USING MULTICORE MICROPROCESSOR IN DESIGNING GENERIC ACQUISITION SYSTEM FOR OIL FIELD DOWN HOLE LOGGING TOOLS

By

Mohammed Mahmoud Elsagher Farrag

A Thesis submitted to the

Faculty of Engineering at Cairo University

In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In

Electronics and Electrical Communications Engineering

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
March 2012

USING MULTICORE MICROPROCESSOR IN DESIGNING GENERIC ACQUISITION SYSTEM FOR OIL FIELD DOWN HOLE LOGGING TOOLS

By

Mohammed Mahmoud Elsagher Farrag

A Thesis submitted to the

Faculty of Engineering at Cairo University

In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In

Electronics and Electrical Communications Engineering

Under the Supervision of

Prof. Dr. Amin M. Nassar

Professor
Electronics and Communications Department

Dr. Hossam A. Fahmy

Assistant Professor
Electronics and Communications Department

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
March 2012

USING MULTICORE MICROPROCESSOR IN DESIGNING GENERIC ACQUISITION SYSTEM FOR OIL FIELD DOWN HOLE LOGGING TOOLS

By

Mohammed Mahmoud Elsagher

A Thesis Submitted to the

Faculty of Engineering at Cairo University

In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In

Electronics and Electrical Communications Engineering

Approved by the

Examining Committee

____.
Prof. Dr. Elsayed Mostafa Saad , Member.

____.
Dr. Ibrahim Kamar , Member.

____.
Prof. Dr. Amin M. Nassar , Thesis Main Advisor.

____.
Dr. Hossam A Fahmy , Thesis Advisor.

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
March 2012

ACKNOWLEDGEMENTS

I would like to express my great gratitude to my supervisors, Prof. Dr. Amin Nassar, for his understanding guidance and for keeping things easy for me, he used to turn any obstacle into a solvable matter, things that I can't find any equivalent words to express how valuable they were to me, and to Dr. Hossam Fahmy for keeping track of all my movements and giving me his full attention and his brilliant guidance, he is always a source of motivations and positive energy.

Also I would like to thank Mr. Adolfo Recio- Huston USA, for the valuable resources he sent me. And Mr. Nalin Weerasinghe-Kabushiki Kaisha, Japan, for helping me getting the correct material and contacting the proper experts in Schlumberger product and research center - Kabushiki Kaisha Japan.

Finally, I would like to thank my Family for their support and understanding.

ABSTRACT

Oil Industry depends on utilizing various techniques to get the valuable hydrocarbons content of the ground. In order to get this done in a profitable matter, a critical information mining processes are held, starting on surface, through seismic explorations , then through logging the dig space seeking data that would lead to the proper judgment on the very near next step in well life cycle , or to update formation models for future plans. The outcome of the logged data is valuable enough to sacrifice the yet valuable time in gathering these data. This gathering process could last for long days. The purpose of this study is to provide a solution for dealing with the, yet still happen, failures of electronic sections during the logging job, speed up the operation by using hardware capable of achieving the required processing in less times, which increases maximum allowed logging speed, And to reduce logging tools' maintenance cost by proposing a modular, robust design, that could have a way out of a long downtime loss in case of an unexpected electronic failure in Downhole logging tools during a job.

The logging process starts from producing Electrical responses of used sensors, digitize them, process the acquired data depending on the measured physical quantity and how it is measured, and convey these processed data, on a communication link, up to the surface. Advanced Processors keep inspiring people with many potential benefits that could be obtained through the high computation energy they could provide. Multi-core Microprocessors, and parallel processing, offer highly-capable processing power to applications starting from server stations down to embedded systems. To do more than one task at a time speeds up performance, allows more complex applications to be feasible. In this study, other aspects of Multi-cores and Multi-processors are taken, which are Reliability, service availability, and compatibility. The Application introduced here is hungry for high reliability implementations that could live and "amuse" working in hostile environments, such as oil wells. The aim of the following chapter is to provide a brief view of what that field of industry may concern of, and to introduce wireline logging applications, which is the target application here.

The first chapter will illustrate why does wireline logging has to depend on reliable processing infrastructure that has to live in environments such as Downhole a well that either being drilled, engineered, or producing. Chapter one describes as well the problem

proposed here and its severity on the industry. Chapter two describes the boundaries of the application, and what a suitable solution should offer. Following chapters will describe the proposed solution in terms of the stated design Objectives, and then an Implementation of the proposed solution will be presented. The last chapter summaries the whole topic, discusses conclusions acquired, and possible future work.

INDEX OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

One of the major steps in searching for, and extracting oil and gas at field reservoirs is to collect Petrophysical data that describes the geologic Formation around a well. This data is used as an input to help determine the presence and quantity of hydrocarbons in a reservoir, and to estimate the potential profit of the next step taken in the well's life cycle.

Started by Conrad and Marcel Schlumberger, in 1926, France, the first log was produced. It was a simple Resistivity log shows recorded resistivity measurements of the surrounding Formation (Ground Layers) against depth, inside a well. The resistivity log is used to evaluate the possibility and amount of hydrocarbons content in the Formation. Also to determine how mud had invaded the side walls of the well under test, this is known as invasion profile. This simple application was achieved by a Mechanical device that carries current and voltage Probes Downhole. It used to measure the amount of injected current, versus the voltage drop, measured at certain depth. That simple device was named, by Marcel Schlumberger, as "Sonde", a French word, that is now so well known in this industry, to call sensors or mechanical parts in a Downhole logging tool.

As the need for more information increased, the tools used to acquire data evolved. Downhole tools are now, not only to collect data, but also to take samples of fluids and rocks, and to present imaging of the formation or the well itself. These applications required more capable hardware and firmware that are able to capture, analyze, process, and communicate with surface. As the environment is tough, and faults that may lead to wasting time, or sometimes to close a well, are so costly, more processing and control power are needed.

1.1 Well Logging

After this many Formation properties were measured, like Porosity, Density, and Lithology. Other properties a well in Open-hole section (the section which is still not covered with Metal Casing) like Invasion profile, Mud Resistivity. More services came into the sight, like Borehole Imaging, and Dip-measurements. Other services are required in Cased-hole, or the cased part of the well. The cased part is the part of the well covered with metal casing. Services in the cased part are like, Cement-bond evaluation, which evaluate the quality of Cement behind the casing, also Cement Formation bond, Micro-annulus evaluation, Casing

Corrosion, and Casing thickness. Back to the Open-hole section, another different type of service are provided, which are Reservoir Sampling, and Seismic. Reservoir Sampling is used to capture and Analyze Fluid samples from the Formation, these analyses are made Downhole, also to evaluate formation pressure at certain stations (Measuring Points). Seismic is used to evaluate formation layers by getting the impulse response of Formation layers, which is used in Explorations. Other auxiliary measurements are usually taken, like Temperature, pressure, Borehole wash outs, head tension/compression, which measures the mechanical tension and compression on the tool string sent Downhole. That is used to help safely driving the journey Downhole, and to detect any stuck conditions. After all, the main measurement obtained is Depth. In general, measurements made by logging are physical characteristics of the well fluids and or Formation.

1.1.1 Logging Computations

The Computations are the algorithms used to combine outputs from various Tools (Processed data acquired by tools' sensors) in a way to produce meaningful, electronically calibrated (against Electronic Circuits' errors due to the variation of temperature), Borehole corrected (by suppressing the effect of mud), and then calibrated. The acquired data is sent up hole, on surface to another computation station, where certain Petrophysical model parameters are applied on the acquired data.

1.1.2 The Log

The Log is data surveys acquired by Downhole tools. To log Physical Properties against depth, as well as other geometrical properties of the Borehole, Open, or cased against depth, is a "Depth Log". The Depth Log could be against either Measured Depth or true vertical depth, in cases of deviated wells, the measured depth is not the same as the true vertical depth, due to the deviation angle of the well. In addition to Depth Log, there is also Time Log (Station Log), which is acquiring measurements verses time, where measurements are taken at certain depth points, or where fluid samples are taken. Figure 1-1 is an example of the final Depth Log that shows Gamma Ray (GAPI), Bit Size (supposed hole diameter), and Caliber Readings in the first track (linear scale), Induction Resistivity in the second track (log scale), Density, and Neutron Porosity in the last track.

1.1.3 Log Interpretation

Interpretation is the translation of log curves into meaningful information. Correlation between various types of measurements is used to get quick and sufficient information about the logged interval; the relation between Density, Porosity, and resistivity could be a direct indication of the presence and roughly the amount of hydrocarbons or other elements. Not only the physical measurements are logged, but also many other auxiliary measurements like borehole radius, head tension, and Spontaneous potential are recorded versus depth as well. These measurements are important to verify the physical measurements taken, for example as washouts would be a problem for Neutron Porosity logging tools, so Neutron porosity won't be valid at that depth of deep washouts. Another type of data is also logged, which is Log Quality Control or LQC logs. It is to verify the integrity of the measurements, and the proper condition of the Hardware.

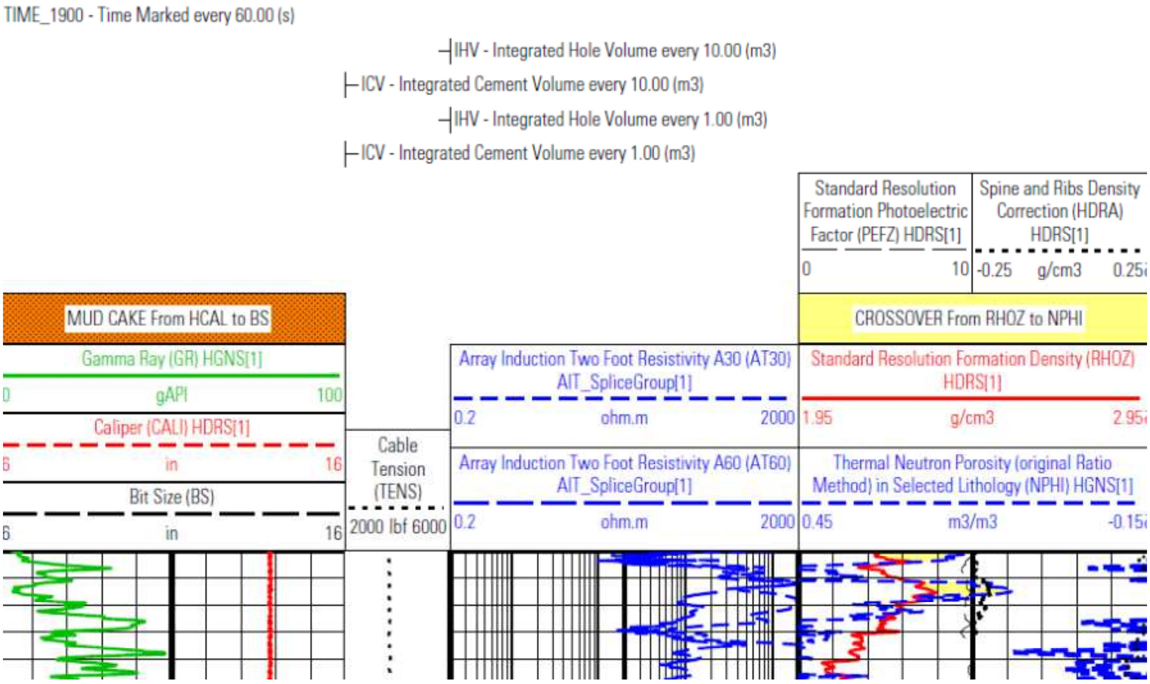


Figure 1-1 Resistivity, Density, and Neutron Porosity depth log

1.1.4 Inversion Model

Geophysical remote sensing data can be used to help solve practical environmental, engineering or exploration problems. In some cases, when only limited knowledge about the subsurface is required, inferences drawn directly from the data can be sufficient, however, when more detailed information about the subsurface is needed, quantitative models of the earth need to be estimated. This is geophysical inversion. In a typical geophysical survey, or what is called in Wireline Logging “A Log”, Energy is put into the ground in form of Electrical, Electromagnetic field, Mechanical Sonic, or Ultrasonic, even Magnetic resonance Energies and record a response via Logging tools. The Acquired Log is simply the observed data. The values of the data depend upon the distribution of physical properties in the subsurface. This is the goal of the inverse problem is to determine the distribution of the physical property or properties that gave rise to the data. Inversion Model is a generic Item used in various fields of science. In Geophysics, Inversion Models are used to get more detailed picture of the logged interval. The acquired data, measured by logging tools, is fed to the model; The Model applies numerical iterations on the Model parameters in order to get the measured data as close as possible to the values predicted by the model. When Iteration/Verification process is completed, the Inversion Model is now capable of giving much information about the logged interval, as well as an expressive simulation.

1.2 Wireline Data Acquisition

The logging task is done by oil field services companies named generally as “Wireline”. It is not a brand name; it is general name of a collection of services that are provided during a time in which another specialized, crew, and equipment takes over the well site, executing what is known in this industry as “Wireline Logging Job”.

1.2.1 Downhole Equipment

Logging Equipment are the is the logging tools used downhole to get the measurements needed , or do the task required during a logging job. In a single tool string, there are always various tools that produce different kinds of physical measurements, which together are used to get meaningful, interpretable log. Figure 1-2 shows the tool string used to produce the Log Shown in Figure 1-1. Starting from the bottom, the first tool from the bottom measures resistivity based on induction theory, it is used with oil based mud drilled wells, as it works in nonconductive environment. The second tool from the bottom measures density using a radioactive source, placed in the tool before running in hole. It also measures Lithology and Form Factor [4]. In addition to that it measures Borehole radius using a moving caliber, which is open when the tool starts logging up, and Micro-resistivity [5], which refers to resistivity measurements in a shallow depth of investigation. The third tool from the bottom measures Neutron Porosity using a neutron producing radioactive source, and Formation’s Natural Gamma rays. The fourth tool from the bottom is the Telemetry Cartridge which is responsible for modulation/demodulation and bus-mastering all downhole tools below. The last thing is the Logging head which interfaces the telemetry cartridge with the logging cable.

1.2.2 Logging Cable

Logging Cable is the way to convey Power and Data Downhole as well as to provide the Mechanical connection with the tool string. Heptacable is a cable with seven conductors and armor. It is the cable used for running most open-hole tools,. Each telemetry system uses different conductor assignments for both data and power transmission. Other Types of cables like coaxial cable and mono-cable (single Conductor) are used, but mostly in perforations applications. The cable itself is used in measuring depth and tension via Surface Equipment which interpret cable tension and speed into numeric values used to produce the log as well as to control speed during logging.

Figure 1-2 GR - Density –Litho logy, and Induction Resistivity Tool String

1.2.3 Well Sections and Conveyance

There are three main types of logging jobs based on the well environment, they are:

- **Open-Hole Logging**: when the logging section is not cased, Just the ground or formation, with its walls covered with drilling Mud.
- **Cased-Hole Logging**: is the case when the logging section is within a Casing. Cased-hole log is used to check casing conditions and some properties of the formation that could be checked in cased sections.
- **Production Logging**: is used to perform some well tasks, or to acquire a log while the well is producing. Pressure equipment is used so often in this kind of logging. It is done by slim tools that could run inside the tubing.

