fields. The address field is 8 bits long, so theoretically up to 255 nodes can be supported. In MACRO protocol, the address field is divided in two. The first part determines the master address, while the second one determines the slave address. This allows the ring to have multiple masters. In the PES Net, the upper three bits are reserved for possible multiple master addresses, while the lower five bits can address 32 slave nodes.

The frame sent by the master has eight bytes intended for real-time data (PWM information) and one for non-real-time data. The PWM signal information can be sent in two ways. The simplest way is to send two digital 8-bit values: the ‘up’ time, t_{up}, and the ‘down’ time, t_{dn}. Based on these two values, on-board digital logic can generate a PWM signal (Figure 5.4). Two time values are given and implemented in reference to the synchronization pulse. The controller sends these two values for every switch.

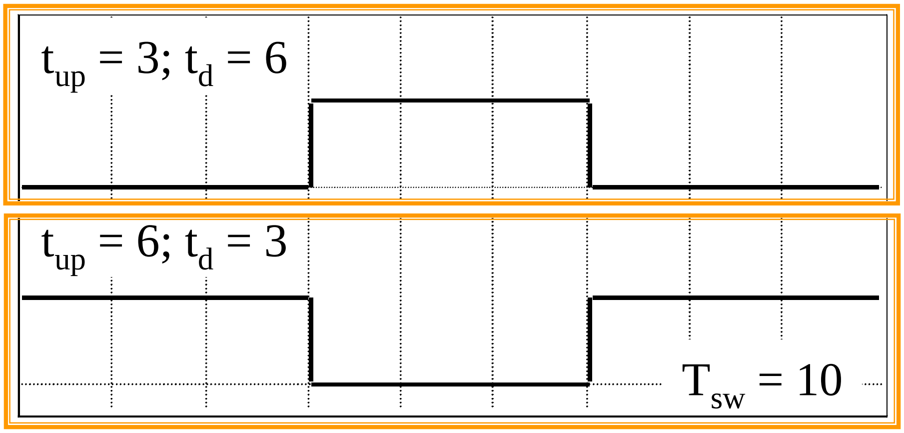


Figure 5.4 Generation of PWM signal based on two-byte digital information

Table 5.1 Adopted Data Field Format in Master-to-Slave Data Frame

Data Byte	***Purpose***
Data 9	Non-real time
Data 8	t_{down} for the switch n, higher byte
Data 7	t_{down} for the switch n, lower byte
Data 6	t_{up} for the switch n, higher byte
Data 5	t_{up} for the switch n, lower byte
Data 4	t_{down} for the switch p, higher byte
Data 3	t_{down} for the switch p, lower byte
Data 2	t_{up} for the switch p, higher byte
Data 1	t_{up} for the switch p, lower byte

In the soft-switching case, the PWM signals can be locally generated. This is possible because there is some form of local intelligence on the PEBB board and because the PWM signals for the auxiliary switches are related to the PWM signals of the main switches. The shape of the auxiliary PWM sequence depends on the used soft-switching technique, but both PWM signals can be generated based on the t_{up} and t_d for the main switch. Consequently, the auxiliary PWM signals are generated on the slave board, without sending any extra information. Parameters necessary for the auxiliary switch PWM generation can be negotiated in the initializing phase. The other option is to allocate a space in a data frame for auxiliary switches t_{up} and t_d, and treat them, in communication terms, in the same way as the main switch transition times. Any other combination of different timing instants with different resolutions can be implemented, as long as the total information can be coded with 64 bits in every switching cycle.

Table 5.2 Adopted Data Field Format in Slave-to-Master Data Frame

Data Byte	***Purpose***
Data 9	Status
Data 8	Non-real time data
Data 7	Non-real time data
Data 6	Real-time data, higher byte
Data 5	Real-time data, lower byte
Data 4	Voltage, higher byte
Data 3	Voltage, lower byte
Data 2	Current, higher byte
Data 1	Current, lower byte

The frame sent by a slave node has six bytes intended for real-time information (digitized current and voltage values) and three for non-real-time data (temperature, fault and status information). The data formats have been adopted in this thesis are given in Tables 5.1 and 5.2.

Synchronization Frame

The algorithm implemented is a synchronization sequence described in Chapter 4. The master sends a synchronization sequence when synchronization is required. Each slave node recognizes the ‘Sync’ command and goes into Synchronization-state, within which it waits for its own address. When the address is received, it generates a synchronization signal for its phase leg.

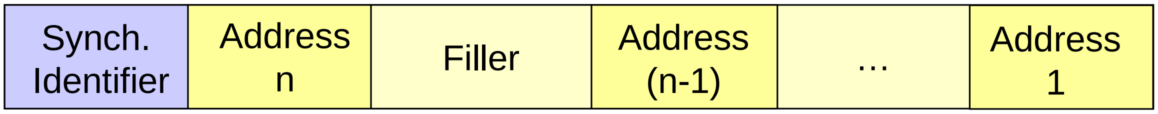


Figure 5.5 Synchronization Frame

Format of this sequence is shown in Figure 5.5. Frame starts with Synchronization identifier, and is followed by the several 8 bit long data blocks containing addresses of slave nodes and ‘filler’ fields. The first address to be transmitted is of the slave node that is last to receive the frame. The number of address data blocks sent equals the number of slave nodes on the ring, which we want to synchronize.

The ‘filler’ is used to implement the delay into synchronization frames. From the hardware and software implementation point, the simplest way to define the filler is in terms of data words. Instead of having an extra timer that would measure the desired delay, it is much simpler to send several empty data bytes in between address fields of a synchronization sequence.

For example, in the implemented three phase system the measured propagation delay through one node is 468 ns. This delay is approximately equal to 4 or 5 data words, because it takes 100 ns to transmit one word at ring speed of 125 Mb/s. If 4 bytes are used, the synchronization sequence is shorter, so the bandwidth of the whole system is affected less. In the case that a 5-word delay is used, synchronization is more precise.

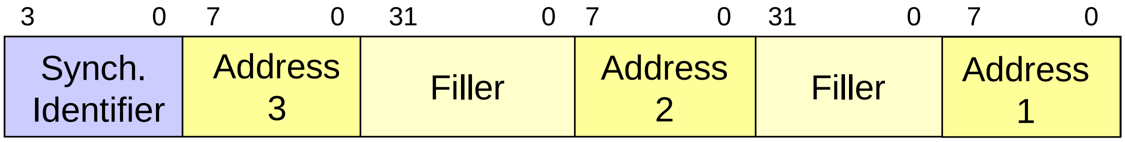


Figure 5.6 Synchronization frame for the three-phase converter

The synchronization sequence for a three-phase system (one master and 3 slave nodes), shown in Figure 5.6, has a format:

- A synchronization identifier is followed by the address of the node that is the last one to receive the frame (node 3 in Figure 5.2.),
- After a 5-byte delay (the node address and 4 empty data words, which takes 500 ns to transmit), it sends the address of node 2.
- After another 5-byte delay, it sends the address of node 1.

Use of this algorithm reduces the propagation delay from 468 ns to (500 ns – 468 ns) = 32 ns. This means that its implementation delay is reduced by 93%. If needed, an even better synchronization can be achieved, but a more complicated implementation is required. For that purpose, a delay would be defined in terms of master node's clock frequency or as fraction of data words.

This algorithm relies on a specific ring topology. The delays are independent of the switching frequency of the converter, but are related closely to the data transmission rate. An increase in communication speed causes a decrease of propagation delay.

There is one more assumption made, and it concerns the state of a TAXIchip: the slave node has completed sending the last package-byte to-be-transmitted. This should always be true because the slave speaks only when spoken to, and transmits the same number of bytes that it has received. Otherwise, a TAXIchip may have a buffered byte of data that is waiting to be sent, which would distort the timing of a synchronization sequence.

It is a good practice to induce a taxi-synchronization command [28] to be sent prior to the ring-synchronization sequence. This forces the TAXI-receiver to re-synchronize its clock to the TAXI-transmitter clock. Because of this, the time between the last byte transmitted and the beginning of the synchronization sequence should be about 1.5 of the line-word length (duration of 15 line-bits or 120 ns), but, if possible, it could wait the time for the full length of the package to be transmitted.

Slave nodes use the synchronization pulse generated during the synchronization sequence as a time-marker for resynchronization of the digital counter logic, which controls A/D converters and PWM generator. Use of A/D with a built-in multiplexer allows a use of the same A/D for the conversion of current, voltage and temperature. The most critical and fastest changing variable, such as current, should be measured last and as close as possible to the time when data is sent to the controller. The least critical and slowest variable, such as temperature, should be measured first.

Command Frame

Command frames consist of one word - the actual command, as shown in 5.3c. Since TAXIchip™ is coding 8 bit data word as a 10 bit word (because of 4B/5B coding) [28], not all of the combinations are utilized for data word coding. The unused combinations are defined as commands. TAXIchip™ receiver has a separate output bus for data and separate for commands.

A total of 15 different commands can be specified. Six are predefined.

1. Address Identifier
    - Marks the beginning of the data frame.
2. Network Synchronization Identifier
    - Marks the beginning of the synchronization frame.
3. Shutdown Identifier
    - Sent by the master when shutdown of the system is desired.

4. Mater-Reset Identifier

   - Sent by the master when reset of the system is desired.

5. TAXI Sync Command

   - Used to keep TAXI-Rx synchronized to TAXI-TX. This is generated automatically by TAXI-TX.

6. Broken-line Indicator

   - No light in the incoming fiber.

In the case of a broken line (no light in an incoming fiber), TAXIchip will detect this as a reception of a ten zero bits. This is interpreted as a command “1111 1111”. This command is not used in this protocol, so reception of such a command indicates the broken line or improper function of TAXIchip™ transmitter. In the case when there is no data to send, the TAXI-TX would send a TAXI synchronization command, and it would never allow TAXI-Rx to receive the stream of 10 bit zeroes, since it can be interpreted as a valid command. This particular command can be used for the detection of a line failure. This sequence is not used for any other purpose by the protocol, so the node knows that failure has occurred in the case of a repetitive reception. After a failure is detected, the block can employ a converter shutdown.

5.3 PROTOCOL IMPLEMENTATION AND APPLICATION

Hardware Implementation of Master and Slave Nodes

The architecture of a master and the slave hardware is similar. The difference is that the master does not need the components intended for the interface with the converter (gate drives, A/D and PWM generator), but it has a digital signal processor (DSP) with implemented control algorithms. A filed programmable gate array (FPGA) is used for communication interface in both cases. In the case of the master node, the FPGA can be of a smaller capacity. The master's

FPGA is programmed in such way that the DSP ‘sees’ it as Read registers that contain sensed values (currents and voltages) received from slave boards and Write registers in which the newly calculated duty cycle information is stored.

A master node (Figure 5.7a) contains a DSP, boot memory, FPGA, optical and TAXIchip^{TM} receiver and transmitter. A slave node (Figure 5.7b) has a FPGA, boot memory, optical and TAXIchip^{TM} transmitter and receiver, gate drives, sensors, and A/D. All phase leg control modules (slave nodes) are the same.

Use of TAXIchip^{TM} receiver and transmitter [28] chips makes the FPGA design easier and its speed requirements lower. It provides all of the multiplexing, demultiplexing, encoding, decoding, and time recovery functions. TAXIchip is used as sophisticated serial-to-parallel and parallel-to-serial register. Parallel data output or received by FPGA at a speed of 10 MHz is converted to a pseudo-ECL serial 125 MHz datastream using the FDDI standard.

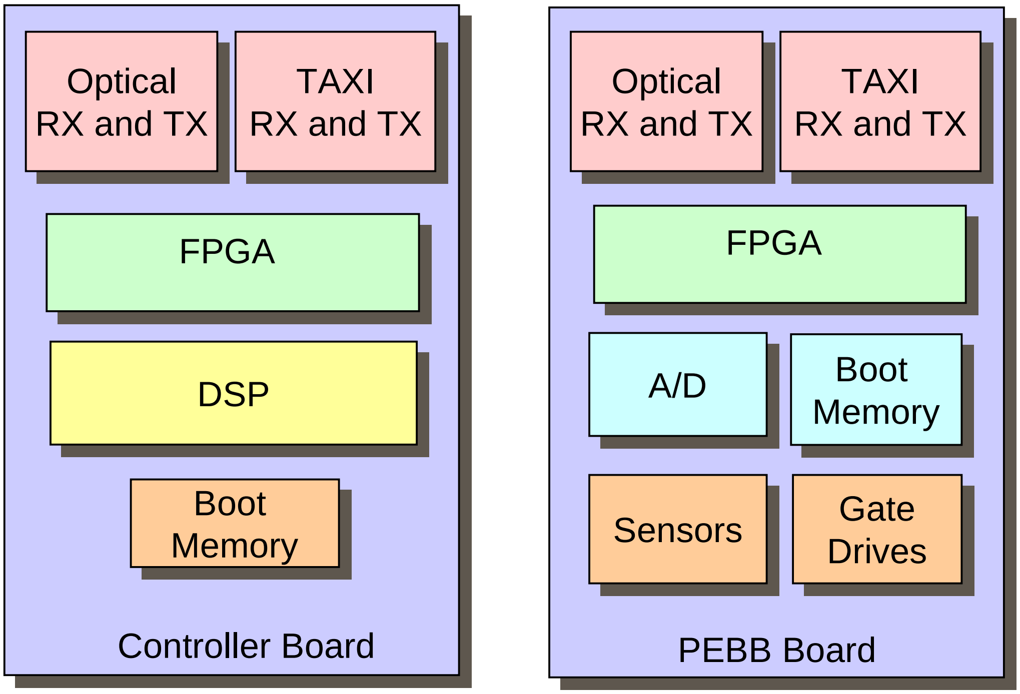


Figure 5.7 (a) Master node structure; (b) Slave node structure.

The optical transmitter and receiver operate at 125 Mb/s [29]. They can be located on a smaller independent board that can be used as an FDDI transceiver and can be interchanged easily with an optical transceiver that supports a different type of optical fiber.

The slave node's functional blocks are implemented in an FPGA, as shown in Figure 5.8. The communication control block is the most complex. It controls the operation of TAXIchip™ receiver and transmitter, performs communication protocol functions, and communicates with other functional blocks within the node.

The PWM Generator block produces PWM signals for the node gate drives. The data necessary for this are stored in PWM Memory block. The Sensor Control block supervises the operation of A/D converters and stores the measured values. The Fault Control block keeps track of the fault events and reports them to the master node. This block includes a watchdog timer (WDT) that is activated in the case that the valid data has not been received for several communication cycles. The CRC block generates and examines cycle redundancy check (CRC) fields in the data frames.

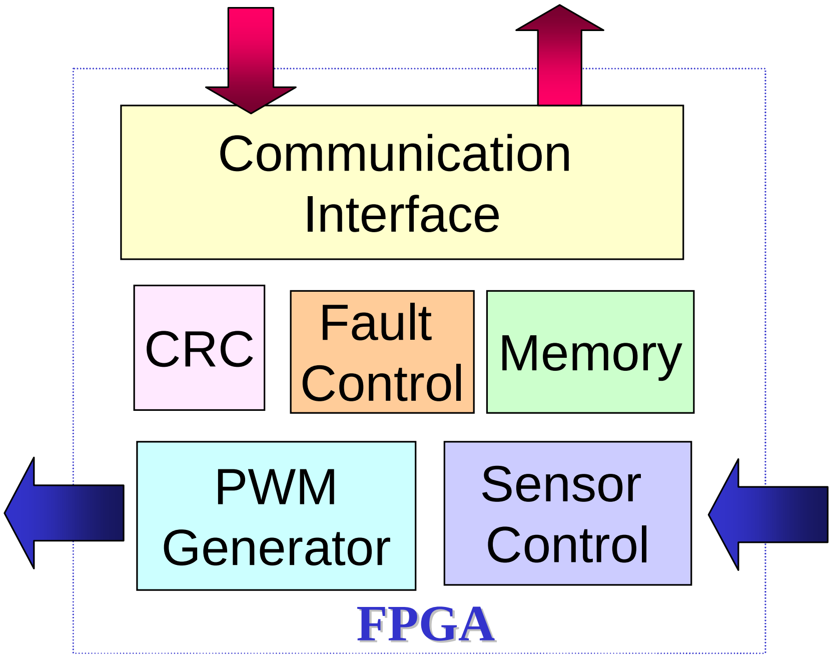


Figure 5.8 Functional blocks implemented in slave FPGA.

Communication Control in Slave Nodes

PES Net communication control is built into the hardware, specifically in an FPGA device. This results in a faster performance. It is easiest to describe and implement the operation of the designed communication system using state machines [30] and VHDL [31]. State machines are supported in the Altera's MAX plus software for programming FPGAs in both VHDL and AHDL [32].

The state machine concept implementation is represented graphically in Figure 5.9. 'clouds' represent the states. The arrows represent the state transitions. When a specified condition (written on top of the arrow) is fulfilled, or an appropriate 'event' occurs, the state change occurs.

In the slave nodes, the events that cause 'change of state' are commands received from the master through the optical fiber (address identifier, synchronization identifier, shutdown) and signals from within, or from outside of the FPGA. States act as memory elements that keep track of the system history and generate appropriate actions for a specific event.

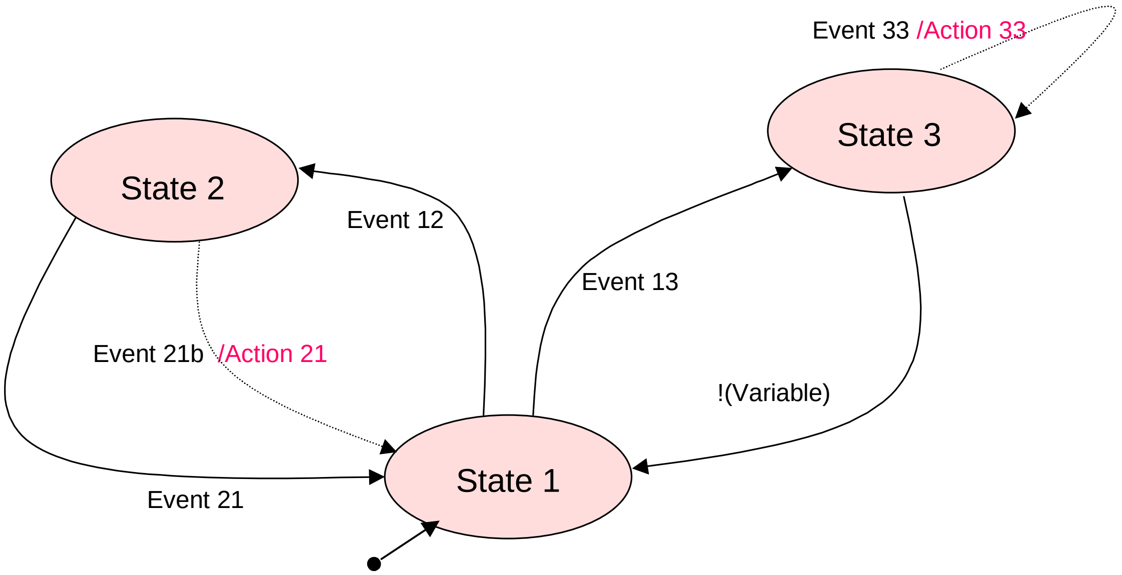


Figure 5.9 State machine concept

Principal state diagram of the communication control block is shown in Figure 5.10. Defined states are:

1. Wait State
2. Address State
3. Active State

4. Synchronization State

5. Shutdown State

6. Initial State

The slave node starts out in the Initial state. In Initial state power switches are always off, and cannot be turned on. The purpose of this state is to assure that the ring is operational before the power converter is turned on.

Once the node receives an initialization command, it goes to the Wait state. From this state, depending on the received command, it can go to Address state (used to check the data frame address), Synchronization state (where it waits for a local address so it would synchronize), or Shutdown state (if the watchdog timer activates or Shutdown command is received).

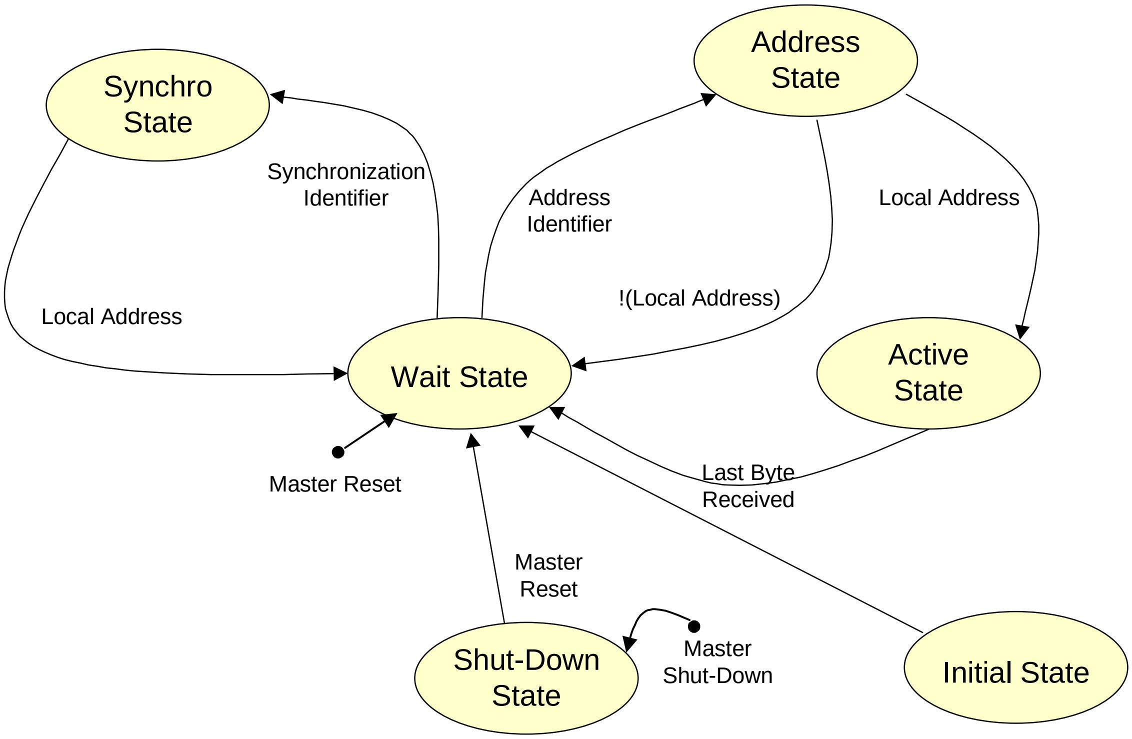


Figure 5.10 Basic state machine diagram of the communication control block

The complete state diagram must include fault conditions, which are described next.

Fault Types

In the network, faults can occur for the various types of reasons. In the PES Net they are classified as the hardware, communication, and protocol faults. The hardware fault alerts the system of the mal-functioning of the hardware. The communication fault is identified when there is a problem with a network connection (a fiber break or bad reception). The protocol fault occurs when an unexpected command is received.

Some of the faults are fatal and require the immediate system shutdown. The others are just reported to the master node in the ninth byte of the data field. Flag registers at each of the nodes keep track of the fault events. After the faults, along with the measured variables, are reported to the master node, flag registers are reset. These registers and the associated logic form the Fault control block in Figure 5.8. Implemented fault types are briefly described below.

PEBB Fault

Hardware fault. A phase leg is generating a fault signal because the gate drive detected a switch over-current condition. Can be generated while in any state. Fault is reported to the master. Communication operation continues as normal.

Watchdog Timer Fault

Communication fault, connection failure or communication control block failure. When a fatal communication error occurs, and no data is received for the time specified in a software code, Watchdog timer shuts down the converter. Can happen in any state. The system goes into the Shutdown state.