To drive downhole tool string inside a well, if the well is vertical, tools' weight could be enough to pull them down the well. In horizontal wells, or wells with high deviation angles, tools' weight is not sufficient, other conveyance tools are used [7], just as a Tractor. The Tractor is a conveyance device, used to tract tool strings inside deviated wells it can run in Open or cased holes, but it works manly in cased or production logging as there where it has proven its best performance. Figure 1-3 shows a tractor driving logging tools in and openhole section.

Figure 1-3 View of a Tractor in Open Hole

TLC is another way of conveyance. In which drilling pipes are used to drive the logging tool string all the way to the logging section, up and down. This way is so widely used, as it is a rather simple way, but it consumes valuable time. In depths around 2000 meters, it might take about 12 hours to put the string down in a TLC job, and another 12 hours to take it up again. Figure 1-4 shows a view of a TLC job.

1.2.4 Surface Equipments

Surface Equipment is the Hardware starting from cable drum, depth-tension gear, and the log processing station. Surface Equipment is able to provide the necessary data that make it possible to drive the tool string in or out of the well, even if no power is supplied to the downhole string.

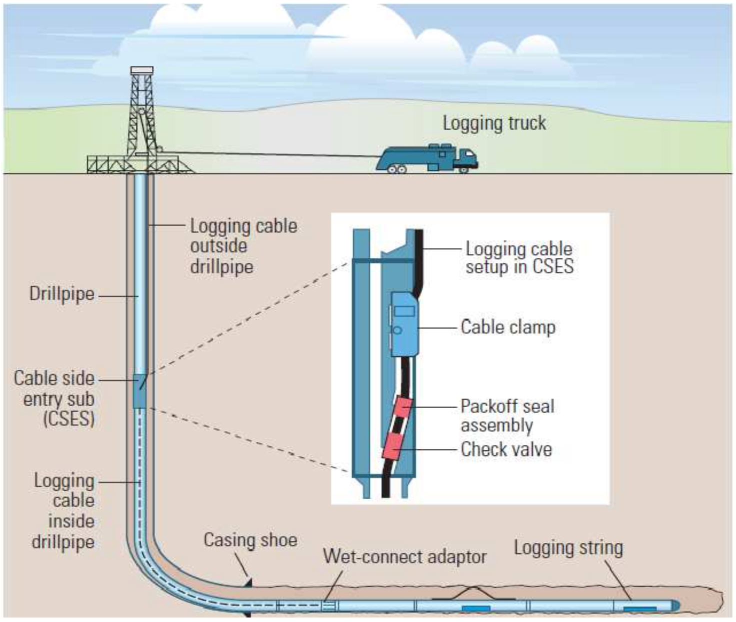


Figure 1-4 TLC Job

Depth-tension data is necessary to avoid stuck condition, in which the tool string gets pulled to well’s wall due to static pressure, or a well collapse around the tool. Depth-tension devices make it possible to know how much tension to safely put on the cable.

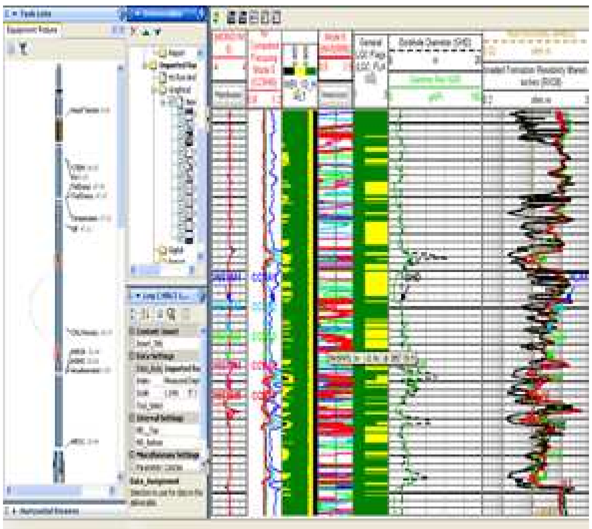


Figure 1- 5 Acquisition Software runs on surface processing unit

1.3 Performance and Reliability of Downhole Electronics.

1.3.1 Well Environment

Well Environment is one of the most hostile environments an electronic hardware can survive. The temperature ranges from -30 C at the surface to around 250 C at some parts of the Globe. This is the most common logging temperature range. Logging tools are supposed to sustain bumping caused by mechanical lifting, or inside the well itself, against well walls, or casing liners. This mechanical shocks can break the tool itself, or electronic boards inside. The vibrations can cause bad solder jointed to appear quickly, or disconnect electrical joints with even minor defects.

1.3.2 Natural Formation Pressure

Natural formation Pressure in most of common depths can reach up to 30 KPSI. The electronic boards have to be fixed inside metal protector shield.

1.3.3 H2S

H2S is found in production wells, during drilling, or during perforation, in some wells, it has a corrosive effect on steel, which houses the electronic cartridge, the primary problem, it causes, is metal embitterment.

1.3.4 Humidity

As the housing of the electronic cartridge is designed to, should, block firmly fluids in and out of the cartridge, the air trapped inside contains certain amount of water vapor, in downhole condition with high temperatures and pressures, the trapped water vapor and other gases cause severe corrosion to the printed circuit boards, in a way that could cause electronic failures either with the PCB or ICs. That is why humidity should be extracted out of electronic cartridges; this is done by many simple means, like adding humidity-absorbing chemicals.

The severity of a downhole failure is classified in terms of money, loss time, and injury. Time loss represents the nonproductive time. If a tool failed downhole and the conveyance was a wireline in a good vertical well, the time to take out the tool string, troubleshoot and to replace the failed tool with a backup, at, for example, 1000 feet depth would be, in today's terms, 2 hours at least. Which costs a direct money loss equals to rig rate at that downtime. Some rigs has a daily rates around 50,000 $/day, others cost more, some Rigs/drilling ships apply $ per second, for example 5 $ per second. So 2 hours would be 36,000 dollars. Another delayed loss that well owner will face, will be how many barrels/hour the well will produce times $ per Barrel, that will be much more loss. If the conveyance was by TLC , for a 1000 feet , it would take the normal rig to get pull out of hole , and then to run in hole again , in around 9 hours. That is added to troubleshooting and fixing times. i-e a small solder joint at one lead of the telemetry chip or an amplifier chip, could cause a catastrophic incident even with the backup tool is available. Table 1-1 shows a typical severity matrix used today in one of the major service companies. The table shows that a loss time of 48 hours is as severe as losing a life. The logging industry over years had overcome these conditions but still challenging the overheating and pressure effects on electronic cartridges of various logging tools, as most of the tools are composed of an electronic cartridge with or without a Sonde /Sensors parts.

EXTERNAL LOSS CATEGORY	Owner + Service Money Loss (K$)	Days Lost/Rest	Oil Spill (Liters)	Non Productive Time (Hrs)
Light	< 10	< 1	< 100	< 4
Serious	> 10 and < 100	> 1 and < 100	> 100 and < 1000	> 4 and < 24
Major	> 100 and < 1000	> 100	> 1000 and < 10,000	> 24 and < 48
Catastrophic	> 1000 and < 10,000	Loss of Life (fatality)	> 10,000 and <100,00	> 48 and < 72
Multi-Catastrophic	> 10,000	Multiple fatality	> 100,000	> 72

Table 1-1 Severity Matrix used in Field Services Companies

1.4 Multi-processing and Multi-core Microprocessors in Embedded Systems

Multi-core processor consists of multiple central processing units, residing in physical package. Increased Integration has been a hallmark of processor evolution for many years. Similar to how different types of functionality have been integrated onto a single processor core, the sheer availability of transistors has made it feasible to package together several processor cores resulting in the development of multi-core processors. The move to Multicore processor architectures in the embedded processor industry has been spurred by other trends occurring over the past few years as well, including:

- Diminishing return from instruction level parallelism enhancing technique – the micro architectural techniques employed to extract parallelism from one instruction stream are becoming more expensive relative to the performance gain[17].
- Clock scaling reaching limits due to power constraints – an empirical study [23] suggests that 1% clock speed increase results in a 3% power increase. Space, power, cooling techniques are key constrains in oil industry.

A system with multiple processor cores has been available for years. It differs from multiprocessor (more than one processor, with the same of different architectures and memory space) systems in number of characteristics such as:

- Communication latency: The communication latency of a multi-core is typically lower than a multiprocessor. Bandwidth between cores in multi-core is also typically higher than in a multiprocessor. The reason is the proximity of the processor cores. Consider the processor cores.
- Number of processors: in a typical multiprocessor system, number of processing units , here is a full processor and one unit , is much more than the maximum number or cores , which is the processing unit in multi-core architecture.

There are other advantages to integrating cores onto a single device. Multi-processor systems are by no means new, but until now discrete cores shared memory using memory/cache coherency schemes that relied on cores' snooping one another's memory/cache accesses over an external bus. If the snooping core detected an access to a memory block for which it had an updated version in its cache, it would have to write the updated block to main shared memory before the requesting core could retrieve a valid version. So as well as increasing the memory bandwidth and latency, the integration of dual cores allows inter-core buses 3 to 4 times faster than external buses, making snooping far less of a strain on the inter-core buses than a discrete processor solution would be[24].

Embedded systems are where microprocessors are used to do Application Specific tasks, constrained by time scheduling of executed tasks, maximum dissipated power, and reliability.

Embedded applications obligate the hardware, and here, a microprocessor, to face environmental conditions that the host application works in. they are also obligated the running operating system to meet certain hard, or soft real time deadlines.

Both homogeneous and heterogeneous multiprocessor architectures have been used in building embedded systems. Heterogeneous multiprocessing is used when there are parts of the embedded software that would need the power of a digital signal processor and other parts need a micro-controller for the housekeeping activity. With the growing complexity of embedded systems and the rapid improvements in process technology the development of systems-on-chip and of embedded systems increasingly is based on integration of multiple cores, either homogeneous (such as processors) or heterogeneous. Modern systems are increasingly utilizing a combination of processors (CPUs, MCUs, DSPs) which are programmed in software, reconfigurable hardware (FPGAs, PLDs), and custom application–specific hardware. It appears likely that the next generation of hardware will be increasingly programmable, blending processors and configurable hardware.

Here, an application is introduced , that relies on what can a multi-core processor do , in saving total area , eases communications between the processing team , and more important , to offer reliability through offering a redundant processing resources.

1.5 Research Objectives and Methodology

The objective of this research is to point out the importance and potential benefits of using advanced processing hardware to offer reliability in one of the applications, in which a reliable system is a necessity.

This work focuses on preparing the application, which is here, a wireline logging, to be ready to use a multi-core microprocessor, as a mean of performance enhancement, and portability. It doesn’t go into any architectural specifics of a microprocessor, instead, it studies the application it might b used in. It starts by describing the application, which is well logging, describes its needs, and major hazard it might face. By doing this the main problem is pointed out. The proposed solution is presented, which is mainly depend to reorganize the parts of the main application in a way to be able to host and get the benefit of multi-core microprocessor , in both reliability and design modularity which are the main targets of the proposed solution. Parts of the proposed solution is being synthesized and tested, as hardware, in an FPGA, that work as one of the Main peripherals , and in software as part of processing tasks, which is cable telemetry, one of the main duties that the used Multicore processor has to carry on. Finally the unhandled sides of the proposed solution, and points for some potential enhancements, are presented in future work section.

1.6 Summary

- Petro physics started with Conrad Schlumberger in 1920s, depending on the primitive electrical measurements of the Formation in determining the existence of hydrocarbons. Other techniques are introduced to measure various properties of the Formation, Casing, and production Flow as well.
- Logging Tools are The Hardware sent to the well in order to gather these data.
- Telemetry system is the most common used to convey the acquired data to surface processing system via logging cable.
- The purpose of this study is to redesign the electronic parts of wireline Downhole tool string using Multicore processors in a way to improve Reliability and overall performance, maintenance and upgrade of these tools.

CHAPTER 2

PETROPHYSICS AND TELEMETRY BACKGROUND

This chapter discusses some facts about some of the physical properties that logging tools measure. They are not even a significant portion of all services and measurements that could be provided in logging job, but they are common to nearly all of them. The purpose of this chapter is to take the presented measurements types as the “Sonde” section measurements referred to in the next chapters.

2.2.1 Petrophysics and Well Engineering

As pointed to in chapter 1, Petrophysics is the science in which physical and geological information about a spot of earth is analyzed in order to calculate the nature, amount and quality of trapped hydrocarbons. Upon petrophysical analysis, well engineering process calculate the benefit of the trapped hydrocarbons and the best way to schedule, reach and extract them. Next sections present some basic measurements taken by logging tools that are necessary for petrophysical analysis.

2.1.1 Well Inclination

The purpose of an Inclinometer is to determine the orientation of the logging tool in earth’s reference system, from which the orientation of the wellbore can be deduced. An inclinometer measures angles that the tool axes has declined from the magnetic north, the relative bearing, and acceleration. By knowing, the date and well coordinates, the acquisition software transforms the magnetic north into the geographic north. The result is an indication of where on earth the measurements, of other tools in the tool string, have been taken. Inclinometry tools are always required in making an image of the formation around the well, and so it is an integrated part in all micro-resistivity imaging tools.

Physics of measurements

The tool purpose is to determine a reference frame, where X, Y, Z axes are:

X: Horizontal component of earth’s magnetic field.

Y: Magnetic east.

Z: The downward directed vertical.

Tool axes as seen by the tool are:

i: Tool Reference Axis.

j: Tool Perpendicular Reference Axis.

k: Tool Vertical Axis.

An Inclinometer expresses tool position in earth's system of axes

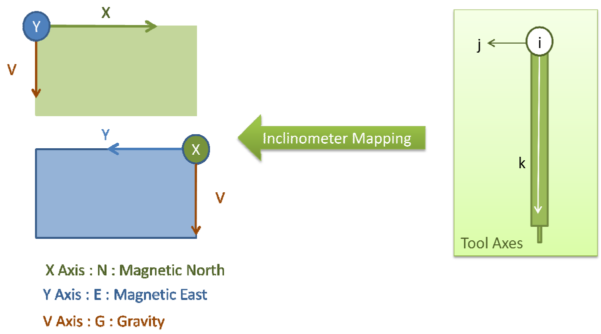


Figure 2-1 Inclinometer Measurements

Tool Orientation is defined by Deviation, Azimuth, and Relative Bearing. These results could be obtained by combining results of three-axis accelerometer with three axes magnetometer. An Inclinometer is simply three accelerometers; each is on one of the three axes. Obtained values are used in a mapping process between the two systems of coordinates. They are defined as follows:

Deviation is the angle between vertical and tool axis k, measured in the vertical plane (Z, k).

Azimuth is expressed as two components.

- AZR: Azimuth of reference line, which is the angle between North and reference line projection on the horizontal plane (N, E).
- AZX: Azimuth of well's deviation: angle between North and tool axis projection on the horizontal plane (N, E).