VLTN Fault

Communication fault. The TAXI receiver has generated a ‘violation’ message. There was a problem receiving data from the serial line. Can be generated while in any state. If the system is in Active-state or Shutdown-state, it remains there; otherwise it goes to Wait-state. VLTN flag is set to high, data received is assumed to be all zeroes. The Fault Control block generates a message to notify the Watchdog timer.

Protocol Fault

Protocol fault. Unexpected command received. Generated when node is expecting data or its address, but receives an unexpected command instead. For example: node that is in the address state waiting to receive its address instead receives synch identifier. Can be generated while in Address-state, Synchronization-state or Active-state. If the system is in Active-state it remains there, so it could resume sending of sensed signals. Received data is assumed to be all zeros. The Fault Control block generates a message to notify the Watchdog timer.

Invalid Data

Protocol fault. Data that is received is, for some reason, labeled invalid. Such data is not used by the system. The fault is reported to the master. The Fault Control block generates a message to notify the PWM Generator, so it would produce PWM signals based on old data.

The source of the error may be:

- A data packet is smaller than expected: one or more data bytes were rejected along the communication line. It may be caused by a VLTN fault in the preceding node. It is detected when the command is received while the node is waiting to receive data (Active-state). Perhaps the expected detection method would be to see if nine continuous data bytes are received, but the communication hardware and software do not require data to be received in consecutive clock cycles.

- VLTN fault is received while node is receiving data (Active-state).

- The Watchdog timer is activated, so all the data is marked as invalid.

- Reception of master shutdown has the same effect.

CRC Check Fault

Communication fault. The CRC check sum does not match the received data frame. It should be generated while the node is in Wait State. The fault is reported to the master; old data will be implemented, just-received data is discarded. This fault is not reported to the Communication Control block.

Information about Fault events is reported to the master node in each communication cycle.

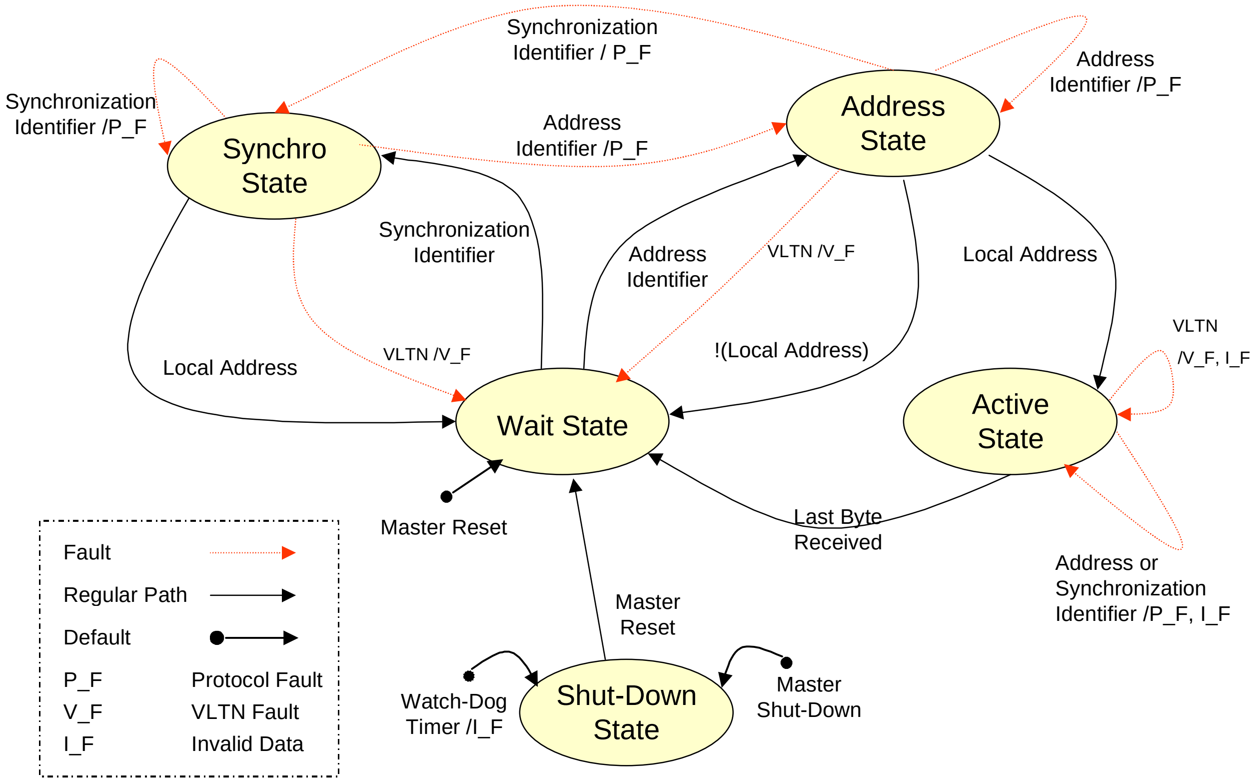


Figure 5.11 State machine implementation of the communication control block.

State-Machine in Communication Control Block

The behavior of the slave node’s Communication Control block is implemented using State machine concept, as shown in Figure 5.11. Modified diagram is used to implement the Communication Control block in the master node.

In the master node receiving and transmitting functions are separated. The transmitting block is simple and does not require use of state-machine concept. The receiving part utilizes this concept. In this case, the Address-state is unnecessary, because the master node receives all the data and uses the address field as a pointer to the memory location, in which the data should be stored. The Synchronization-state is also unneccessary, because the controller does not require high-precision synchronization with phase-legs.

Wait State

This is the default operational state. Received data or commands are just passed to the next node. If a valid command is received, the system enters the appropriate state.

Fault Case: *If VLTN fault occurs while in this state, the fault is not reported because that data is from/for some other node; no data is sent instead. That error will be detected and reported as Protocol fault at the reception point, since the data frame will become at least one data byte shorter, depending on the number of VLTN faults.*

Address State

The system can be in this state until the next byte is received. The state is entered if the Address identifier is received. If the frame's address field matches the node address (local address), the node goes to Active-state, if not, it goes back to Wait-state. Data received in this state is passed down the ring.

Fault case: *If a synchronization command is received while in this state, the system goes to Synchronization state. The information from the last, correctly received, data frame is going to be implemented. The Protocol fault is reported. If the received command is an Address identifier, the system remains in this state, but reports the error as a Protocol fault.*

Active State

This state is entered from the Address state. Data received in this state is stored locally, while the measured signals are sent to the master (i.e. passed to next node). The system goes to Wait state when all data is received or a Shutdown or Reset command is issued.

Fault case: *The highest priority in this state is given to the data transmission, once it has started. In other states, where no locally stored data is sent from the node, priority is given to the newest received command. Therefore in the case of protocol violation, such as the arrival of Synchronization identifier (arrival of a Shutdown or Reset command is not a protocol violation), the system remains in* *this state so that the node can finish sending sensed data, but a fault is reported. In case of a fault, received data or commands are not used.*

Synchronization State

The system enters this state after it has successfully received a Synchronization identifier. It remains in this state until the local address is received. When the local address is received, the synchronization flag is set to one. The system exits to a Wait state. Data received in this state is passed down the ring.

Fault case: *The VLTN signal sends the system to Wait state and VLTN Fault is reported. A protocol fault occurs when the Address identifier is received while in this state: the system goes to an Address state and a Protocol Fault is reported.*

Shutdown State

The system can exit this state only when a master-Reset command is received. This state can be entered when either a master Shutdown command is received or the Watchdog timer expires. A slave node keeps passing the data on the ring in order to keep the ring functional. The Shutdown command itself must be passed on. There is no fault case defined for this state.

Initial State

This is the default state on start-up. The node never enters it again. It is used for an initial network check-up. In this state, the node waits for the initialization command from the master node. After it is received, the node changes its state to a Wait state. Until that event occurs, the node passes data down the line. When the master node receives the initialization frame back, it knows that it can start sending control information. If there is some fatal fault at the initializing node, it does not pass on the initialization command.

5.4 MAXIMUM SWITCHING FREQUENCY

To calculate the maximum switching frequency of a converter using PES Net, the following assumptions are made:

1. There is only one master and all slave nodes are PEBB modules;
2. The switching period and the control sampling period are the same;
3. Maximum computational and communication delay must be less than two switching periods.

Under these assumptions, the communication and functional timing diagram of the converter is shown in Figure 5.12.

When the controller finishes the control algorithm calculations, master node initiates sending of Master Data Frames (MDFs) to each slave node. After certain ring propagation time (T_P) these frames are received in the slave nodes. The MDFs are substituted on the ring with the Slave Data Frames (SDF). The 'Sync' signal is triggered by the interrupt generated by an on-board crystal. It initiates sending of a Synchronization frame. The T_{SYNC}, the time necessary to transmit the synchronization frame, is larger that T_P, the ring propagation time, because the propagation time is built into the synchronization sequence.

The analog to digital conversion (A/D_I) on the power stage should be performed so that it provides the most accurate variable values for the controller. The data conversion process should be finished as close as possible to the moment when the slave node sends SDF. The A/D conversion can start at any prior time, provided that there is sufficient time to finish the measurement process. The A/D conversion time is marked as $T_{A/D}$ in Figure 5.12.

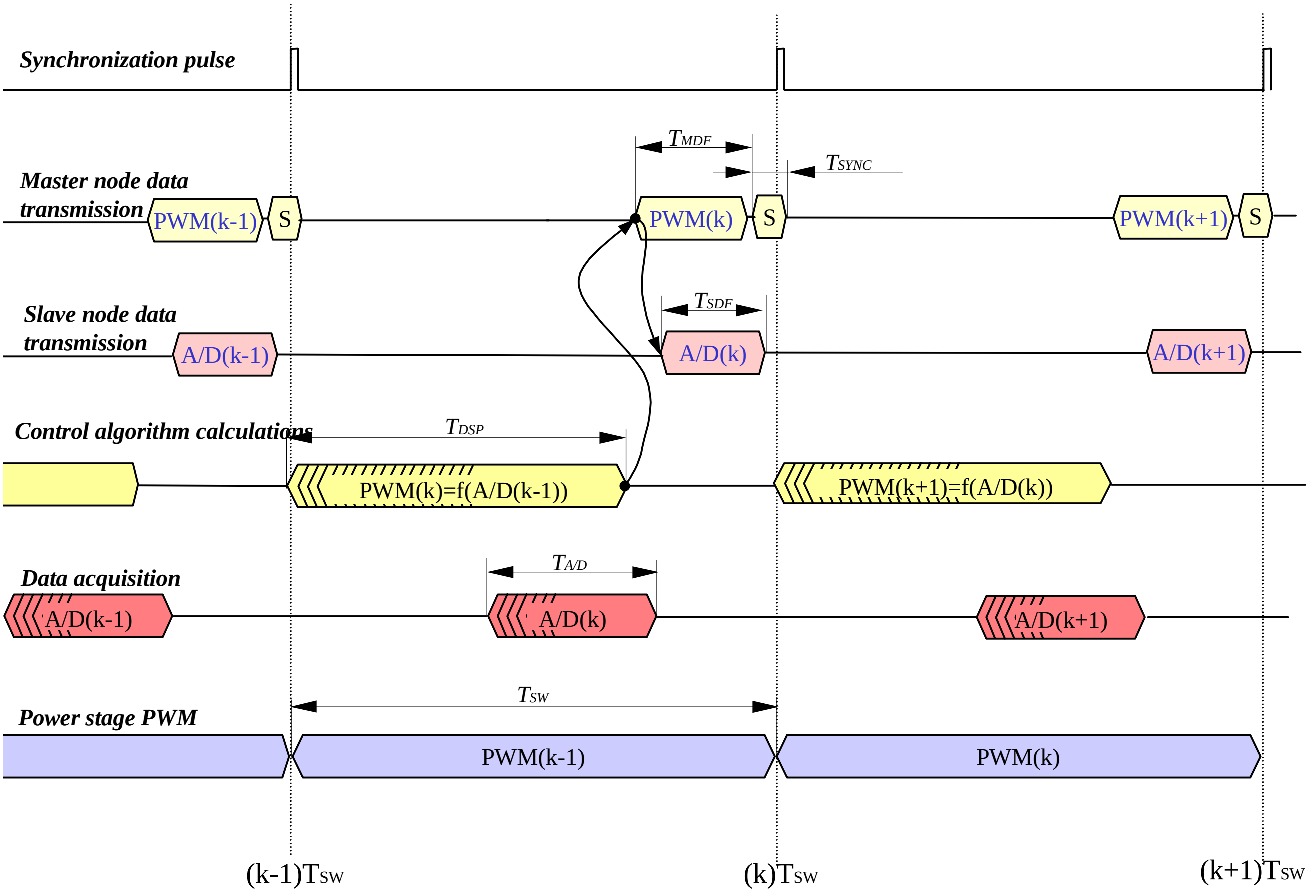


Figure 5.12 Communication cycle timing diagram

It is obvious from Figure 5.12 that the switching period must satisfy

$$T_{SW} \geq T_{DSP} + T_{MDF} + T_{SYNC}\,, \qquad (5.1)$$

where T_{DSP} is the time that DSP requires for recalculating the control algorithm and T_{MDF} is the transmission time for sending MDFs.

The protocol is designed to receive data, while transmitting, so the transmission time for MDFs equals the transmission time for SDFs. This data transmission time can be calculated as:

$$T_{MDF} = n_n * t_{df}\,, \qquad (5.2)$$

where n_n is the number of slave nodes on the ring and t_{df} is the time needed to transmit one data frame.

Duration of one data frame can be calculated as:

$$t_{df} = t_w * n_{wdf} , \tag{5.3}$$

where t_w is a word transmission time and n_{wdf} is the number of the words in a data frame.

A data word has the same duration as a command word. This is because the TAXIchip transmitter codes an 8-bit data word, using 4B/5B coding, into a 10-bit word; a 4-bit command is also coded into 10-bit long equivalent. Duration of one word can be calculated as:

$$t_w = c_b / r_b , \tag{5.4}$$

where c_b is the number of bits in a line word, which equals 10, and r_b is a bit-rate used for transmission of optical signals (for PES Net it is 125 Mb/s).

Duration of a synchronization frame is:

$$t_{sync} = (n_n + 1)\ t_w + (n_n – 1) * t_{sd} , \tag{5.5}$$

where t_{sd} is the time delay implemented into the synchronization sequence. This delay is implemented as the number (n_{sd}) of empty words:

$$t_{sd} = n_{sd} * t_w , \tag{5.6}$$

Total propagation delay can be calculated as:

$$T_p = n_n * t_{np} + t_{mp} + t_{cable} , \tag{5.7}$$

where t_{np} is a propagation time through the slave node, t_{mp} is propagation time through the communication part of the master node, and t_{cable} is propagation time through the optical fiber (negligible compared to other values).

Propagation time through the slave node depends on a propagation time thorough the TAXIchip™ receiver ($t_{TAXI-RX}$) and transmitter ($t_{TAXI-TX}$), and through the FPGA (t_{FPGA}):

$$t_{np} = t_{TAXI-TX} + t_{TAXI-RX} + t_{FPGA} , \tag{5.8}$$

The maximum $t_{TAXI-TX}$ and $t_{TAXI-RX}$ equals two t_w; minimal is equal to one t_w. The t_{np} is approximately the same as t_{mp}.

Based on equations 5.1 to 5.6, the maximum switching frequency is:

$$f_{sw\max} < \left(\frac{1}{\mathrm{T_{DSP}} + \mathrm{n_n} * \mathrm{n_{wdf}} * \mathrm{t_w} + (\mathrm{n_n} + 1)\mathrm{t_w} + (\mathrm{n_n} - 1)\mathrm{n_{sd}} * \mathrm{t_w} + \mathrm{t_{mp}}} \right). \qquad (5.9)$$

When the values of t_w= 13 and n_{wdf}= 80 ns are used, and it is assumed that: $t_{np} \cong t_{mp} \cong n_{sd} * t_w$, this equation equals to:

$$f_{sw\max} < \left(\frac{1}{\mathrm{T_{DSP}} + (14 + \mathrm{n_{sd}}) * \mathrm{n_n} * 80\mathrm{ns}} \right), \qquad (5.10)$$

where T_{DSP} and t_{np} depend on implementation; n_{sd} depends on t_{np}, and n_n depends on the converter’s topology.

Chapter 6 EXPERIMENTAL VERIFICATION

6.1 TEST SETUP

System testing was performed in several phases. For all of them, the same hardware was used, but with a different FPGA code. A FPGA device can be easily re-programmed from personal computer (PC), on each power-up.

Control signals were measured with a Hewlett Packard 1661CS Logic Analyzer / Oscilloscope and using a Tektronix TVC501 Time - Voltage converter (TVC) (Figure 6.1). The Logic Analyzer was used to observe digital signals generated by the FPGA. The TVC transforms a PWM sequence into the appropriate low-frequency waveform, i.e. it performs pulse-width demodulation. The output of TVC is observed on an oscilloscope.

Three identical test boards were built (shown in Figure 6.2). These boards contain TAXIchip™ receiver and transmitter, crystal, and an optical interface board with HFBR-1527 and HFBR-2527 [29]. Each of the boards plugs into one Altera UP1 Educational Board [33]. The UP1 board contains two FPGAs and supporting hardware. The FPGA used is FLEX®10K20RC240-4 [34].

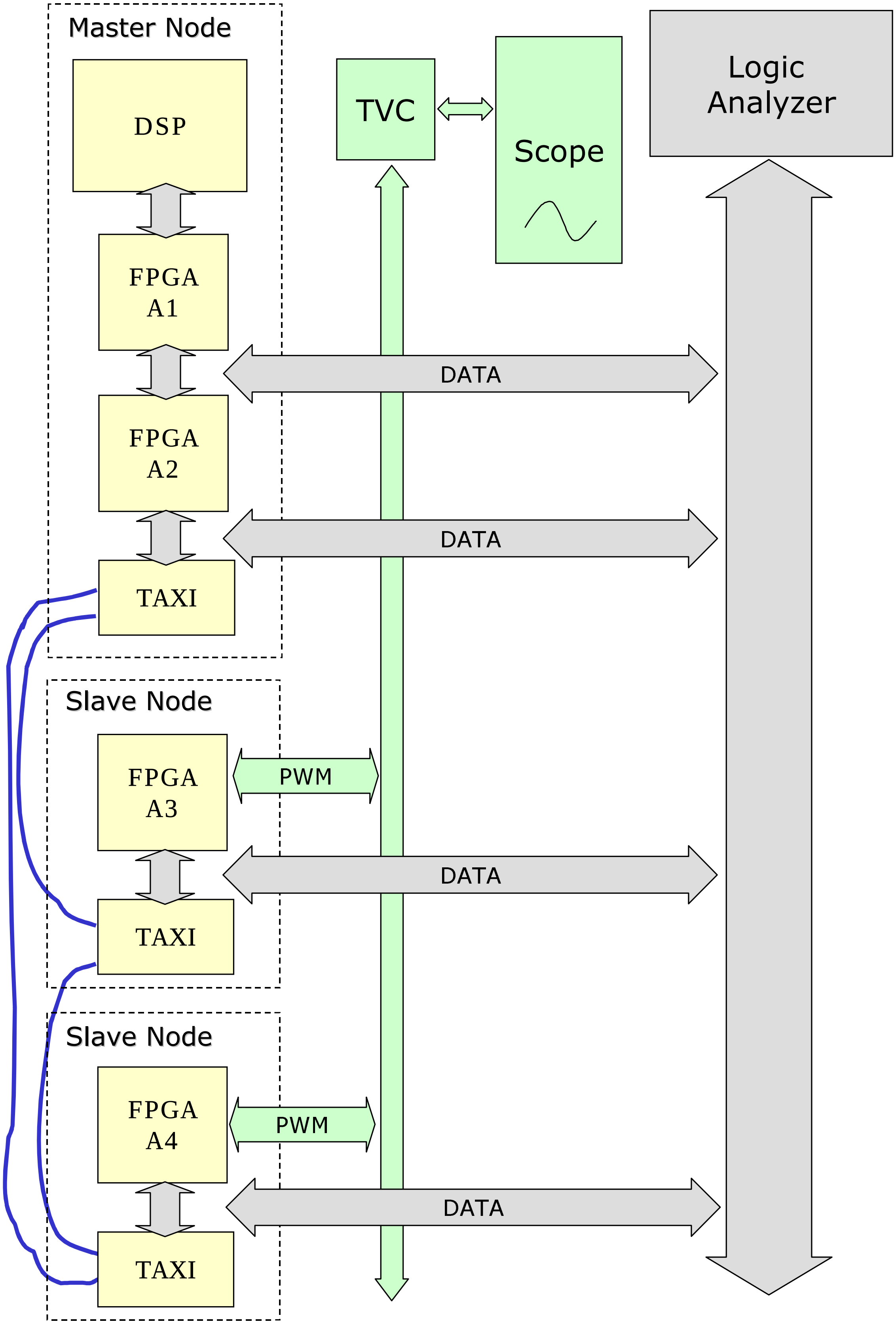


Figure 6.1 System test setup

The test setup has only two slave boards and hence cannot be used in a three-phase converter which has three PEBB modules. However, the DSP code and PWM generation were implemented as if all three slave nodes are present. It remains as a future work to integrate this system with a power stage. All of the VHDL and graphic design files used for implementation of the PES Net can be found in the Appendix.

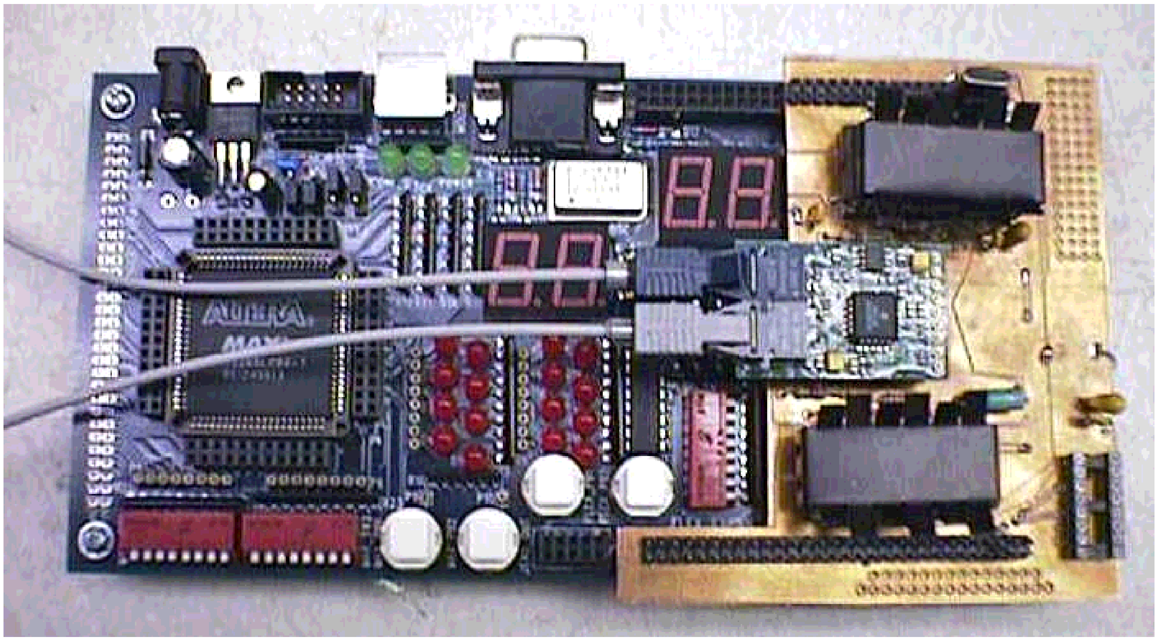


Figure 6.2 Test board

6.2 MEASUREMENT RESULTS

Line Signals

A sample of a data byte at the output of optical receiver is shown in Figure 6.3. Logical ‘1’ is represented by a transition in a signal, while ‘0’ is represented by the absence of transition. The data sent is “0110 0110.” This is encoded with 4B/5B coding in TAXIchip transmitter as “01110

01110”, and supplied to optical transmitter. After the signal propagates through the fiber, the optical receiver converts it from light into its electrical equivalent. The ten-bit sequence is observed.