Relative Bearing is the rotation from the top-of-hole direction to pad 1(Reference line on tools' surface), in a clockwise direction looking down the tool axis. RB is given between 0 and 360 deg. It is not well defined when well's deviation is near zero.

The outputs of an Inclinometer are:

- Fx, Fy, Fz: Acceleration components.
- Gx, Gy, Gz: Magnetic field intensity components.

The calculations of Deviation, Azimuth, and Relative Bearing can be done as follows.

$$|\mathrm{F}| = \sqrt{\mathrm{Fx}^2 + \mathrm{Fy}^2 + \mathrm{Fz}^2} \qquad (2-1)$$

$$|\mathrm{G}| = \sqrt{\mathrm{Gx}^2 + \mathrm{Gy}^2 + \mathrm{Gz}^2} \qquad (2-2)$$

$$\mathrm{gx} = \frac{\mathrm{Gx}}{|\mathrm{Gx}|}, \qquad \mathrm{gy} = \frac{\mathrm{Gy}}{|\mathrm{Gy}|}, \qquad \mathrm{gz} = \frac{\mathrm{Gz}}{|\mathrm{Gz}|} \qquad (2-3)$$

$$\mathrm{fx} = \frac{\mathrm{Fx}}{|\mathrm{Fx}|}, \qquad \mathrm{fy} = \frac{\mathrm{Fy}}{|\mathrm{Fy}|}, \qquad \mathrm{fz} = \frac{\mathrm{Fz}}{|\mathrm{Fz}|} \qquad (2-4)$$

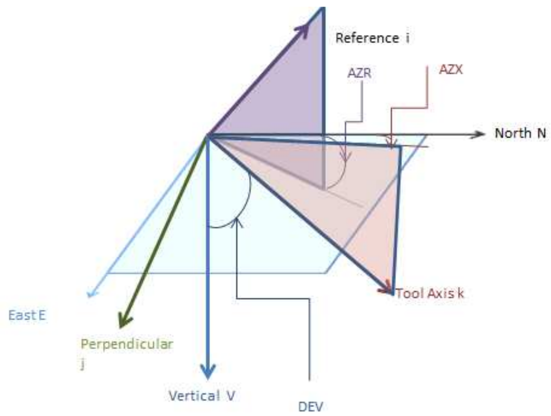


Figure 2-2 Tool and Earth's Axes

From figure 2-2, the relation between Angles and Axes could be evaluated as follows:

$$\mathrm{DEV} = \tan^{-1}\frac{\sqrt{\mathrm{gx}^2 + \mathrm{gy}^2}}{\mathrm{gz}} \qquad (2-5)$$

$$\mathrm{AZR} = \tan^{-1}\frac{\mathrm{fygx} - \mathrm{fxgy}}{\mathrm{fz} - \mathrm{gz}\sin \mathrm{I}} \qquad (2-6)$$

$$\mathrm{AZX} = \tan^{-1}\frac{\mathrm{fzgy} - \mathrm{fygz}}{\mathrm{fx} - \mathrm{gx}\sin \mathrm{I}} \qquad (2-7)$$

$$\sin \mathrm{I} = \mathrm{fxgx} + \mathrm{fygy} + \mathrm{fzgz} \qquad (2-8)$$

Sin I, Represents the inclination of the magnetic field with respect to the horizontal at this point of earth. That is the conversion between Magnetic Axes and geographical Axes, so that the final result would be with respect to the geographical coordinates used by the geologists.

The Relative Bearing would be:

$$\mathrm{RB} = \tan^{-1}\frac{\mathrm{gy}}{-\mathrm{gx}} \qquad (2-9)$$

The outputs of an Inclinometer, acceleration and magnetic field components, change with temperature, therefore, one temperature sensor is used with each of the accelerometer and the magnetometer used in an inclinometer. The recorded temperatures are used, with stored temperature coefficients, to compensate outputs with the current temperature.

Output signals of an Inclinometer

In one of the commercial inclinometer sensors, accelerometer signals have a frequency spectrum that extends from 0 to 200 Hz, with Voltage level up to 10 volts. Normal sensitivity for each inclinometer is 5 volts per g (nominal gravity of 9.81 m/s).

Magnetometer outputs Range from 0 to 10 volts. Magnetic field intensity components, the output of the three dimensions magnetometer, are in range of 8.4 v /Oersted. (The Maximum Earth Magnetic field is 0.7 Oersted).

2.1.2 Gamma Rays

Natural Gamma rays measurement is essential in almost any logging. Natural Gamma ray is used in correlation purposes as Gamma rays profile of certain section of the formation is as unique as a finger print.

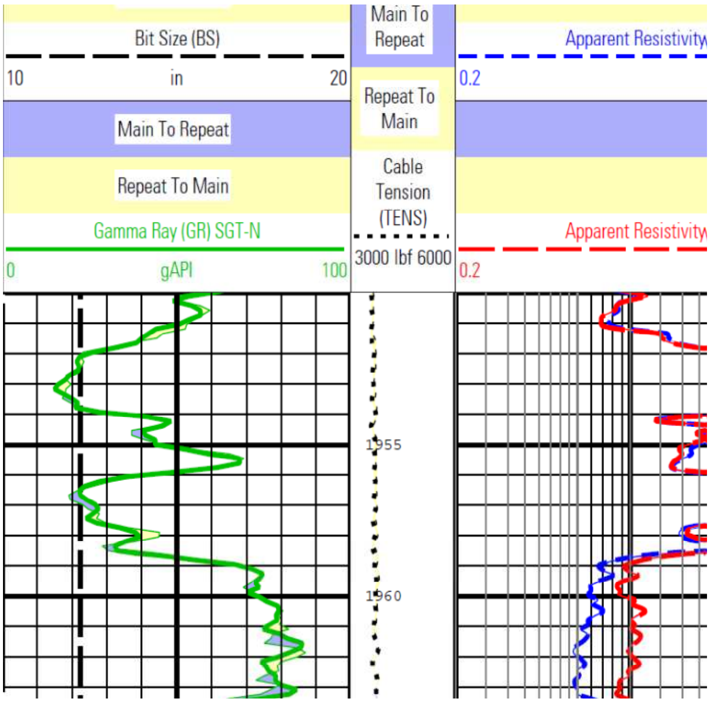


Figure 2-3 Natural Gamma Rays in Sonic – Resistivity log

The Natural Gamma rays are measured by a Scintillation detector [8]. This detector requires high voltage value to excite its “Photo Multiplier”. This high voltage changes with Detector’s aging, It is adjusted in a matter that slight (+/- 50 V) won’t change the detector response. This detector‘s characteristics, under high temperature, may deviate a little enough to be compensated by this high voltage adjustment in the middle of Detector “Plateau”.

Gamma rays emit photons from NaI detector’s crystal, they are, then, amplified by the Photomultiplier’s cathodes and represented in a form of negative pulse. Theses pulses are filtered and discriminated (compared to a threshold), then counted figure 2-4 shows a block diagram of a scintillation detector.

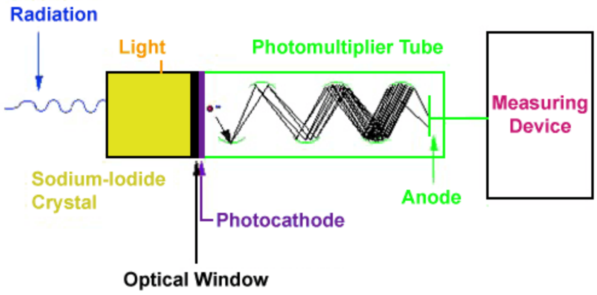


Figure 2-4 Gamma Rays Scintillation Detector

Output and input signals to a GR detector

The scintillation detector uses high voltage with the photomultiplier, a control command adjusts the output of the high voltage power supply. The counted GR is sent as a number every second. The processing unit does statistical calculations on the acquired counts/sec, and outputs the mean of received counts over a certain period of time, say 10 seconds. Detected Gamma rays are negative pulses. High voltage power supply reading is another output; 0 to 10 volts represent 0 to 3000 volts.

2.1.3 Micro-resistivity Imaging

Micro-resistivity Images are used to get formation dip, fractures, and faults [21]. The identification of fractures is based on the observation of resistivity contrast with the host rock. A common industry trend in designing resistivity images, is in the form an array of sensors over number of arms (pads) spaced in a way to cover most of the borehole space [5].

Figure 2-5: Micro-resistivity imager – Earth Imager – by Baker Atlas

Spaced pads sensors are mainly probes to sense the returned current through the formation, and then by applying a factor represents geometric shape equivalent to the part of formation involved in the measurements, which depends on the geometry of current probes, an estimation of resistivity is acquired. The acquired value of resistivity represents the resistivity of certain portions of the formation depending on the depth that the induced current could reach. In electrical imagers the acquired resistivity is meant to be Micro-resistivity, or the resistivity of a shallow layer of the formation.

The operating principle of Electrical imagers is that, a bedding surface cutting across a borehole at some angle causes a change in formation Micro-resistivity measured by the sensors. These sensors are installed on pads distributed all around the well circumference as mentioned before. The dip or slope of the bedding plane causes the pads to encounter Micro-resistivity changes all along the analyzed portion of the well. The difference in resistivity over depth through formation layers, or in places where faults or cracks, that contains a different material, exist, is used to calculate the dip.

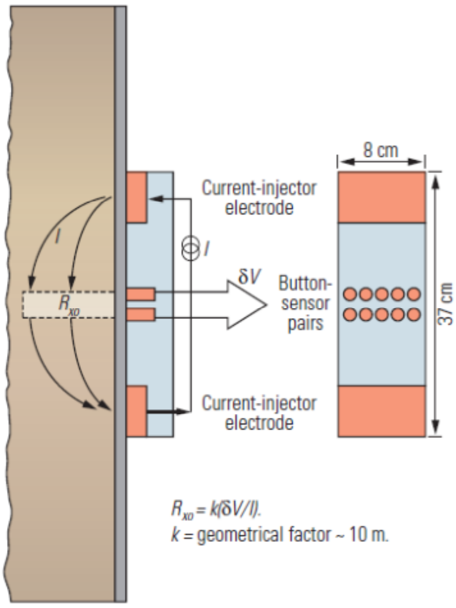


Figure 2- 6 Schematic diagram of a Micro-resistivity Imager – OBMI – by Schlumberger

Input control signals in a micro-resistivity Imager

According to the tool principle of operation, a current source is required to conduct the variable current injection into the formation in a controlled manner. A current value control command is sent from the processing unit, and a controller in the tool, and current value is monitored as a number per second, and sent back to the control commands issuer, which is either the processing unit on surface or a downhole controller.

The tool contains a caliber that carries tool pads, to open and close it, a command is sent from surface processing unit to a set of relays in sonde section to open a bath for the sent AC to operate a motor, or to fire a solenoid. An open / close code command word is needed. Also caliber reading is monitored and sent back to the processing unit on surface.

Output signals from a micro-resistivity Imager

Each pad has a group of sensors, 32, in a typical tool. All of them are multiplexed and added to a single frame per second. Four frames per second are sent to a control unit, either on surface or downhole, to apply DSP preprocessing on the acquired frames, then, by the processing unit, a snap shot at the measurement depth is put in the log.

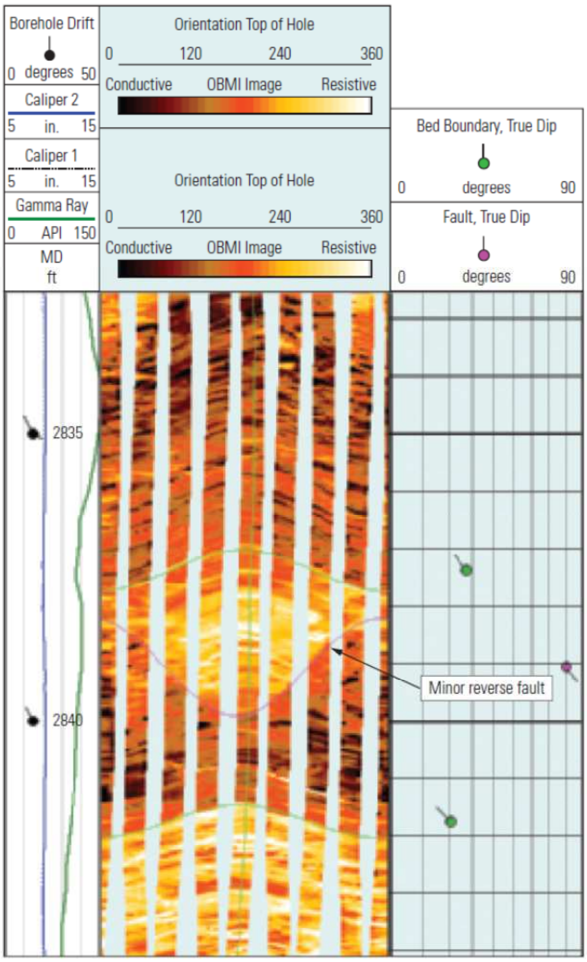


Figure 2-7 Micro-resistivity Log sample

2.1.4 Auxiliary measurements

Temperature Measurement, which could be done by simple RTD connected with current source and voltage terminals. To measure temperature is itself, target information, besides, it is used in further computations, as many sensors have to be temperature compensated. Temperature value can be sent up as a number per second.

Downhole Force Measurement, is an important quantity in logging runs , as it represents the amount of tension applied on the logging head , and so, it reflects many situations that a tool string could face downhole , for example , if a " stuck " condition happens during logging up , head tension will severely increase , if not managed, it will cut the head's weak point (the best scenario) . Also, compression force could damage the tool string, if a sever compression is applied on the string from surface, for example, in TLC jobs mentioned in chapter 1. DF is simply measured by a resistor bridge, fed by 7 Hz, 10 volts pulse. The bridge unbalance voltage value would reflect the applied force. This value is sent Uphole for processing.

2.2 Telemetry

Telemetry System is an old communications scheme that is widely used in many applications which involve a communication link on a relatively short distance, in our situation around 30,000 feet. Telemetry is “Historically” defined as the communication system that provides remote control over other party as it allows automatic monitoring, alerting, and record-keeping necessary for safe, efficient operations. One of the Current systems in use, in wireline logging industry, is commercially identified as EDTS, or Enhanced digital Telemetry System. EDTS is the system preferred in high bandwidth required applications such as imagers/scanners. EDTS implements Enhanced Fast Tool Bus, EFTB, as its main way of collecting data from various stations (tools) of a tool string. EFTB is carried by a shielded twisted pair cable. It is auto terminated. Data is capsulated as packets, which start from the very bottom tool, being relayed in every tool above, and then sent by the upper tool, which works as the telemetry cartridge, whose function is to re-capsulate collected packets in super packets, and send them via OFDM over cable, or what is known as DMT.

2.2.1 Cable Telemetry

A Logging cable is the media used to carry all power and communications between downhole and surface units. One of the famous types is a seven conductors’ cable, shielded, and armored. Cable band width is about 700 kHz. Table 2-1 shows the electrical characteristics of some cables used today in wireline logging.