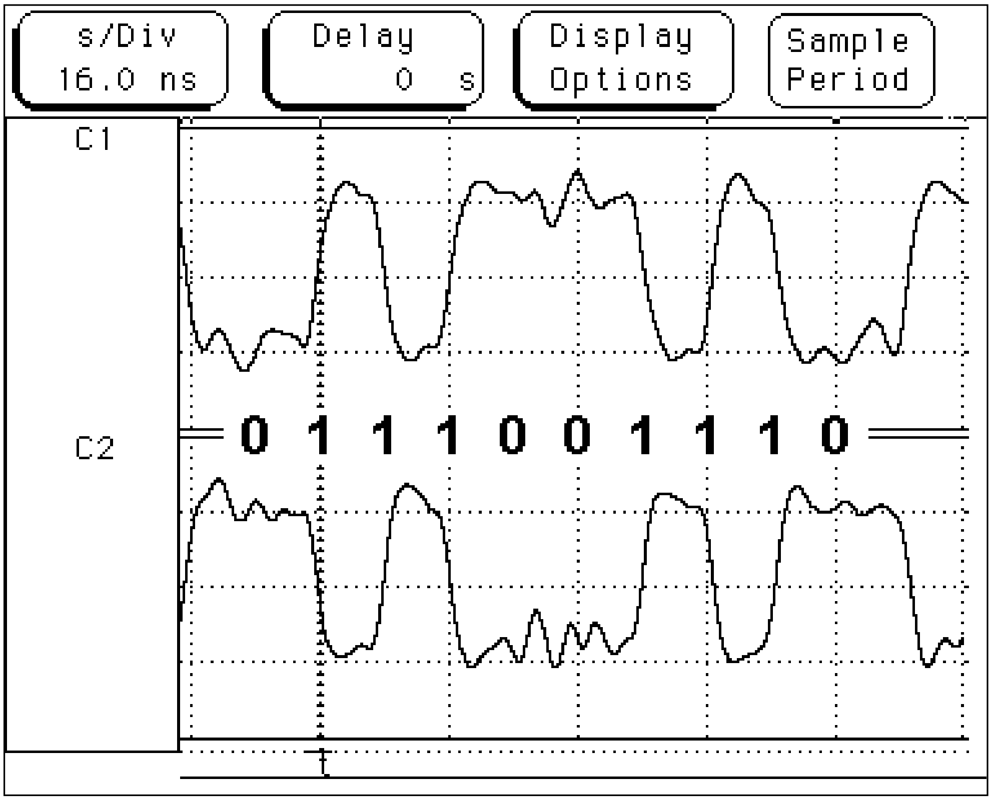


Figure 6.3 Pseudo-ECL data sequence sent at 125 Mb/s

Propagation Delay Measurement

One board was programmed to act as a test transmitter. The other two were programmed as slave nodes. This setup is shown in Figure 6.4.

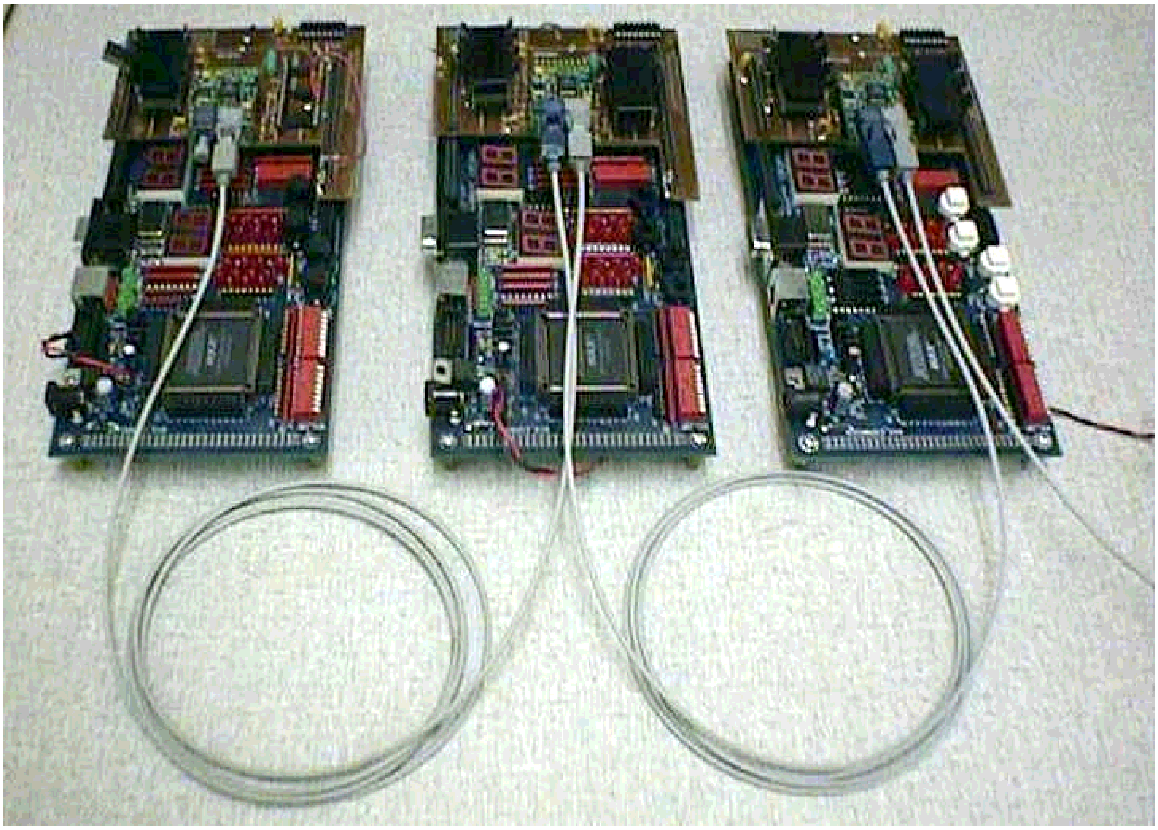


Figure 6.4 System test setup: test transmitter is implemented in the first board to the left; the other two boards act as slave nodes.

A sequence of eight bytes is transmitted. After data becomes valid on a FPGA-TAXIchip data bus, a strobe (STRB) command is issued. The TAXIchip generates acknowledge signal (ACK) in response. The transmitter sends the data to the next node, which substitutes it with its own data.

In Figure 6.5, we can see that the time from the first strobe command of the first node (STRB1) to the first strobe command of the second node (STRB2) is 468 ns. This is, in fact, a propagation delay through a fully operational slave node. Its duration is inversely proportional to the communication speed.

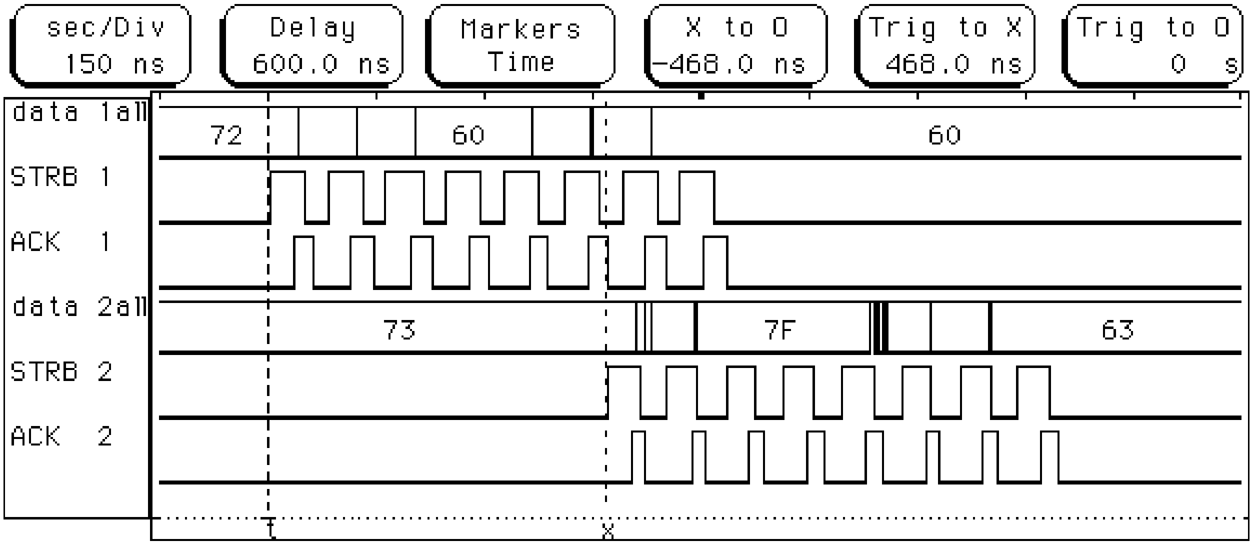


Figure 6.5 Measurement of slave node propagation delay

Switching Cycle and PWM Measurements

For this test, the test transmitter was substituted with Analog Devices SHARC ADSP-21062 (master node), Figure 6.6. The DSP interfaces with a FPGA device that is located on a board plugged into the SHARC Development board (A1). This is used to interface with one of the test boards (A2), programmed to perform master node communication functions. In this phase, the master node consists of a DSP and two FPGAs, because the FPGA on A1 has insufficient capacity. Two other boards (A3 and A4) were programmed as fully operational slave nodes.

The DSP was programmed to perform open-loop calculations based on space vector modulation for a three-phase inverter. The calculated data (duty cycles converted into ‘t_{up}’ and ‘t_{down}’) are written into A1 FPGA registers. From there, data are transmitted automatically to the ring.

Once in every switching cycle an interrupt occurs. This interrupt is used for sending a Synchronization identifier. An appropriate register (SYNC) in A1 is set to high, and after it has been read by A2, it is set to the low (RegCLR), Figure 6.7.