Code	Outer Diameter	weight	Ends fixed break strength	Safe Working Load	Max temp 8 hours	DC resistance (Ohm/kft)	Insulation resistance (M Ohm.kft) >	Capacitance Center conductor pF/ft	Voltage Ratings (Volts)	Current Rating (A)
1-23ZA	0.233	103	5890	2945	450	7.1	15000	40	920	1.61
1-25P	0.257	118	6530	3265	265	10.4	15000	32	1030	1.1
1-25ZA XXS	0.257	118	8390	4195	450	10.4	15000	32	1025	
1-32ZA XS	0.319	195	11620	5810	450	2.8	15000	48	1200	
2-32ZA	0.319	196	9990	4995	450	6.8	15000	44	825	
7-39P XXS	0.395	247	15440	7720	265	10.9	15000	49	720	
7-39ZA XXS	0.395	258	15440	7720	365	10.9	15000	45	720	1.1
7-46P	0.464	332	16690	8345	285	10.9	15000	40	875	1.1
7-46P XS	0.464	333	19410	9705	285	10.9	15000	40	875	1.1

Table 2-1 Logging cables characteristics

The logging cable is connected to the telemetry modem thru a cable interface circuit. Wire pairs are used to establish up and down links. Power lines are superimposed on communication lines. AC and DC lines are superimposed as well.

DMT modulation [12] is used to modulated, and demodulated, data coming from, and going to, downhole tools.

2.2.2 Local Communications

Packets come from all tool string tools to the telemetry cartridge to be rearranged in super packets, and then sent over the cable. The downstream super packets are divided up to separate packets, arranged in a frame, and then sent to tools below the telemetry cartridge, over a local bus connecting all tools , one of the widely used protocol to do so , is called EFTB. Figure 2-10 shows a flow chart representing the main functions of EFTB. As a communication protocol, EFTB is expressed in layers. Layer 1, is the physical layer when modulation and demodulation take place. Modulation in EFTB is Biphase mark [15]. Layer 2, is the Frame layer, in which data chunks are placed with separating time intervals. Each chunk or packet comes from a tool in the string. That is for both the uplink, and downlink frames. Layer 3, is the Transport layer, where control flags and CRC is calculated and checked for each packet, each tool relays traffic comes from the tool below and adds its own packets , if it has any.

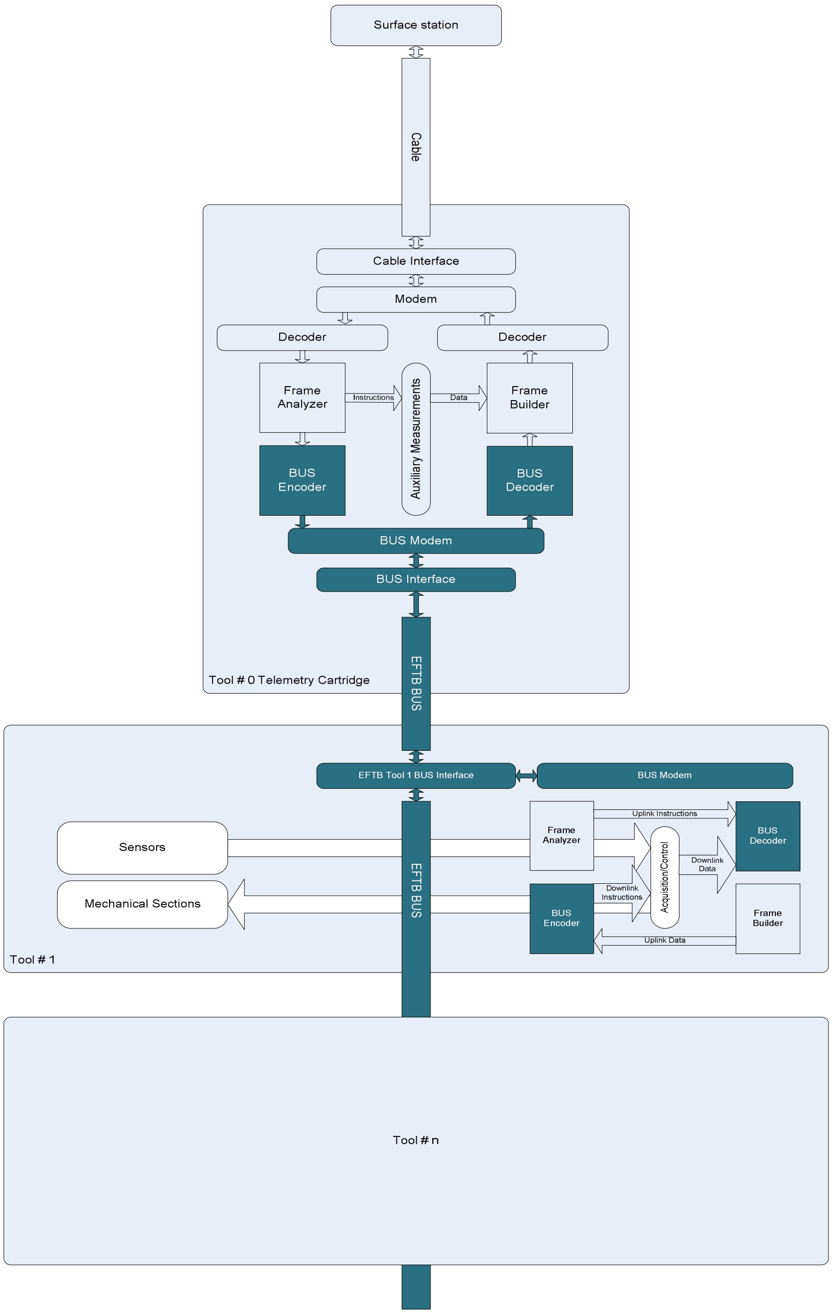


Figure 2-8 Telemetry in logging tool string

Layer 4, is the Message layer, where a message, which is a collection of data, comes from each sensor in the tool, is composed from the coming packets, or divided over packets to be sent. In Figure 2-10, only the first 3 layers are shown. Figure 2-9 shows how Messages are mapped into packets.

	Uplink Message							STATUS	WINDOW	Depart Clock	Address	Length	sync	
	Uplink Message Fields						List Head							
	WORD 6	WORD 5	WORD 4	WORD 3	WORD 2	WORD 1	Word 0							
CRC	M1 NMP = 8000	Msg 0 Word 2	Msg 0 Word 1	Msg 0 Word 0	Msg 0 Clock	M0 NMP = 8000	1	2	1C	Packet 0	0A	7	4B6F	Packet 0
CRC	Msg 1 Word 4	Msg 1 Word 3	Msg 1 Word 2	Msg 1 Word 1	Msg 1 Word 0	Msg 1 Clock	0	0	80	Packet 1	0A	7	4B6F	Packet 1
CRC	Msg 1 Word 10	Msg 1 Word 9	Msg 1 Word 8	Msg 1 Word 7	Msg 1 Word 6	Msg 1 Word 5	0	0	1C	Packet 2	0A	7	4B6F	Packet 2
CRC	Msg 2 Word 1	Msg 2 Word 0	Msg 2 Clock	M2 NMP = 8000	Msg 1 Word 12	Msg 1 Word 11	3	0	80	Packet 3	0A	7	4B6F	Packet 3
CRC	Msg 3 Word 2	Msg 3 Word 1	Msg 3 Word 0	Msg 3 Clock	M3 NMP = 8000	Msg 2 Word 2	2	0	1C	Packet 4	0A	7	4B6F	Packet 4
	CRC	M5 NMP = 8000	Msg 4 Word 1	Msg 4 Word 0	Msg 4 Clock	M4 NMP = 8000	1	0	80	Packet 5	0A	6	4B6F	Packet 5

Figure 2-9 EFTB Packets' Structure

As shown in the figure a Packet consists of Sync word, Length field, Destination address, in which window it will be sent, status word, and the payload filled of words from the upper Message layer. The start of each message in a packet's payload words is indicated by the List Head word. And finally a CRC field is calculated for each packet. Packets are sent in windows. A window is when each node can send its packets. Window size is a mean of traffic control, managed by the telemetry cartridge.

Frames of Packets are composed in layer 2, there are two ways to issue that frame, according to EDTS, there are only One to Many, and Many to One Communication. So, there are two sorts of behavior in this layer. They are Master and, Slave responses.

The Master response in what currently telemetry cartridges do, in there, the one to may frame is composed, transmission window sizes readjusted in other frames. Window sizes and delays are set by the master station. station responds in a simpler way, it waits for its dedicated packet sent in the frame due to the protocol, each packet in the fame have a one bit sequence number, and each packet is sent twice, to avoid retransmission in case of failures.

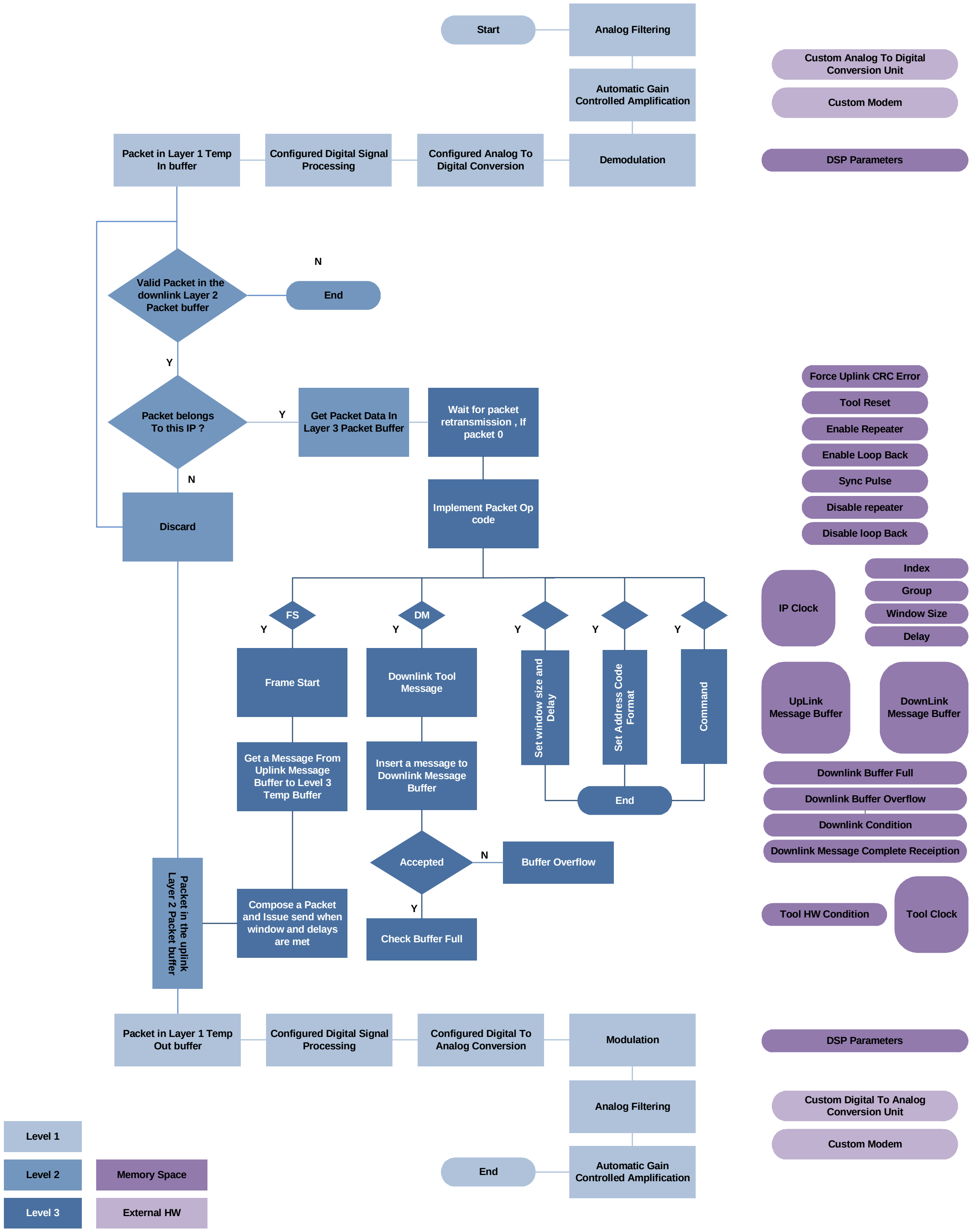


Figure 2-10 EFTB protocol

So , a packet with sequence 0 is the main and with sequence 1 is the retransmission , When a sequence 0 is received , The station will wait for retransmission before processing the received Packet. After the packet is received and processed, the Slave station will send one packet of its uplink buffer when permission is granted and its sending time interval comes. EDTS Specifies the Frame format and delay calculations for each station in the string, according to its place. Frame rate is 62.5 Hz, for 16 ms Frames. More on telemetry can be found in [15], [16].

2.3 Summary.

- The main purpose of this study is to provide another way of implementing some very common logging tools in way to enhance its performance, emphasis on reliability, robustness and reduce maintenance costs of the electronic part.
- Telemetry Cartridge, Inclinometer, Micro resistivity Imager are of the tools under study here.
- Telemetry is the used communication system to both convey data up to the surface and commands downward. Part of the used telemetry system is the communication bus collects data from various logging tools in a tool string.
- Inclinometer system, another ingredient in preparing imaging logs, is based on mapping tool axes into Erath axes.
- Micro-resistivity imaging is one of the major applications, by which Dip measurements are acquired, and other formation and rock properties. The theory of measurements are based on injecting high frequency AC current into the formation and obtain formation response by sensors geometrically placed in order to cover the bore hole.
- Auxiliary Measurements such as temperature head force and Natural Gamma Ray is found in any openhole log, basically for correlation and environment compensation of the main log.

CHAPTER 3

PROPOSED DOWNHOLE DATA ACQUISITION PROCEDURE

In this chapter, the proposed solution is discussed. In order to address the required reliability discussed in chapter 1, and by taking benefit of the advanced processing techniques , a generic acquisition system , that provides enough redundancy , and can tolerate a failure , without an external intervening, is the target solution proposed here. The idea is to transform function specific logging tools into generic control, preprocessing, and communication part, and a function specific sonde part, where all sensors and function specific power supplies are held. What is discussed here, is how to build the first part, how it could provide reliability, fault tolerance, and first of all, to interface the sonde part.

3.1 Behavioral Description

Staring in the behavioral Domain on The Architectural Level, figure 3-1 shows an outline of the proposed Acquisition system. The main differences between this and most applied designs could be seen as communications redundancies. As shown the figure, the proposed design is represented as a main telemetry station and other peripheral stations. The main one, is just a regular peripheral station structure in addition to the cable modem section, denoted as "Main Telemetry station", the backup telemetry station is a repeat of the cable Modem section for redundancy purposes. Each of main and backup station is separated electrically and mechanically from one another, in a way that each can work separately, as mentioned in chapter 1, failures due to flooding will completely dismantle the flooded section , for such estimation, the backup section should be working independently. The box denoted as "Link" represents the shared communication medium. Each station is a communication node; it sends and receives data in both directions. Each has other two data exits. Here, "data" refers to the acquired information from the attached sensor section (Sonde section), data sent as surface commands to certain station, or packets sent by another station. The main telemetry station collects other stations' data, to be sent to the cable modem. The other data exits are "Data outlet "and "Emergency channel interface ". The first one is used to deliver data product to where the acquired or delivered data can go, in case of the main station, either out, after unpacking other stations packets and adding its own acquired measurements data, (auxiliary measurements described in chapter 1 and 2), or in, as when the main station receives the downlink frame data from modem section. The second data exit is the emergency data channel, as the name depicts, another emergency channel, independent from the common station interfaces, could be used in case one station interface completely fails. When a station fails, the failed station's data will be directed to another station that would, at this time, do both home works.

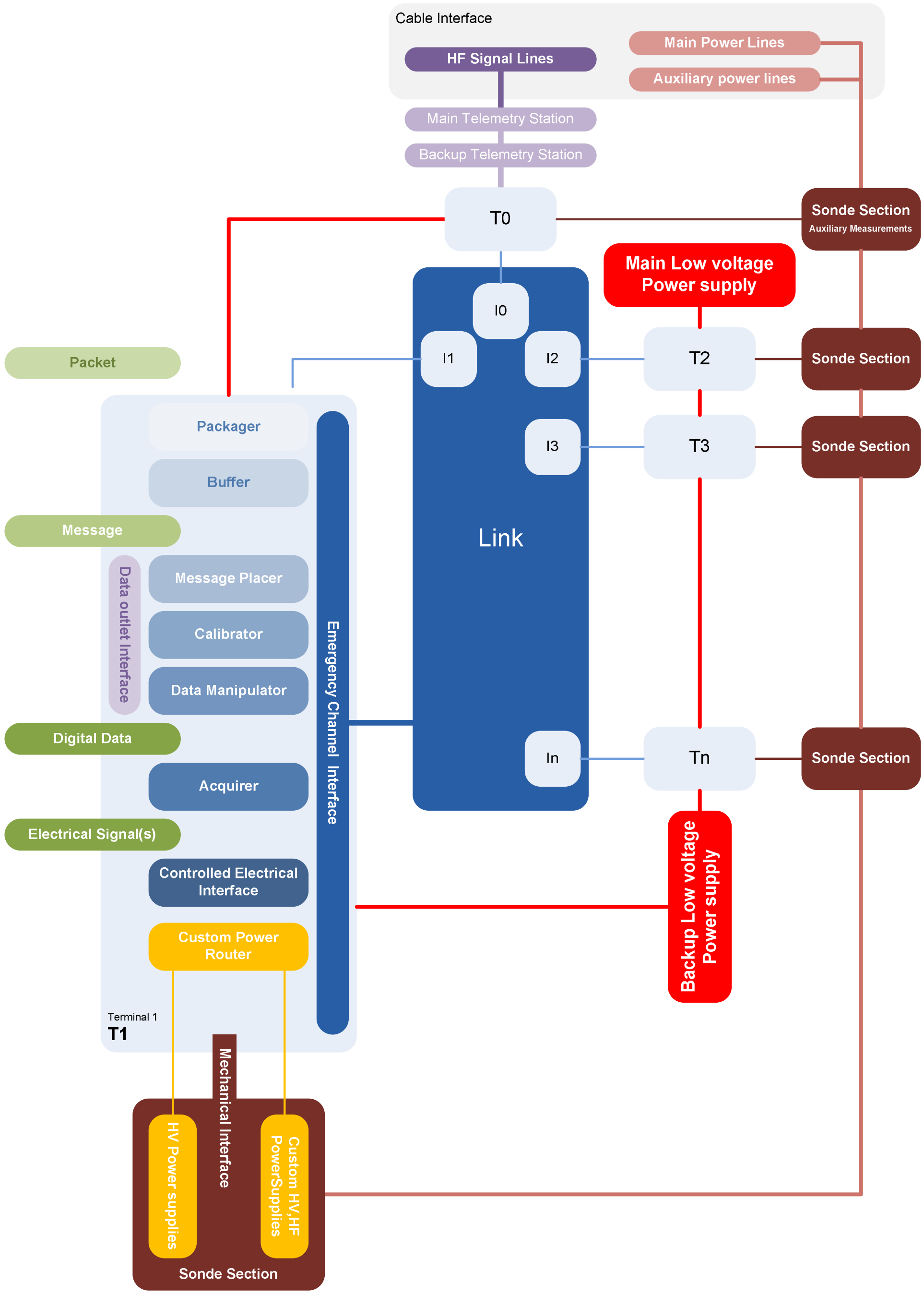


Figure 3-1 Proposed Acquisition Downhole tool string

Main and backup telemetry stations are not passive modems only; they also host error control and data transmission/recovery control over the cable, monitor next stage's status that will pass the payload to or get it from.

The design has to fulfill the following objectives in order to achieve targeted benefits and to be realistic as well:

- Telemetry protocols can be implemented. Migration from one to another still applicable without any Hardware changes.
- Can interface with the same terminals to various sensors, mechanical parts, i-e, it can live with many sonde types.
- Can provide computation power and electrical Interface able to open large space for developing more sophisticated types of sensors, sondes or more intelligent electromechanical parts.
- It has to be so modular and generic in order to cut down manufacturing and maintenance costs.
- It has to provide enough redundancy that the likelihood of having a downhole failure would be much less.
- Real time systems can fit in.
- Digital signal processing is the pivot stone in preparing all kinds of measurements, so, the design should be comfort with that.

3.1.1 Node Processing

As mentioned in Chapter 1, in order to achieve the generality, each node can be any tool, with custom sonde. In this part, that is mainly related to what is to be run on node's main processor, the software configuration proposed is introduced.

One of the Goals of this study is to use the ability of Multi-core – Multiprocessor environment to achieve higher reliability to the application on focus. In order to translate dedicated HW custom made tools into fixable generic SW Application, HW roles is to be turn into processes. The trend taken here, is to represent each HW Tool with a single process that can be run on a single Core, with other independent processes representing other functions of the same tool or other distant (May be failing) tool as well . So this section starts with the representation of tools as SW processes, and then discusses the target multiprocessor environment, generated code for that platform, and finally applying this on one logging task, the Cable Telemetry.

3.1.1.1 Logging Tool String SW Model

Each tool in a logging tool string is assigned a function that it can uniquely achieve. Figure 3-2 shows a typical tool string (on the left) with the corresponding processes it could be converted to. The functions achieved by the some of the discussed services in chapter 2 are shown in the figure, in addition to other processes, those are not shown in the figure, and are embedded in all nodes, which are the Acquisition process, and Inter-Node Link process.

Acquisition Process: The process responsible for dealing with Sonde Interface section, it sends and receive command words, responds to Caches' requests, and emergency requests. Its input is downlink traffic from Inter-Node link process, and output is uplink traffic to the same process.

Inter-Nodes Link Process: It represents The Inter-Nodes Communication protocol. Many, currently available, protocols could apply, none is covered here, for any, the following requirements have to be met.

- Listens to the communication media, terminals announce their presence.
- Each terminal is assigned a unique ID (form which, the 8 bit Node ID is deducted).
- Assigned a Master and descending command hierarchy.
- When a Node Fails, it announces a Node Failure and a restudy of the available working Nodes are done, and a new master is reelected.
- A token is generated at the master and routed it, with parameters, to all in order, then returns back, the master initiates that when requested by other processes.

Input/outputs to/from Inter-Nodes link process, are all other processes, it performs the connection between cable telemetry process to all other processes, and a node to node intercommunications.

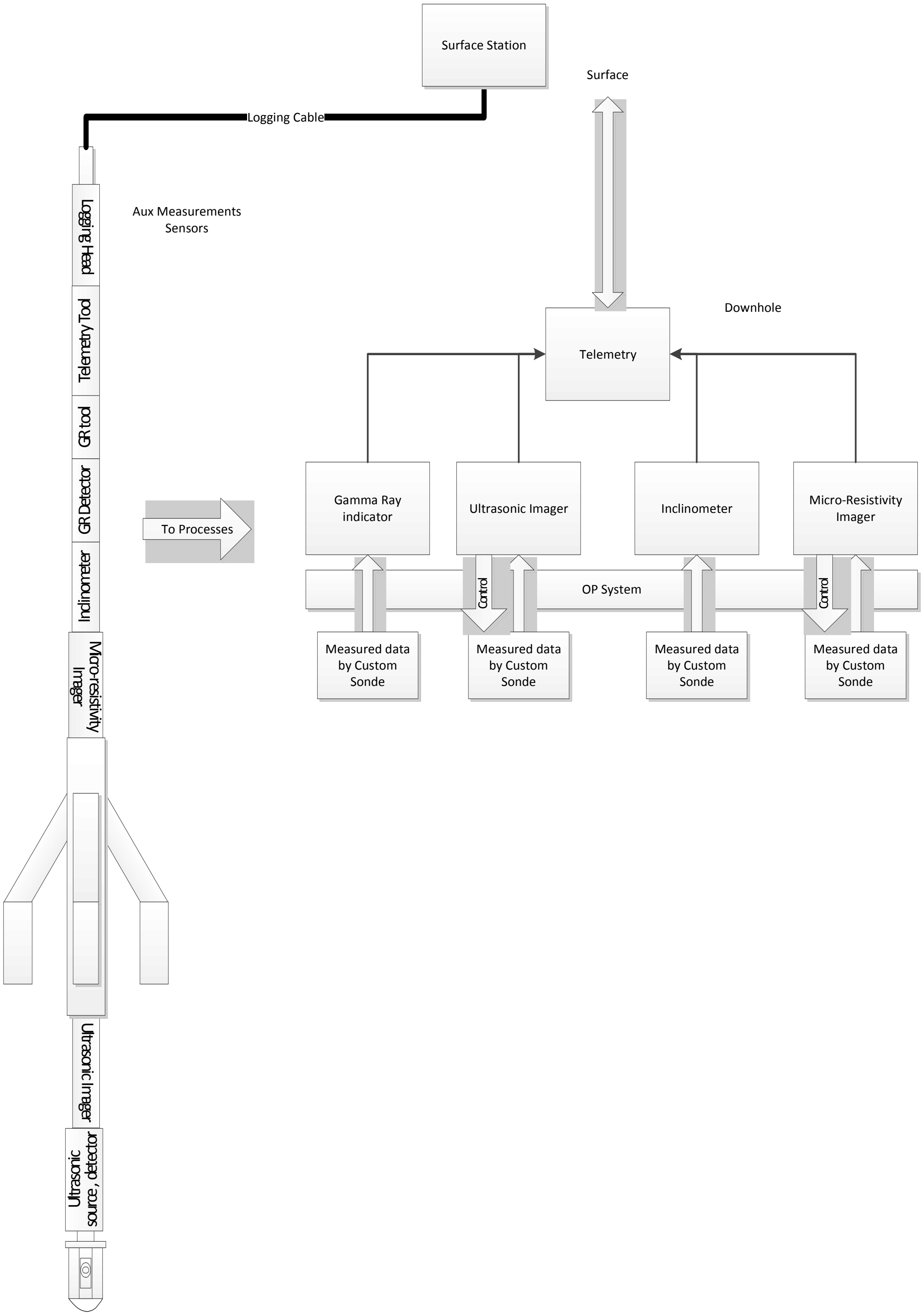


Figure 3-2 Logging Processes

Telemetry Process: The Software Subroutine that carries out cable telemetry tasks, it starts in a default limited mode that will work on any cable. Then it is initialized from surface with the desired parameters to get the maximum from the cable. Its input/output is to Inter-Node Link Process.

Micro-resistivity Dip meter, Auxiliary Measurements, and Ultrasonic Borehole Imaging Processes: Processes responsible for processing the acquired data from sonde interface, According to the sonde type and the required preprocessing for each service. Their input are Acquisition Processes, and output are Inter-Nodes Link process.

3.1.1.2 Node Structure

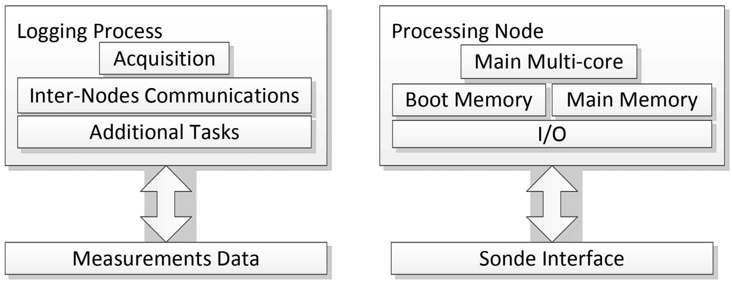


Figure 3-3 Processing Unit (Node)

Figure 3-3 shows a representation of the processing unit in each Node. It hosts:

The main Multi-core, and Boot Memory, which carries: start up operating system, cable telemetry routine, and inter-nodes link routine. It also contains **RAM, and I/O**. Each node takes the following steps.

Boot up Task: After Power up, each Node boots up to be able to perform cable telemetry, and Acquisition, and nodes-Intercommunications.

Load Task: After booting up each Node in row when passed a token, starts its own Cable telemetry process, and directly getting loaded, from surface, by the processes it will execute, as each node is informed by its connected Sonde about the services that Sonde is provided, though the given Sonde Type Figure 3-4 proposes how Loading process could be implemented.

Logging Task: After loading the Logging stage starts, all loaded processes as well as booted processes will work in parallel, occupying Parallel threads offered by Multi-core processor. Figure 5-4 shows a flowchart of the logging Task.

In the following section one of the mention processes, Cable Telemetry, is synthesized.

3.1.2 Logging cable Telemetry as a Process

The Logging cable carries all collected data from all other logging tools packed by the telemetry tool into super frames as discussed in chapter 2. The whole telemetry task can be subdivided to:

- Cable Modem.
- Message Extractor / Packager.
- Message dispatcher / Collector.

The following sections discuss the implementation of first section.

The Used Modulation on Logging cables is Discrete Multitone Modulation [12]. Discrete multitone (DMT) modulation is an attractive method for communication over a non flat channel with possibly colored noise. It is widely used in ADSL. Discrete Multitone (DMT) modulation is a multicarrier technique which makes efficient use of the channel, maximizing the throughput by sending different numbers of bits on different sub-channels. The number of bits on each sub-channel depends on the Signal-to-Noise Ratio of the sub-channel. The performance of a DMT system can be further increased by using powerful Coding Techniques [13], which could be flexibly achieved through the use of enhanced processing HW. During this study the used coding will be allowed to vary according to the channel Capabilities [12] [14]. Figure 3-6 shows a block diagram of DMT Transmitter. DMT steps are:

Parsing Stage is the very first step is usually done through PAM, or QAM Constellation. QAM is chosen here. The resulting of each bit block is a complex symbol. An Input Complex Vector of size N is the result.