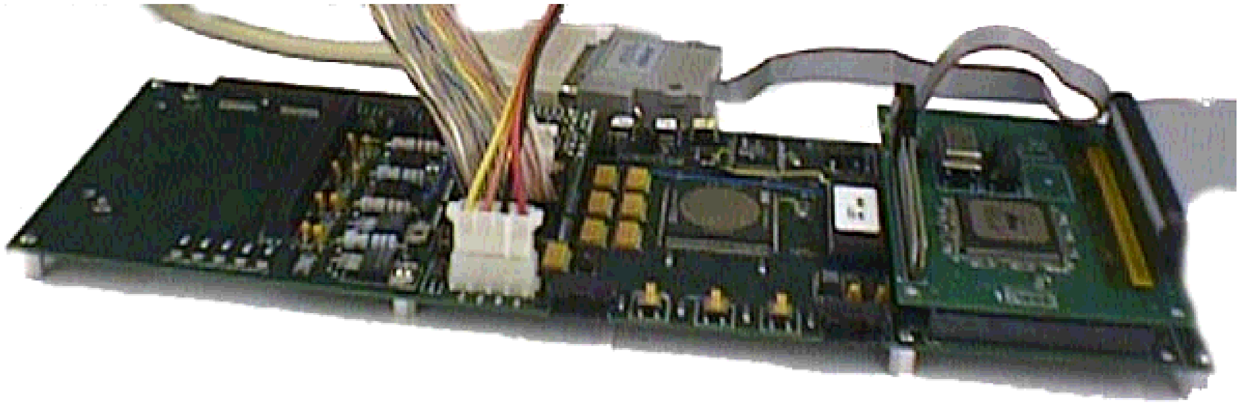
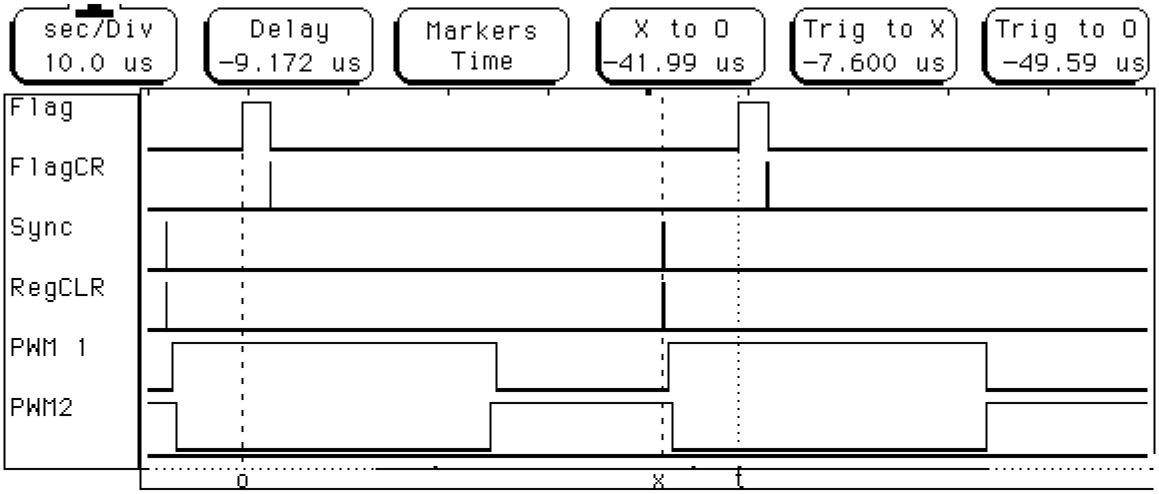


Figure 6.6 SHARC ADSP 21062 emulator board with Altera board

The DSP uses the same interrupt as a time-reference for the start of calculation. As soon as it is finished, the DSP writes the PWM information for each of the nodes into the A1, and sets register Flag to high. This register is set to low, after A2 sends all of the data frames to the TAXIchip transmitter.

The Logic Analyzer was triggered on Flag pin. The time distance from trigger (in Figure 6.7 it is marked by ‘t’) to the preceeding ‘SYNC’ (marker ‘X’) is the time that the DSP uses to perform the calculations, i.e. T_{DSP}. In this case, that time is 7.6 μs.

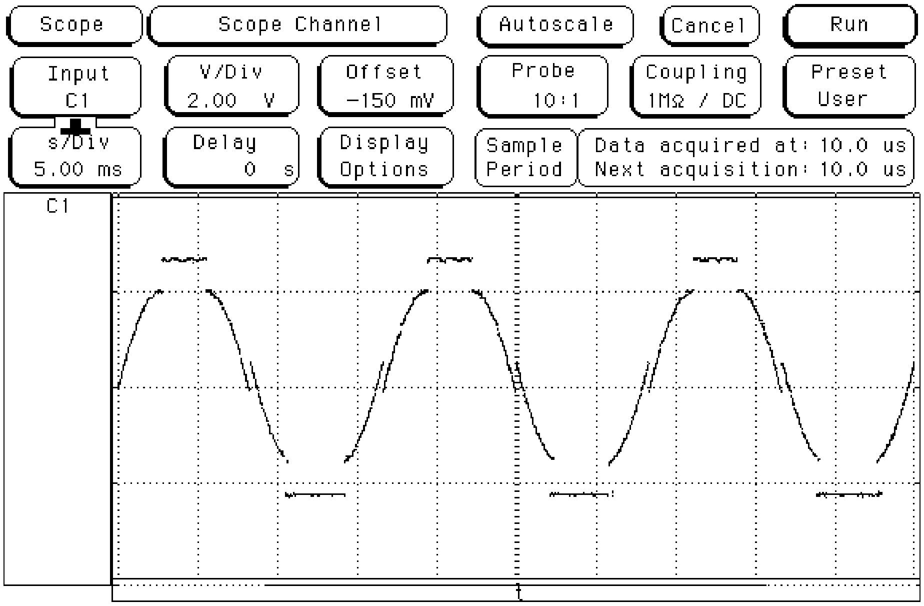


sec/Div 10.0 us
Delay -9.172 us
Markers Time
X to O -41.99 us
Trig to X -7.600 us
Trig to O -49.59 us

Figure 6.7 Measurement of the DSP calculation time

The time between two ‘Flag’ pulses is the switching period. In Figure 6.7, it is the time between ‘0’ and ‘t’ markers, and it equals 49.5 μs.

Information that is sent by the master node is the information for generation of PWM pulses. Based on this information, slave nodes generate PWM signals. For the testing purpose, the PWM signal from one of the nodes, was lead to the TVC, Figure 6.8. When the measured waveform is compared with the simulated waveform, obtained using the DSP Emulator (Figure 6.9), it is obvious that they are the same.



Input C1
V/Div 2.00 V
Offset −150 mV
Probe 10:1
Coupling 1MΩ / DC
Preset User
s/Div 5.00 ms
Delay 0 s
Data acquired at: 10.0 us
Next acquisition: 10.0 us

Figure 6.8 Measured low frequency content of the PWM sequence received from the master node. Open loop SVM control algorithm for a three-phase inverter is used.

The demodulated waveform and its originating PWM signal are shown in Figure 6.10, during three line cycles. Figures 6.11a and 6.11b show the same waveforms with increased time-resolution so that changes in the PWM sequence can be observed in more detail.

The PWM signals generated by two adjacent nodes are shown in Figure 6.12. It can be observed that these signals are shifted by 1/3 of the line cycle. Consequently, the resulting sinusoidal signals are shifted by 120°.

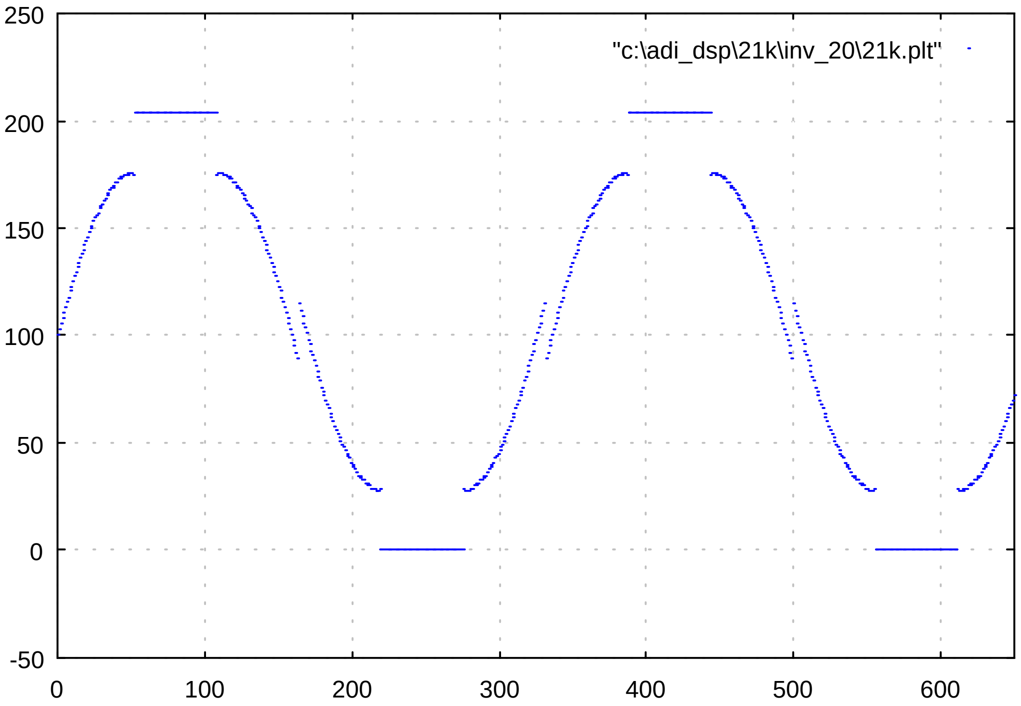


Figure 6.9 Modulating signal used by DSP to produce PWM sequence.

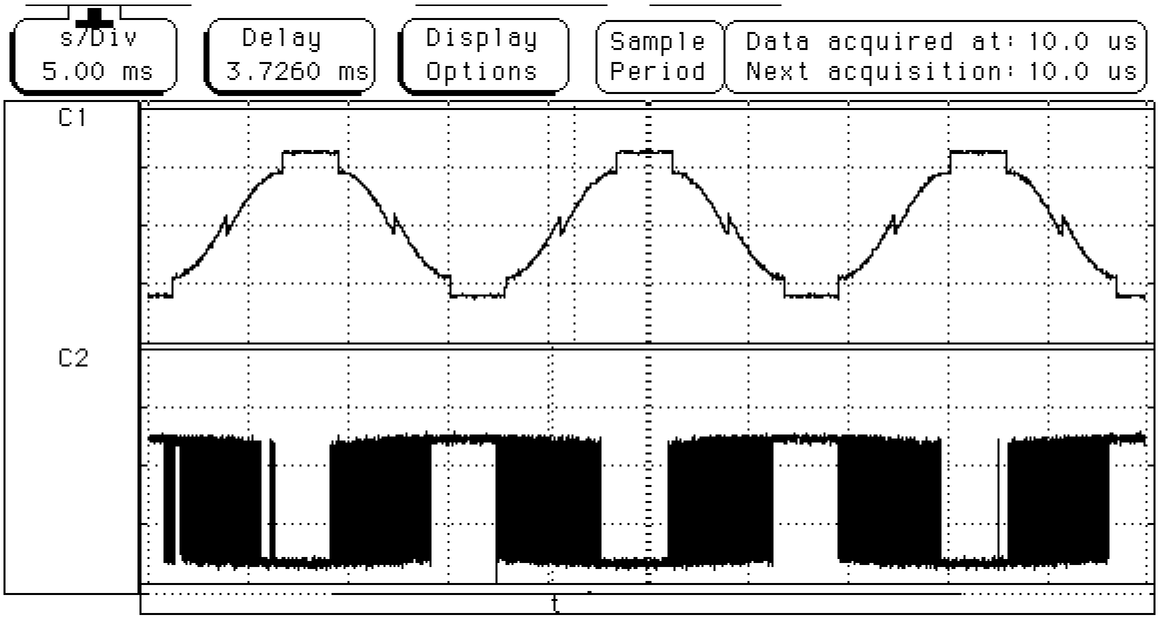
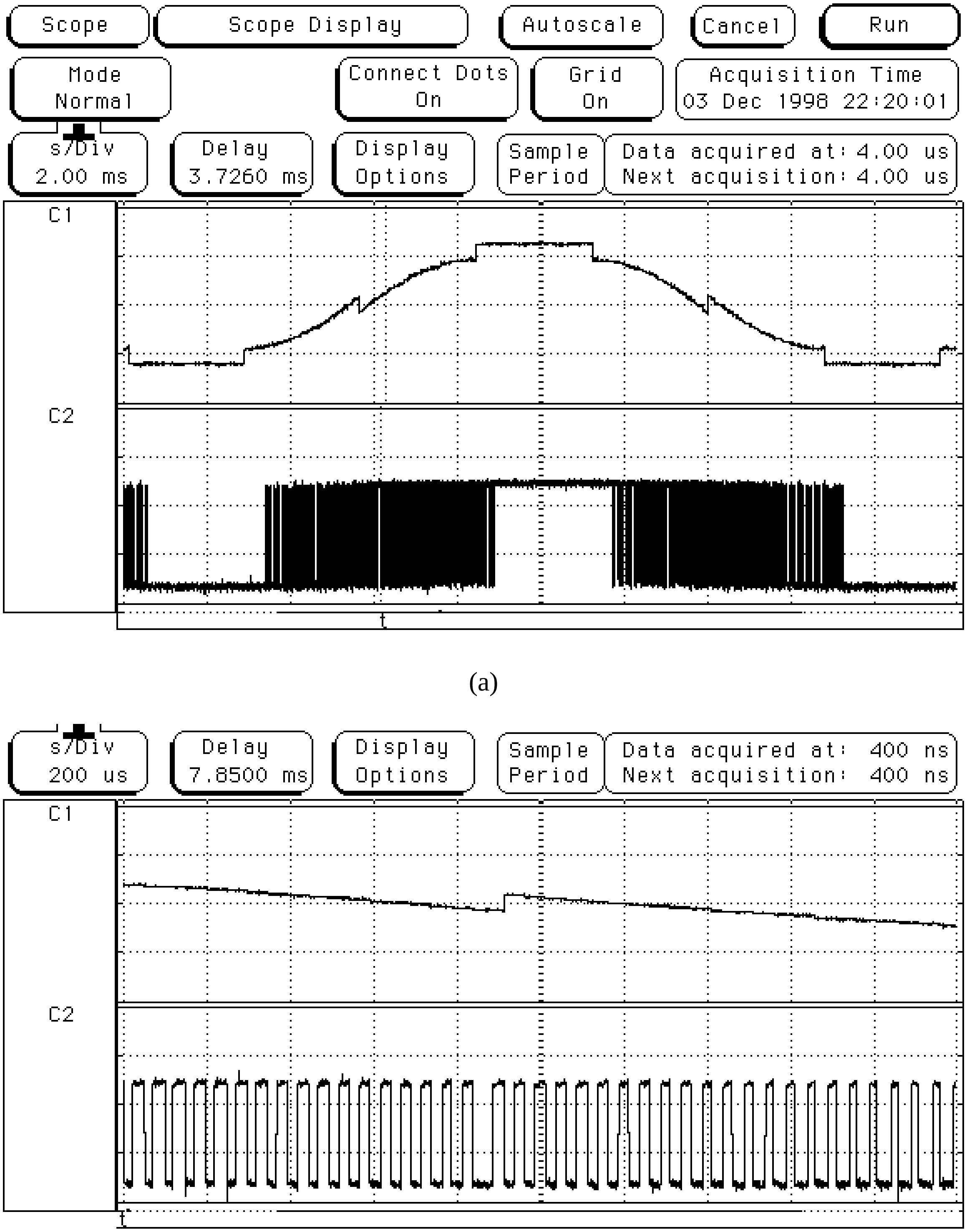