Pre Coder: In order to achieve less error rate in the coded symbols, the input block is coded in a way to produce the complex output based on the previous combinations, a (3/2) Convolutional encoder is added [14] as the pre stage, and the number of memory cells (states) is adjustable according to the required depth in history as in [14].In the applied algorithm the input bit stream is coded as follows:

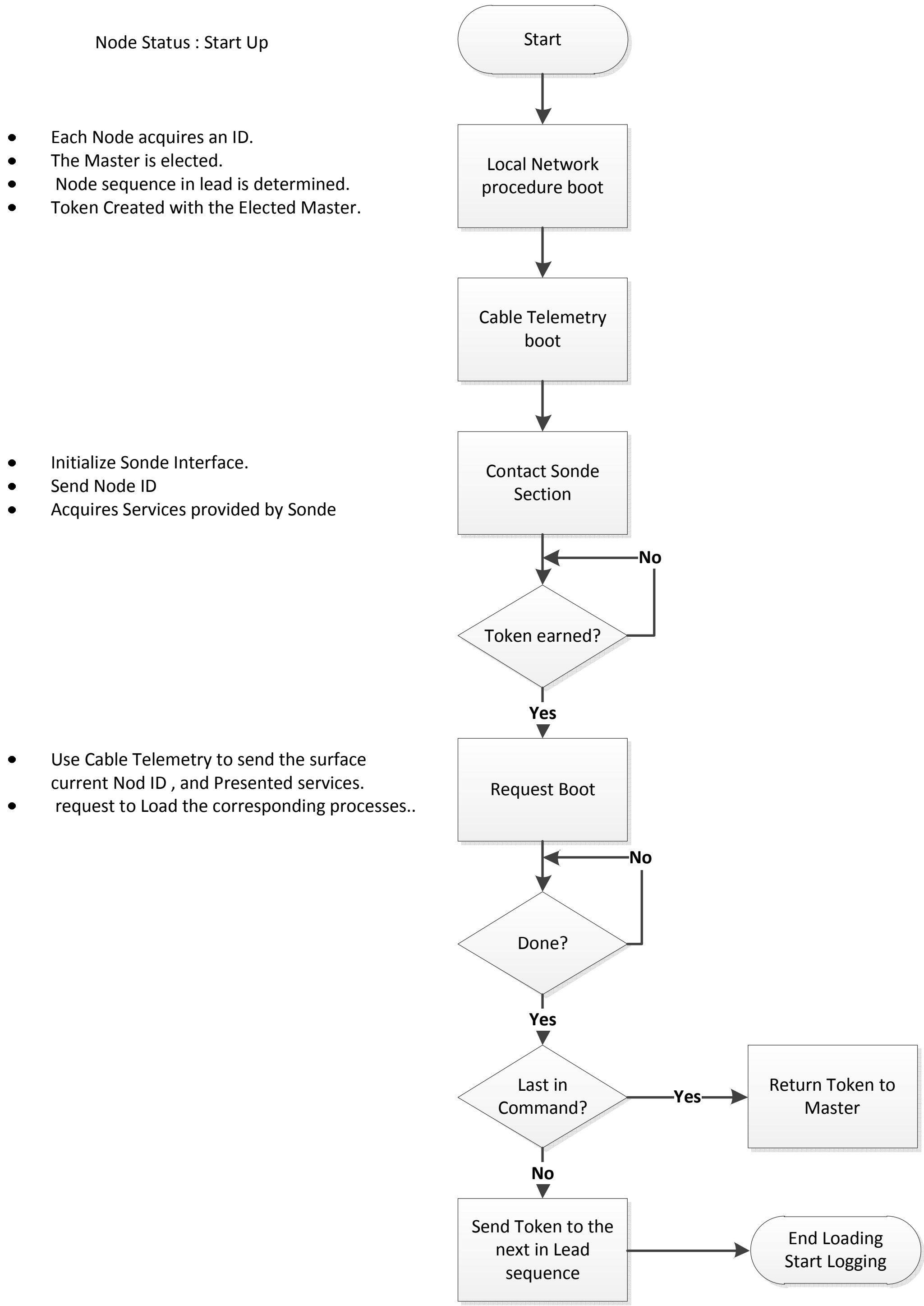


Figure 3-4 Process Loading

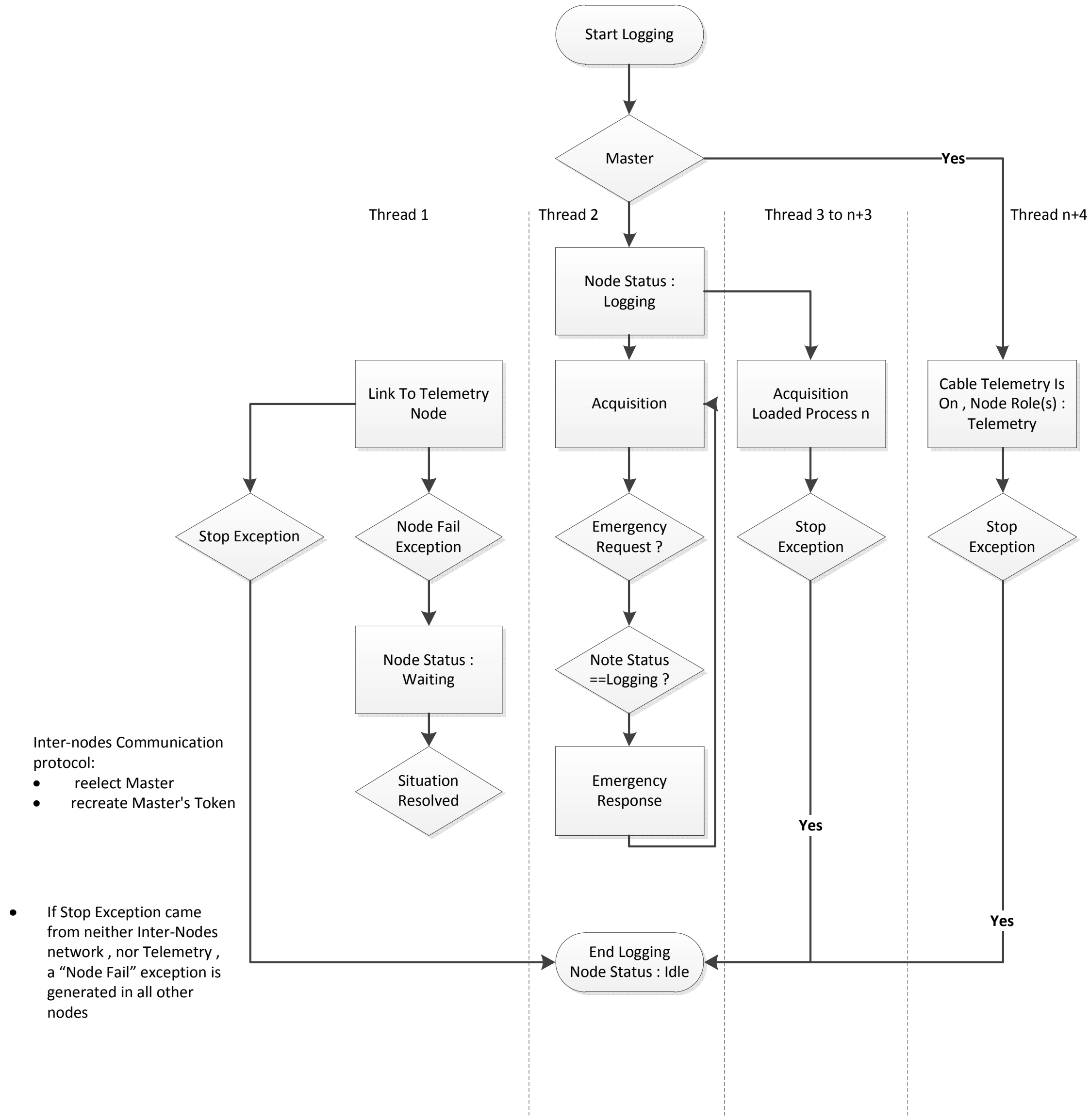


Figure 3-5 Logging Task

$$S_0 = S_1^{-L} \oplus J_1^{-1} \oplus S_0^{-1} \tag{4-1}$$

$$S_i = S_{i+1}^{-1} \oplus J_i^{-1} \qquad , i\ (\ 1\ to\ L-1) \tag{4-2}$$

$$S_{L-1} = S_0^{-1} \tag{4-3}$$

L: Number of States, in the Convolutional encoder.

S_n^{-m} : Sate *n*, at *m* previous time steps (in the m$^{\text{th}}$ previous sample).

The Implemented Convolutional Encoder accepts any number of inputs, and any number of states, The Number of states used is the length of a sequence. In Sequence based Codes, The

Code Gain is a measure of the MECD is increased, which is lower BER (Bit Error Rate) [14], and direct reason for MLD (Maximum Likely hood decoding).

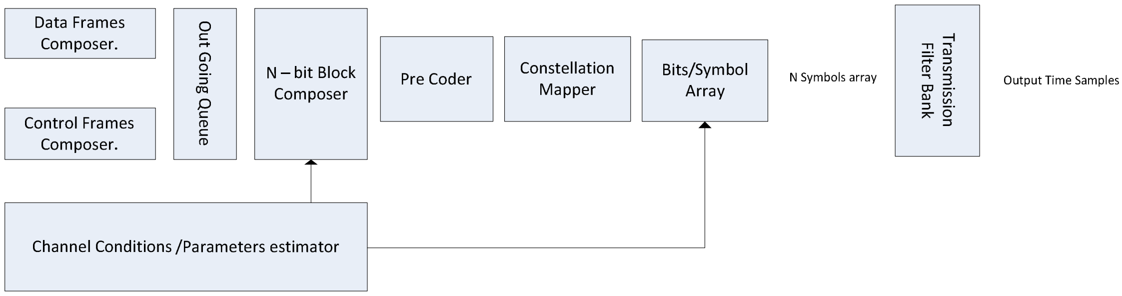


Figure 3-6 DMT – Transmitter

Constellation mapper: Mapping used here is QAM; Mapping is a one way process, in which the decoded N bits vector (by the Convolutional encoder) is mapped to M complex symbols. The encoding process, based on the Code Partitioning Method described in [13], is done by increasing the input and reducing the output, the code efficiency increases, and the throughput, but the required energy per symbol increases as well. In order to reduce the transmitted energy per symbol with the same bit rate, multidimensional constellation is introduced. The implemented code produces any arbitrary multidimensional QAM, although following stages would only require dimensions of multiples of 4. The coding process is a matrix multiplication of the input bit vector with the mapper matrix. Mapper matrix is generated once per setup. It could be changed according to the cable and calculated S/N over the channel, so, channel feedback is used here to readjust all code parts, bit rate, symbol rate, bits per symbol ratio, Constellation dimension, and Convolutional encoder depth up to the moment, For any arbitrary input/ output, there is a mapper matrix as in [14] [13], two types of mapper generator are used in the implemented code. Figure 3-7 shows an example of the generated mapping matrices for 8 dimensions QAM, with 8 input bits, the input is fed to the Convolutional encoder to get 9 bits , with an extra redundant bit used for error correction purposes (to reduce BER , and allow rapid error recovery) . As in [13] the imposed automatically generated matrices are leading to partitioning the output symbols in away as in part b in figure 3-7.

Number of Convolutional Encoder (3/2) needed: 1 each symbol is equivalent to 8 bits, Total Parts are: 1

0	0	1	0	0	4	2	4	0
0	1	1	0	2	4	2	4	4
1	1	1	2	2	4	2	0	4

(a) Mapping Matrix generated for (9 to 3) mapper.

00000000000 maps to: 0 0 0 0
10000000000 maps to: 0 0 4 4
01000000000 maps to: 0 4 4 0
11000000000 maps to: 0 4 0 4
00100000000 maps to: 4 4 4 4
10100000000 maps to: 4 4 0 0

(b) First 6 entries in the De-mapper Look Up Table. which also shows the effect of Symbols partitioning in Multidimensional Constellation.

Figure 3-7 Example of Constellation mapping

The figure also denotes the way adopted here to perform the decoding. As the target environment is fixable enough to accept variable parameters attributes and produce fully custom modem, the de-mapping, actually the whole decoding process is implemented via LUT. Just one memory process for each symbol. The decoding process in applied DMT is using Maximum likelihood decoding, through the implementation of Viterbi Algorithm [22].

Bits per symbol array: A famous feature of DMT systems is that in tough conditions (high noise channels) the number of bits sent over each sub-channel could be reduced, but another matrix multiplication between the produced symbols from mapping and generated bits /channel matrix.

Transmission filter bank: The following is the usual derivation of DMT Transmission and reception principle. The input to this stage is an array of complex symbols representing the mapped code. The symbol array (vector) is multiplied by Tones Vector, matrix Multiplication to result in the time signal s(t), where:

$$s(t) = \sum_{k=-\infty}^{\infty} \sum_{n=0}^{N-1} a_k^{(n)} \; g_n(t - kT) \qquad (4-4)$$

Where:

s(t): Base Band Envelop of the Transmitted Signal.

$\sum_{k=-\infty}^{\infty} a_k^{(n)}$: Complex representation of each Original Symbol of the main Data Stream.

$g_n(t)$: N orthogonal basis Functions [14] (Tones) of Amplitude c and duration T. It is Chosen here as:

$$g_n(t) = \left(\frac{1}{\sqrt{N}}\right)\sum_{k=0}^{N-1} e^{\frac{j2\pi nk}{N}}\ p\left(t - \frac{kT}{N}\right), n \in \{0, \dots, N-1\} \qquad (4-5)$$

Where: p (t) is an ideal unit-energy reconstruction filter for a sample rate of N/T:

$$p(t) = \sqrt{\frac{T\sin\left(\frac{\pi N t}{T}\right)}{N\,\pi\,T}} \qquad (4-6)$$

At Each Symbol:

$$s(t) = \sum_{n=0}^{N-1} a^{(n)}\ g_n(t) \qquad (4-7)$$

$$s(t) = \sum_{n=0}^{N-1} a^{(n)}\ \left(\frac{1}{\sqrt{N}}\right)\sum_{k=0}^{N-1} e^{\frac{j2\pi nk}{N}}\ p\left(t - \frac{kT}{N}\right) \qquad (4-8)$$

$$s(t) = \sum_{k=0}^{N-1} \left(\frac{1}{\sqrt{N}}\right)\left(\sum_{n=0}^{N-1} a^{(n)} e^{\frac{j2\pi nk}{N}}\right)\ p\left(t - \frac{kT}{N}\right) \qquad (4-9)$$

$$s(t) = \sum_{k=0}^{N-1} \left(\frac{1}{\sqrt{N}}\right) A_k p\left(t - \frac{kT}{N}\right) \qquad (4-10)$$

Where $A_k = \left(\sum_{n=0}^{N-1} a^{(n)} e^{\frac{j2\pi nk}{N}}\right) = \mathrm{IDFT}\{a_n\}$

The implementation is illustrated in Figure 3-8.

Channel Control: This section is responsible for performing S/N Measurements over the channel; adjust, according to a learnt response model, the proper Channel width (number of sub channels, Constellation dimensions, and bits per symbol array. That is no so far implemented here.

Channel Equalization: Many techniques are used to equalize channel effects, either known via a channel model or measured during testing channel response. In the current implementation channel equalization isn’t included.

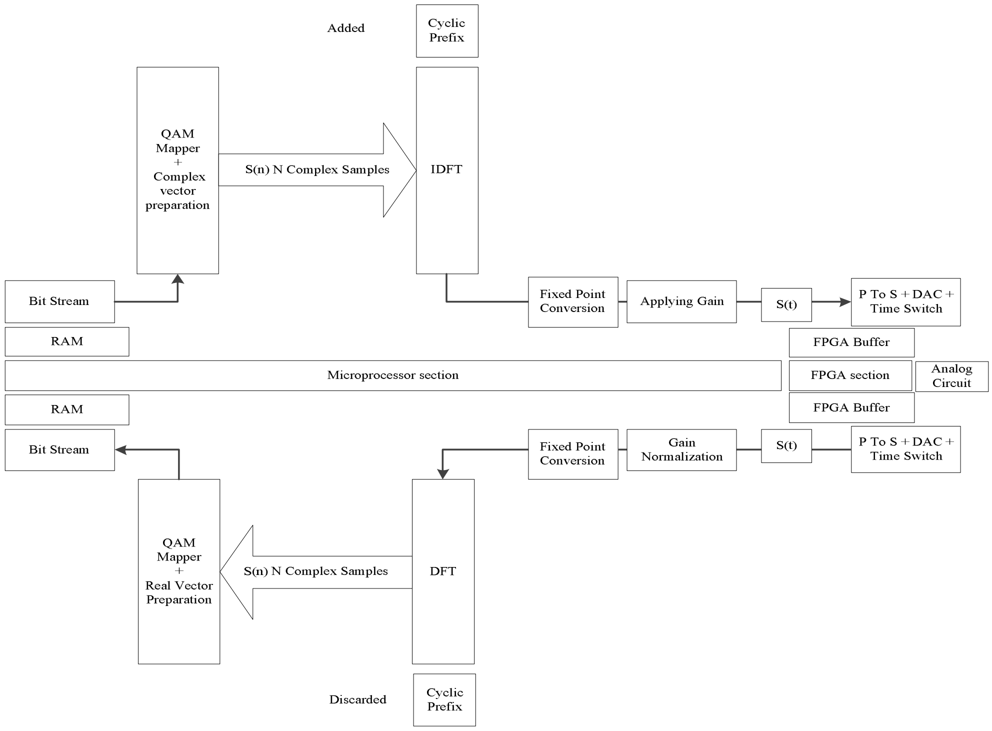


Figure 3-8 DMT Modem

3.1.3 Node Communication

Figure 3-9 shows the parts undertaken in this section. The communication system here is the one responsible for linking nodes. It is the data feed mechanism to the mentioned “Inter-Nodes” communication tasks achieved by the main processor, mentioned in the previous section. It describes the desired behavior of “Inter-Node” task, in addition to the emergency way out , which is a part of dedicated HW in the sonde interface , as will be described in next sections, it runs on the main processor, starting from row measurement data acquired by the node. According The Conventional OSI network layers model, only four layers are here. At each the exchanged payload would be named and shaped differently, starting from blank sensors measurement data, to electrical waveform on the shared media link. To get some redundancy in case of a failure in any of the shown sections, an Emergency channel is added to take the charge, this channel should be able to modulate, carry, and demodulate data at blank sensor measurement data level, Message level, and Packet level, i-e only physical level manipulation, the whole task is to route the load to another interface, which is supposed to do both jobs for himself and the assigned tasks of the sick-left interface. As the Emergency can route data of any of the three mention levels, and that it is a physical only medium that won’t intervene with packets, messages or higher level data structures, The Routed data should always be self-expressed.

From the previous section, in order to achieve the compatibility with any local communication protocol, it has to be applied completely through software, a DSP section to achieve the required modulation, and a physical interface with the transmission medium. EFTB, mentioned in chapter 2, and other protocols can be applied, though, they all share the physical interface to the shared medium. Ethernet is a candidate, as well. Local communications between nodes are not presented further in this study.

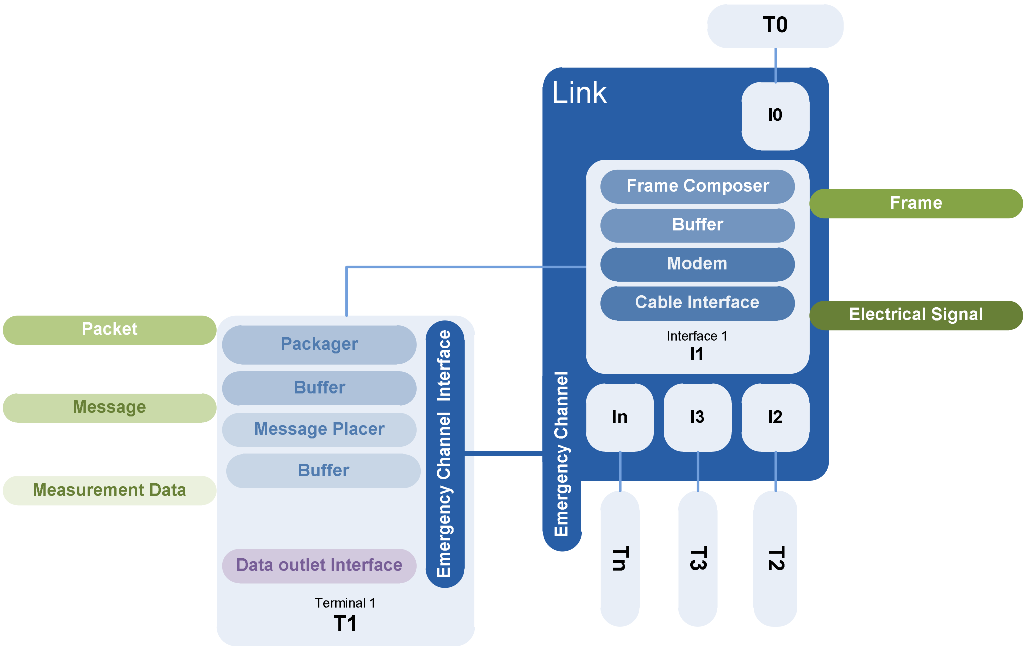


Figure 3-9 Node Communication

3.2 Functional Description

The whole design can be divided up into three main functions, communications, acquisition, and processing sections. In order to achieve the required redundancy, and modularity, the tool string is designed to be of identical nodes, of equal capabilities.

3.2.1 Node Acquisition

Each Sonde has its special data exchange and control requirements, this section, is the part of the proposed design that deals with custom sonde requirements through a generic interface. In order to achieved target reliability (generic interfaces allows other idle parts of the string to work instead of the defected ones). Figure 3-10 shows a block diagram of the proposed Generic Interface to Sonde Sections. Upon their functions, Sonde interface is divided into the following functions.

Data/Command Tagging: To be able to route commands received from surface and to mark data with its producing sensor/ array ID, a Tag is placed as a header in data chunks and command words. Tag hosts Data/Command type, ID of the Sensor/ array of sensors/actuator it is attached to. All words are of a fixed size.

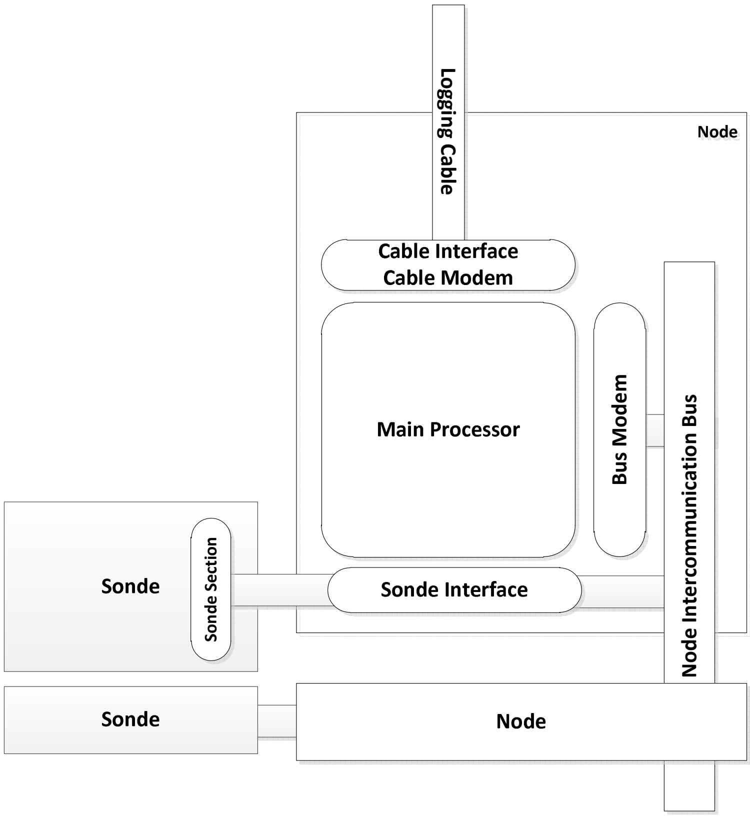


Figure 3-10 Design's Functional Units

Caches: the sonde interface has caches contain data frames with tags. They are used to buffer traffic upward and downward and interface data frames with the main microprocessor, or the emergency channel, when data is routed to another node, in case of a failure in microprocessor section hosted by their node.

Interface Health watcher: It Monitors channel health, by comparing expected traffic according to its traffic table, and assign /switch to another. It Also Monitors Channels reported errors to detect inappropriate behavior of the channel itself, or the corresponding sonde port. It does this on samples over time, when an error happens, monitors channel status, if it didn't return to its proper state after certain time (2 frame times) it reports a malfunctioning channel and ask for channel isolation, if all channels are occupied – malfunctioning, a sonde failure is reported to the main processor.

Synchronous Data Channels: This type of channels receives the generated digitized data frames (samples, combined samples or a certain pattern, according to the sonde type – tagged with sensor id data header). Sonde requirements are updated at start up the channel sends enable line to the sensor array as a whole. On the sonde side, any channel can be fed by multiplexed traffic, the array is treated as a normal channel; the only difference is at Sonde side. The exchanged data chunks have different sensors IDs, unlike the other single input channels.

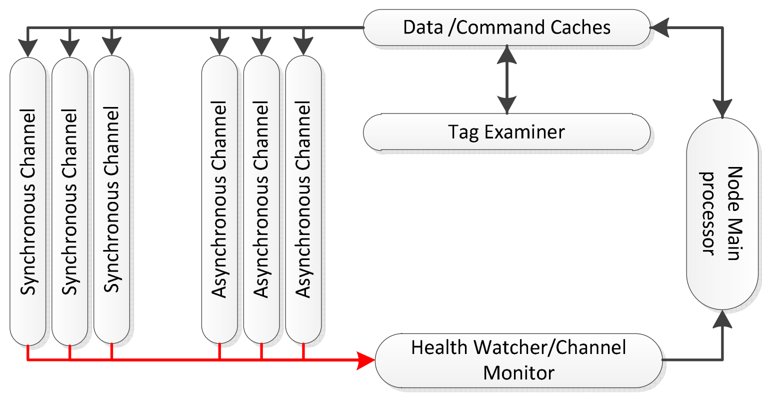


Figure 3-11 Node-Sonde Data Exchange

Asynchronous Command/Data Channels: when receive a command tagged frame, it assets write, and waits for acknowledgement. It could be i2c bus and controllers on both sides. It reports to the local interface health checker.

In the implementation, sixteen Asynchronous channels only are implemented, have of them are receivers, and the other half are transmitters.

Emergency Channel Interface: in case of an irreparable error occurs with a terminal, that disallows regular behavior, like a hardware failure in any level, and other higher level redundancies couldn't be used, another final way out channel, takes it from the data frames from and to Sondes, with their node IDs, and route it to another near station, that agrees to host this another burden. In case of an emergency, all node acquired data, and surface/ main node processor's commands can be routed to other nodes, given that sonde section is in well condition and can work. In such a case, the other "rescuing node" processing unit will take over both workloads.

Tagging: In order to address each Sensor via each channel, the data frame is divided in a way that:

- Each frame has a universal address, in order to host frames from other nodes, which is the case of routing frames from a failed node to another.
- The universal address is 15 bit: which limits the number of all sensors /actuators to 32,768, which is much bigger than the most composed tool string used now.
- The first bit is 0 to indicate that the frame is acquisition frame, not a command as discussed in the next section.
- The frame addressed composed of node id and channel id. Node id is chopped when frames are assigned to channels.
- In this implementation, as mentioned before, only 16 channels are addressed.

In each channel, the last 3 bits are chopped (representing 8 channels Max / Interface).

32 bit Acquisition Frame

0	Node ID	Channel ID	Sensor Id	Sample Frame
1 bit	7 bit	4 bit	4 bit	16 bits

Figure 3-12 Data Frame Address/Tag

3.2.2 Emergency Response plan

Each sensor needs a unique Identifier (in case of measured data are swept to another node for transmission. Universal tags are not removed. Data chunks, by the main processor, are placed in the proper telemetry protocol message.

When a Node Fails, i-e its Processor Node doesn’t respond with commands when status words are sent, sonde interface’s local controller starts initializing the emergency exit. It will send a request via Emergency channel’s Modem on the public channel (that all other nodes can see). When other Node senses the request, its emergency channel sends that request to the local controller which forwards it to the main processor in the next status word. When a command word comes with “permission granted”, the rescuer node’s emergency channels modem sends its code word, upon which the emergency modem at the failing Node tunes itself with the rescuer modem. For this situation, only one failing node will be treated at a time. The situation of more than one failing node is not proposed here.

Emergency channel's modem depends on the media connecting the nodes, the design of that modem and the corresponding channel is not covered here

3.3 RTL Level Description

3.3.1 Communications FPGAs

Cable Modem Modulation: Communication over the logging cable is one of the vital tasks handled by microprocessor section. It is divided between multiprocessing model and FPGA section. The implementation of the latter is discussed here; the rest is in further section. Tasks done by The Cable Interface FPGA are:

- Adjust the gain of programmable gain amplifiers of cable interface circuit.
- Send and receive symbol stream from the main processor in the form of 16 bit frames.
- Buffers samples and send them to ADC, and DAC with Baud Rate Adjusted By the main processor. Figure 3-13 shows the FSM used with Cable Interface FPGA.

Node Intercommunications, Bus Modem: bus modem carries the required modulation and bus interfacing for the main processor's sake. It is represented by two (redundant) FPGAs and their associated circuitry. Communication between nodes is not implemented here. As in next sections, internodes communication is an inter processor communication media, an Ethernet is an option, token passing is implemented through EFTB - chapter 2.

Figure 3-13 Cable Interface FPGA FSM

Another way is to use the same HW and SW used in cable telemetry processes, later in this chapter, it will offer better error recovery, and noise immunity, but with the fact that it is not a big distance, simpler ways could be implemented, with simpler modulation and coding. This interface is not covered here.

3.3.2 Acquisition FPGAs

It Interfaces Sonde to the Main Processor and Node Intercommunications Bus. It consists of two (redundant) FPGAs. Its main functions are:

- To interface the sonde section channels.
- Collect sent frames from Sonde Channels and store them for Uplink Transmission.
- Store Downlink Frames and prepare them for transmission to the specific Sonde sections.