Figure 6.10 Demodulated (C1) and originating PWM signal (C2).



Mode Normal
Connect Dots On
Grid On
Acquisition Time 03 Dec 1998 22:20:01
s/Div 2.00 ms
Delay 3.7260 ms
Data acquired at: 4.00 us
Next acquisition: 4.00 us

(a)

s/Div 200 us
Delay 7.8500 ms
Data acquired at: 400 ns
Next acquisition: 400 ns

(b)

Figure 6.11 Demodulated (C1) and originating PWM signal (C2) with increased time resolution.

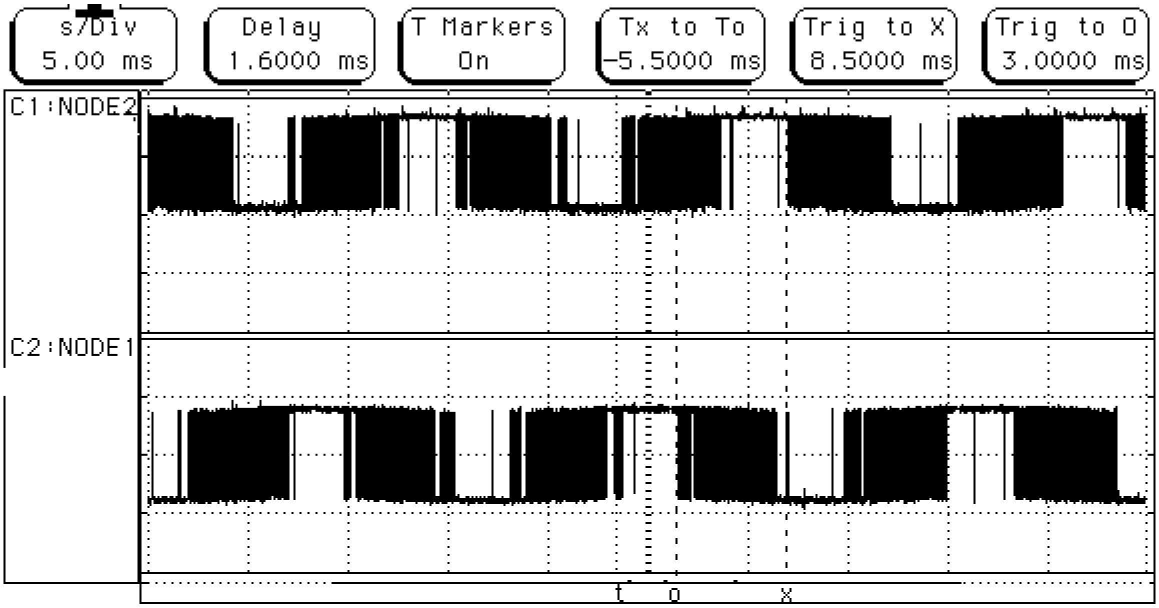


Figure 6.12 The PWM signals received by adjacent phase legs.

6.3 IMPLEMENTATION PROBLEMS

Some of the hardware problems that were encountered are listed below.

- TAXIchip™ sometimes needs a reset after a power up. Otherwise, it does not generate correct ECL signals. This is sometimes manifested in a way that generates ten-times slower TAXI synchronization commands.

- Due to parasitic capacitance, board crystal, Raltron 12.288 MHz 18pF 97JN10, had to be connected to different size capacitor, than specified in [28]. Sometimes, the crystal pin needs to be touched physically so that it will stop oscillating on its third harmonic, instead at its fundamental.

- If a pin of a FPGA is connected to some other hardware, it has to be defined in Altera software. It is wrong to assume that FPGA pin that is not defined is in the state of high impedance; it is not. If it is connected, it has to be defined, or otherwise it can affect behavior of both FPGA and connected hardware.

- The proto-board made on a PCB Board maker has poor contacts. Many short-circuits were detected.
- TAXIchip™ and FPGA consume significant power. Care should be taken to provide sufficient current from a power supply.
- The TVC generates the low-frequency information based on a PWM signal, by measuring a delay between two pulses. The used SVM algorithm produces a segmented PWM sequence, in which for certain longer time periods a switch is turned off. This delay is much higher than the delay between PWM pulses in other segments, so it causes the saturation of TVC. To prevent this, and in order to obtain a correct waveform, the PWM sequence was modified. During the time period when the switch is inactive, a periodic signal that prevents the saturation (one bit long pulse per switching period) is added to the PWM sequence. The pulses in this signal are short enough, so that they do not modify a desired output.

Chapter 7 CONCLUSION

7.1 PROTOCOL SCOPE

So, what should be specified as a protocol and how much flexibility should be left to the user of such system? If the protocol becomes too specific, it would narrow its user base. Since this system is targeting power electronics applications, it should be as general as possible for different power electronics topologies, but it does not have to satisfy requirement for other applications.

Probably the protocol scope would be to define the communication system from the moment when information is written into the master send-registers, to the moment when the information is written to slave PWM registers. In this way it would be left to the engineer to utilize sent information as he or she feels suited.

7.2 ANALYSIS OF PES NET IN TERMS OF PROPAGATION DELAY

In the PES Net system, the propagation delay between two adjacent nodes is slightly below 500 ns. The largest portion of the delay is caused by the TAXIchip™ [28], which is used to provide parallel/serial conversion and data coding/decoding at every node automatically. The availability of the commercial chip significantly simplified the hardware, although faster (custom designed) solutions could be implemented. With this delay, the maximum switching frequency for a three-phase converter can be calculated from (4.1) as 40 kHz.

The calculation of maximum converter switching frequency can also be performed based on data throughput within one communication cycle. Based on equation 5.7 for a three-phase system (n_n=3) and assuming 125 Mb/s bit rate, the DSP calculation time of 7.6 μs (code dependent), and a 4 byte synchronization sequence (n_{sd} = 3), the maximum switching frequency then can be estimated as 85 kHz.

The maximum switching frequency in the designed daisy-chain communication architecture is more constrained by the problem of delayed synchronization pulse than by the limited channel capacity. The use of proposed synchronization sequence pushes the allowed converter bandwidth higher.

A similar effort was made recently to design a communication for a power converter system described in [6], but with a different solution. In the implementation presented there, the hardware and protocol are designed in such way that the propagation delay is minimized. It is just one bit per node, but with the slower speed of communication (2.5 Mb/s), so the delay is $\tau_d = 400\text{ns}$. According to (5.1), this limits the switching frequency of a three-phase converter to around 40 kHz. However, the maximum switching frequency is considerably lower if the capacity of the communication channel is considered. The data packets have only 12 data bits and two command bits, which together with formatting require 10 μs to transmit at the 2.5 Mb/s rate. For a three-phase converter, at least three frames containing PWM data and one frame for synchronization need to be sent in every switching cycle. Because four packages have to be transmitted to three nodes, one frame is received with latency. The time needed for this is five times the time needed to send one frame, i.e. 50 μs. Assuming that the PWM data is updated every switching cycle, the maximum switching frequency is then 1/50 μs = 20 kHz. Therefore, for the system in [6], the problem of delayed synchronization pulse is not as constraining as the small capacity of a channel.

Structure [6] uses a less general design, has a smaller channel capacity and less propagation delay. Its propagation delay can be reduced even further by an increase in communicational speed. The PES Net has better data throughput capabilities per cycle, and it can support higher switching frequencies. When the proposed synchronization sequence is used, the effective propagation delay can be reduced by more than 95%, which makes it smaller than in the network described in [6].

7.3 ENGINEER – PES NET INTERFACE

As a final product, communication interface should be programmed permanently in an FPGA or programmable logic device (PLD) device. Once the board is powered up, it is network's responsibility to initialize and negotiate operating parameters.

Change of the communication crystal affects the transmission speed, so it is easy to lower it, if desired. The same crystal should supply the clock signal to the PWM generator (PWM clock). When the DSP control code is designed, this should be taken into account. The converter will operate at some submultiple of the crystal frequency. User chooses the crystal, based on the desired converter switching frequency and time (counter) resolution.

The PWM counter has to be able to count for a longer time than the converter switching period. Otherwise, the switch operation would not be defined through the whole period. The PWM generator uses the crystal first fundamental as an input clock signal. The generator counts the clock periods (T_{clk}) and uses the received t_{up} and t_{down}, to produce the switch control signals. The maximum communication, and therefor switching, period, T_{sw}, with a counter resolution (n_c) would be:

$$T_{sw} = 2^{n_c} * T_{clk} \qquad (7.1)$$

For example: if a PWM crystal has first fundamental at 25 MHz, one clock period lasts 40 ns. Assuming that t_{up}, t_{down} and counter have 8 bit resolution, based on (7.1) the maximum communication, and therefore switching, period would be 10.24 μs. This means that the highest switching frequency with 8 bits resolution preserved is 97.66 kHz.

With the same PWM crystal if the 16-bit resolution is used, maximum period is 2.62 ms, so the highest switching frequency with all 16 bits fully utilized is 381.45 Hz. This would defeat the purpose of having high-speed communication, because only 381 Hz switching frequency can be supported. In this case, a higher-frequency oscillator should be used to preserve the resolution.

In terms of synchronization sequence, data and timer resolution again need to be considered. There is no point in trying to achieve an excellent synchronization if the PWM counter resolution will annul the effort. At communication speed of 125 Mb/s, the 468 ns delay is measured. The cumulative propagation delay from the first node in a ring to the third is two times this. With synchronization sequence used, this delay would go from 2*468 ns to maximum 2*40 ns (assuming that the 5-byte delay is implemented). But if the PWM counter clock period is greater than 80 ns (12.5 MHz crystal), the synchronization of 80 ns still would not be achieved.

7.4 MACRO AND PES NET

The PES Net design is based on the MACRO protocol. However, some changes had to be made. MACRO is a commercial protocol that targets different applications. It has some general features, which are not needed in this, power electronics, system. The differences between the two are summarized below.

- MACRO closely follows the FDDI physical link layer. Therefore its physical link is defined as a glass multimode fiber, while PES Net will use plastic optical fiber.

- In MARCO application layer, data format is defined differently. The PES Net definition of data bytes does not fit into MACRO's definition for I/O devices, nor in any other.

- In the MACRO network, nodes are synchronized on a pre-specified data frame package. In the PES Net, a novel synchronization scheme is used.

- Faults and reporting methods are not the same. MACRO has following faults defined: byte violation error, frame checksum error, frame underflow error, frame overflow error, and ring break detection. PES Net errors are listed in Chapter 5. A ring break in MACRO definition is based on repeated reception of the receivers' 'VLTN' signal. The PES Net defines a ring break as a no-light output of optical fiber, which is caused when the fiber is damaged physically so no light is received.

- MACRO uses only one of the commands defined in a data layer, which marks the beginning of the data package. PES Net also utilizes commands for synchronization, master-shutdown, master-reset and initialization. MACRO issues those commands by setting appropriate flag bits in a data-package. This results in slower node and network response since the individual frames need to be sent to the each of the nodes (unless a broadcast address is used).

- The differences in the two protocols are not insurmountable. If compromises are made, the PES Net could 'fit' in the MACRO protocol. The most important difference is the synchronization method, which could be implemented in a MACRO as follows:

  - Node waits for beginning of the data frame (CMD 1)

  - Node checks the address following the CMD; if it is not a pre-specified address, it waits for the next frame. If it is, then:

  - The furthest node (number n) on a communication link synchronizes on reception of the address field (second word in a frame)

  - The node before that one (number n-1), synchronizes on 2 words + 4 words of delay = sixth word

  - Node number (n-2) synchronizes on 6 words + 4 words of delay = 10th word

Since the frame is only 12 bytes long, only three nodes could be synchronized using this method and without any changes to the MACRO protocol.

7.5 POSSIBLE IMPROVEMENTS

The capability of individual addressing of the nodes and their increased intelligence allows us to explore further improvements in this system. If this is fully utilized, PEBB modules could be reconfigured ‘on-line’. There could be a ‘silent’ node in a system, which is activated while the system is running. Implementation of broadcast transmission is also possible.

When this kind of system is commercialized and becomes standard for PEBB communication, a space should be allocated in a data frame for a pre-assigned PEBB ID number. This number would characterize the device. When reported to the master node, it would give information about its current, voltage and temperature ratings, manufacturer and other similar types of information.

Furthermore, a desirable feature might be the FPGA programming directly through the communication link during the initialization process. An initial code would be stored in boot PROM, but some parts of the code could change while the system is running. The benefit would be that a smaller capacity of FPGA could be used.

For the PES Net, even higher switching frequencies can be achieved if the protocol is modified. This protocol specifies CRC-16 use. Actually, CRC-8 or a parity check would be enough for this system, since there is also additional protective coding - 4B/5B. Data frames may be shorter. Actually, for a three-phase converter in a regular mode of operation, only two data bytes are used per phase leg. The PWM signal for the second switch can be generated based on the same information. However, there are converters with more than two switches per phase leg, so some generality would be lost if the data frame were shortened.

A topic that may be worth investigating is an implementation of adaptive differential PCM (ADPCM) or adaptive delta modulation (ADM). The communication cycle equals the converter switching cycle, and the switching cycle is of much higher frequency than the line frequency. Since the variables that are measured, such as current and voltages, change with line frequency, their change is slow compared to the communication cycle. Therefore it may be beneficial to implement predictive delta modulation, where the information sent to the DSP would be the information on the measured variable change. The advantage of this is that a smaller number of bits would have to be transmitted for the same resolution, and lower resolution A/D converters could be used.

7.6 SUMMARY

In this work, communication issues in power converter systems were investigated. A new modularized control communication network for this type of system is presented. This type of modularization is in agreement with the PEBB principle. The proposed topology reduces wiring, brings down installation cost, makes error detection easier and increases reliability. The PEBB modules designed with the presented modularized topology are more generic and have more capabilities.

The network design is based on MACRO motion control protocol. In the PES Net, one communication cycle corresponds to one switching cycle of the converter. The maximum communication cycle depends on the number of nodes in the communication system. The proposed control network can support a converter with three phase legs switching up to 100 kHz.

The data transfer is over optical fiber, at the speed of 125 Mb/s. This high transmission communication speed provides fast retrieval of sensed information. The number of lines is reduced greatly since more data is pushed through the lines. The fiber capacity is utilized fully using this topology.

Error detection on reception and local intelligence that decides on the implementation of received data improves the reliability of the system. The transmission medium is plastic fiber optic cable that provides EMI noise immunity.

The newly proposed modular topology for power converters has many benefits, but it also introduces new problems. When the communication of a converter has a ring structure, a propagation delay is created at points of reception. This problem has been investigated here. A unique method of synchronization is presented that also can be applied to motion control networks.

REFERENCES

[1] Hiti, S., *Modeling and Control of Three-Phase PWM Converter*, Ph.D. Dissertation, Blacksburg: Virginia Power Electronics Center, Virginia Tech, July 1995.

[2] Mohan, N., Undeland, T. M., Robbins, W. P., *Power Electronics Converters, Applications, and Design*, New York: John Wiley & Sons, 1989.

[3] Vlatkovic, V. and Borojevic, D., “Digital-Signal-Processor-Based Control of Three-Phase Space Vector Modulated Converters,” *IEEE Transactions on Industrial Electronics,* Vol. 41, No. 3, pp. 326-332, June 1994.

[4] Jacobina, C. B., Lima, A. M. N., da Silva, E. R. C., “PWM Space Vector Based in Digital Scalar Modulation,” *IEEE PESC’97 Proceedings*, June 1997.

[5] Jung, S. L., Huang H. S., Chang, M. Y., Tzou, Y.Y., “DSP Based Multiple-Loop Control Strategy for Single-Phase Inverters Used in AC Power Sources," *IEEE PESC’97 Proceedings,* June 1997.

[6] Du Toit, J. A., Le Roux, A. D., Enslin, J. H. R., “An Integrated Controller Module for Distributed Control of Power Electronics,” *IEEE APEC’98 Proceedings,* pp. 874-880, February 1998.

[7] Milosavljevic, I., Borojevic, D., “Modularized Control Architecture for Power Converters,” *16th Annual VPEC Seminar Proceedings*, pp. 85-92, September 1998.

[8] Xing, K., Lin, R., Lee, F. C., and Borojevic, D., “Some Issues Related to Power Electronics Building Blocks,” *14th Annual VPEC Seminar Proceedings*, pp. T1-T7, September 1996.

[9] Ericsen, T., Tucker, T., Hamilton, D., Campisi, G., Whitcomb, C., Borraccini, J., Jacobsen, W., “Standardized Power Switch System Modules (Power Electronics Building Blocks),” *Power Systems World ’97*, http://pebb.onr.navy.mil/techpapr.nsf, September 1997.

[10] Dai, H., Xing, K., and Lee, F. C., “Investigation of Soft-Switching Techniques for Power Electronics Building Blocks (PEBB),” *IEEE APEC’98 Proceedings*, pp. 633-639, February 1998.

[11] Prasad, V. H., Dubovsky, S., Celanovic, N., Zhang, R., Borojevic, D., “DSP Based Implementation of a Power Electronics Control System,” *15th Annual VPEC Seminar Proceedings*, pp. 61-67, September 1997.

[12] Zhang, R., Boroyevich, D., Prasad, V. H., Mao, H., Lee, F. C., Dubovsky, S., “A Three-Phase Inverter with a Neutral Leg with Space Vector Modulation”, *IEEE APEC‘97 Proceedings*, pp. 857-863, February 1997.

[13] Amy, J. V. Jr, “Integrated Power Systems (IPS)”, Briefing for ONR PEBB Program Review, http://pebb.onr.navy.mil/techpapr.nsf/, April 1998.

[14] Leonard, R. E., Dade, T. B, “The All Electrical Ship: Enabling Revolutionary Changes in Naval Warefare”, *The Submarine Review*, http://pebb.onr.navy.mil/techpapr.nsf/, October 1998.

[15] “Plastic Optical Fiber and HCS® Fiber Cable and Connectors for Versatile Link,” *Technical Data 5963-3711E*, Hewlett Packard, November 1994.

[16] "Fiber Optic Cable," *Catalog 124827*, AMP Optical Interconnection Systems Group, p. 5, March 1996.

[17] Bertsekas, D., Gallager, R., *Data Networks*, Second Edition, New York: Prentice Hall, Inc, 1992.

[18] Holton, C., Claus, R. O., Duncan, P., Milosavljevic, I., "Power Electronics Building Blocks (PEBB) Utilizing Fiber Optics", Presentation for FEORC Annual Seminar, Virginia Tech, April 1998.

[19] *Frequently Asked Questions about LonWorks Networks*, http://www.lonworks.echelon.com, Echelon - The LonWork Company, 1997.

[20] *The History of Fieldbus*, Fieldbus Tutorial-History, http://rolf.ece.curtin.edu.au/-clive/Fieldbus, 1997.

[21] *Selecting the Right Fieldbus*, http://www.gespac.com/html, Gespac, 1997.

[22] *SERCOS (IEC 1491)*, Developer's Kit, SERCOS N.A., March 1997.

[23] Berardinis, L., "SERCOS Lights the Way for Digital Drives," *Machine Design*, August 1994.

[24] Motion and Control Ring Optical, Specification, http://www.macro.org, Delta-Tau Data Systems, Inc., May 1998.

[25] R. Blomseth, W. Capolongo, B. Dolin, J. Lund. "The LonWorks Networks Services (LNS) Architecture Technical Overview", http://www.lonworks.echelon.com, 1997.

[26] Pieder Joerg, "Integrated Power Converter at ABB", April '98 ONR PEBB Program Review, http://pebb.onr.navy.mil/APR98qpr.nsf, 1998.

[27] Milosavljevic, I., Ye, Z., Borojevic, D., Holton, C., "Analysis of Converter Operation with a Phase-Leg in Daisy–Chained or Ring Type Structure", *IEEE PESC'99 Proceedings*, June 1999.

[28] "TAXIchip™ Integrated Circuits," *Data Sheet and Technical Manual, Publication#07370*, Rev. F, Advanced Micro Devices, April 1994.

[29] "Fiber-Optic Solutions for 125 MBd Data Communication Application at Copper Wire Prices," *Application Note 1066*, Hewlett Packard, 1995.

[30] Gajski, D., D., Vahid, F., Narayan, S., Gong, J., *Specification and Design of Embedded Systems*, New Jersey: Prentice Hall, Inc., 1994.

[31] Armstrong, J., R., Gray, F., G., *Structured Logic Design with VHDL*, New Jersey: Prentice Hall PTR, 1993.

[32] "Max+Plus® II Getting Started," *P25-04803-03*, Version 8.1, Altera Corporation, September 1997.

[33] "University Program Design Laboratory Package", *User Guide A-UG-UP1-01*, Version 1, Altera Corporation, August 1997.

[34] "Altera Data Book," *A-DB-0696-01*, Altera Corporation, June 1996.

VITA

The author, Ivana Milosavljevic, was born in Belgrade, Yugoslavia. She received the Bachelor of Science degree in electrical engineering from the University of Belgrade, Yugoslavia in December 1995. For her diploma thesis, she did a study of human motion control. In September 1996 she joined Virginia Power Electronics Center (VPEC) at Virginia Tech. She worked on a joint project between VPEC and Fiber Electro-Optic Research Center investigating communication issues in power electronics. The results of this work led to filing of an invention disclosure for the Power Electronics System Network, PES Net, in 1998. She received her Master of Science degree in electrical engineering in January 1999.

Ivana Milosavljević