Figure 4-14 shows the main Block Diagram of Sonde Interface and Sonde section. The Following Section Describes the Implementation of Each part.

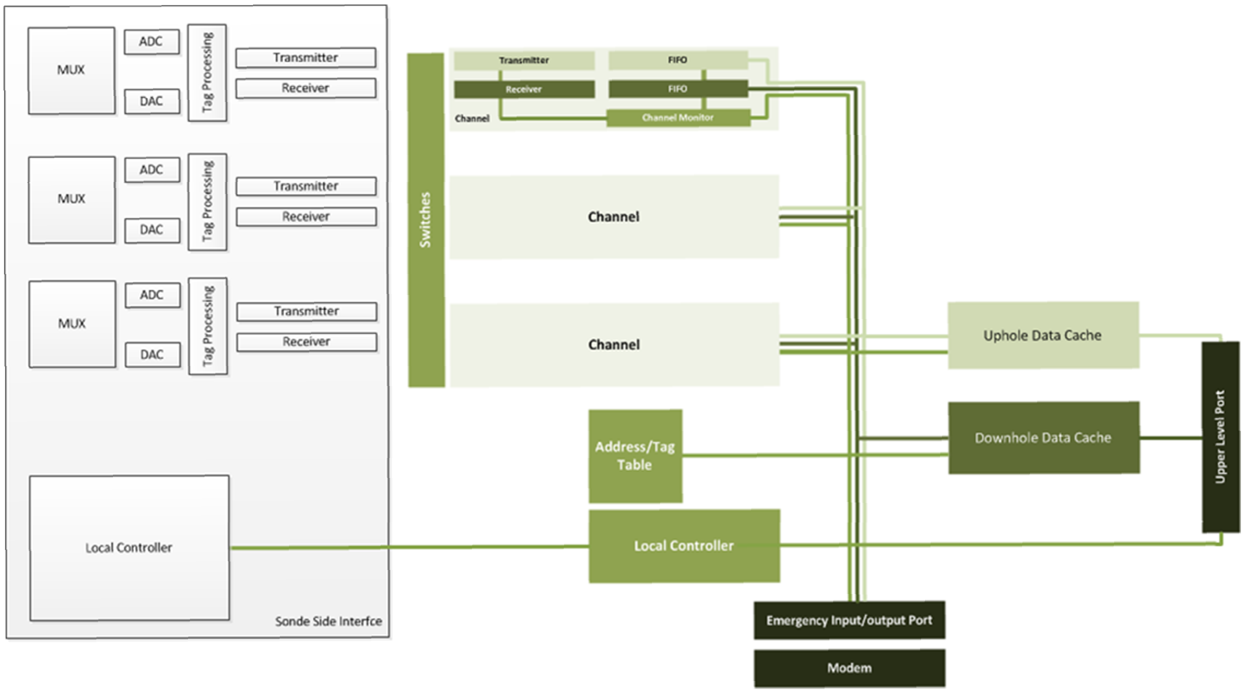


Figure 3-14 Sonde Interface FPGA Block Diagram

Channel: Channels are the first interact with sonde section, it could be either Synchronous, or asynchronous, only asynchronous channels are implemented here. Channels are implemented on basis of i^2c protocol, with speed adjustment. to handle a clock stretching situation. maximum waiting time will be monitored by the master, which is always on the Sonde interface side (node side , not the sonde side), after the maximum waiting time, it send abort sequence on data channel, while clock is low, when detected by the slave it lets the clock and cease transmitting. When the master detects the clock, it appoints the next transmission. writing in i^2c is a straight forward , send the address , followed by data , but to read , it will be a write first , then read , write the address of the slave channel allowed to speak , so , write , supply clock , then listen. But here depending on the setup, relating to each sonde type, the multiplexed channels are talking on a predefined schedule. So, there is no need to send the address, which is a deviation of i^2c protocol. An example of channels dedication is when applying EDTC's auxiliary measurements (telemetry cartridge currently used by Schlumberger) shown in table 3-1.

Measured Property	Signal Type	Range	Rate	Expressed in Bits
Temperature:	Analog	15 to -15 Volts.	1/ 5 sec	16 bit ADC
Accelerometer:	Analog	Analog 15 to -15 Volts.	1/ 10 sec	16 bit ADC
Head tension:	Analog	Analog 15 to -15 Volts.	1/0.5 sec	16 bit ADC
PM High Voltage Monitor :	Analog	Analog 15 to -15 Volts.	1/1 sec	16 bit ADC
Gamma Ray Counts :	Digital	0 to 5 volts.	1/ 1 sec	16 bit ADC
High Voltage Set Up :	Digital	0 to 5 Volts	If required	16 bit ADC

Table 3-1 Channel Assignments for Auxiliary Measurements

The simple UART represents an Asynchronous exchange media, via which commands, statuses could be exchanged smoothly. As i^2c is used, each channel could be represented by a pair of wires that makes them 32 data lines. One channel can be used to multiplex more than one measurement from figure 3-12 , up to 16 different measurement can be acquired in one channel, these measurements are multiplexed , and then de-multiplexed as needed in sonde section , as the first 4 bits only on “Channel Id” are to select one among the 16 channels. Actually in each direction, only 8 channels to select between, but as the same frame is used for control, when a remote rescuing node processer is controlling a failed node sonde interface, each channel of the 16 has to keep its id to allow a selection among all the 16 channel. Each pair of the 8 channels pairs is two separate channels, one is a transmitter only, and the other is receiver only. Table 3-2 shows a summary of channels characteristics.

Active channels update. If miss communications happens, no transmission in a receiver channel, or no acknowledgement in a transmitter channel, they will still try to establish a link for specific time, 48clock cycles (2 frames). if no stable link, a channel fail is sent to the local controller, and another channel is assigned with “Exchange Task”, in which, the local controller sends to sonde controller a message contains the failed channels, the sonde controller is supposed to reconfigure channels, or just be informed, depending on the firmware of the sonde, then an update request is send to node’s local controller, which in turn update its active channels record. if all channels fails, a” Sonde Interface fail” status word is sent to the main controller, by which the backup local controller and its peripherals will take place.

Property	Value	Unit
Channels' Count	16	Channel
Sensor-Actuator /Channel	16	Sensor
Sensors -Actuators /Tool	8192	Sensor
Data Lines /Channel	2	Lines
Overall Data Lines / Single Tool	32	Lines
Interface Used	I^2C (with some deviations – No Addressing)	

Table 3-2 Channels Characteristics

For a simple UART structure is FIFOs, register file, transmitter and receiver. All UARTs are connected to the local processor via a controller bus. As a design parameter the used bus is virtual component bus (VCBUS) as it offers a simple bus structure.

Channel Transmitter: Each Channel's Transmitter consist of a FIFO with (8 of 24 bits) Capacity, with Local Controller Interface, Downlink Cache Interface.

Channel Receiver: Each channel has a receiver consists of RX-FIFO, Local Controller's Interface (same as Transmitter), and Uplink Cache interface.

Both receiver's and transmitter's FIFOs are 8 by 24 Bits Size, with embedded parallel to serial (TX-FIFO) and serial to parallel (RX_FIFO).

Channel Monitor: Channel Monitor section is responsible for monitoring the behavior Of Channel components and it keeps a counter record of errors (Transmission error, Reception errors, Internal Errors of each of transmitters, receivers, FIFOs). Its Report is sent to the Local Controller to calculate the health of all channel's Component, and estimate the necessity of switching to other channels in case of failure. It is implemented as a part of each of Channel Transmitter and Receiver. The calculations and failures record is done in the local Controller.

Universal Caches: "universal" refers to that it is addressed by the universal frame tags. with all frames are written in the order that the controller tells each channel FIFOs to write, it could be simply seen as a queue that the local processor (main node processor) has arranged upon his priority calculation and frames are to be sent in that sequence to reflect a priority pattern chosen by the local controller.

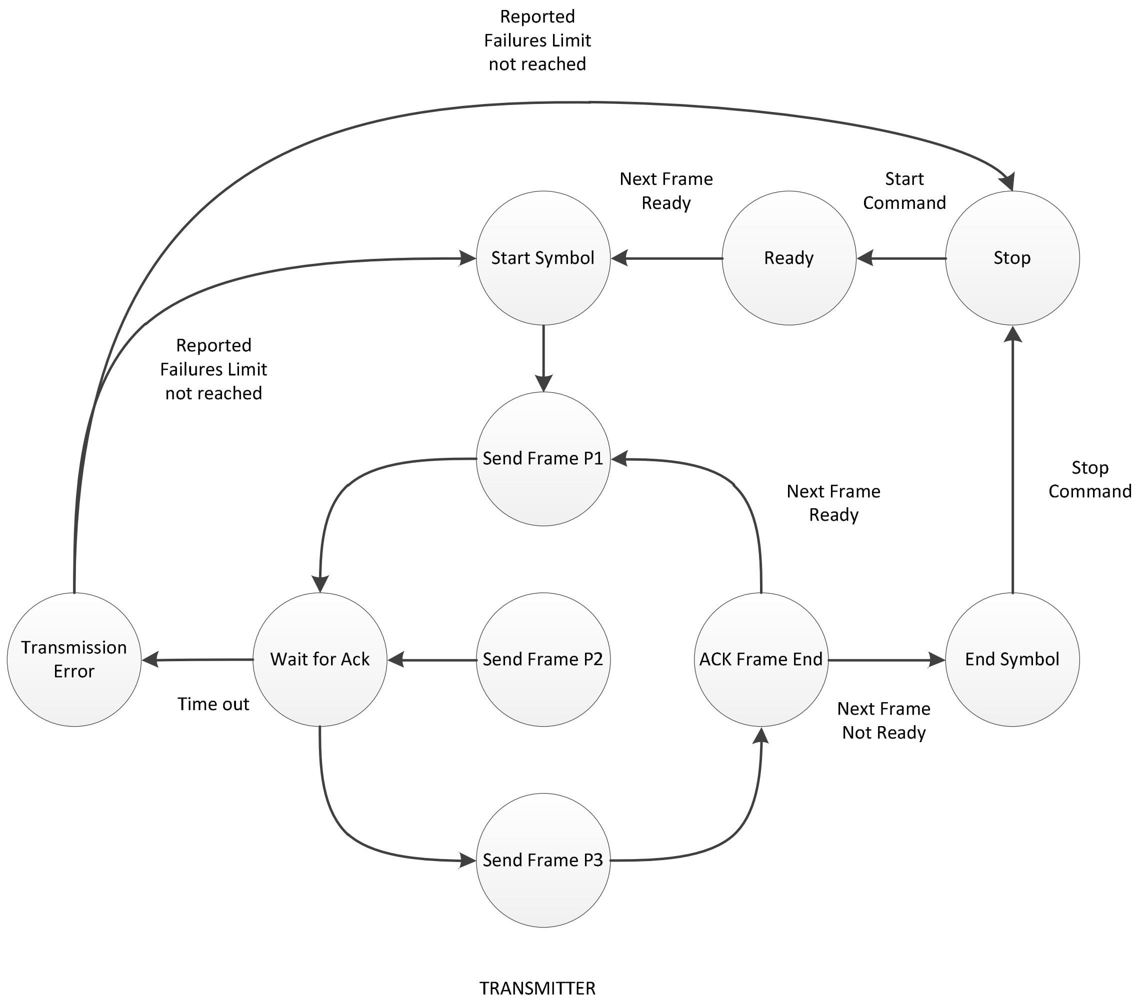


Figure 3-15 FSM of Channel Transmitter (Controller Interface)

So, it could have one entrance and one exit. Data frames enter from the door as permitted by the controller. And each frame in sequence gets out. The purpose of this universal cache is a buffer to compensate time differences between local communications link, or equivalent, and acquisition.

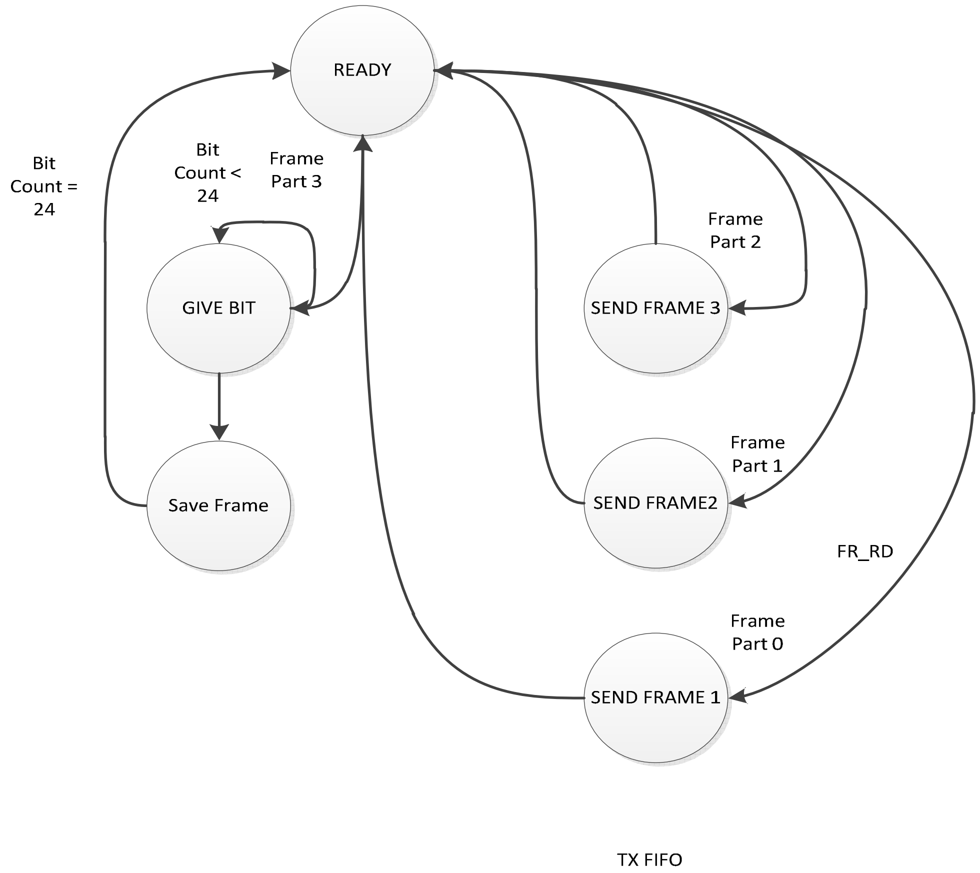


Figure 3-16 FSM of Channel Transmitter (Transmitters' FIFO)

It arranges frames according to the chosen priority. Priority could change, but only for the new frames not the already acquired. It hosts frames with universal addresses, in case other cache is routing its load, the input (is considered as another FIFO), will take its overall priority (or simply a round robin).

As there are two separate defend streams, one uphole and one downhole, it might be a little more convenient to have separate caches for each separate stream. Uphole cache: takes data frames to the processing/telemetry section. Downhole cache: takes data frames from processing/telemetry section to their proper channels. Figure 3- 18 shows FSM of node caches.

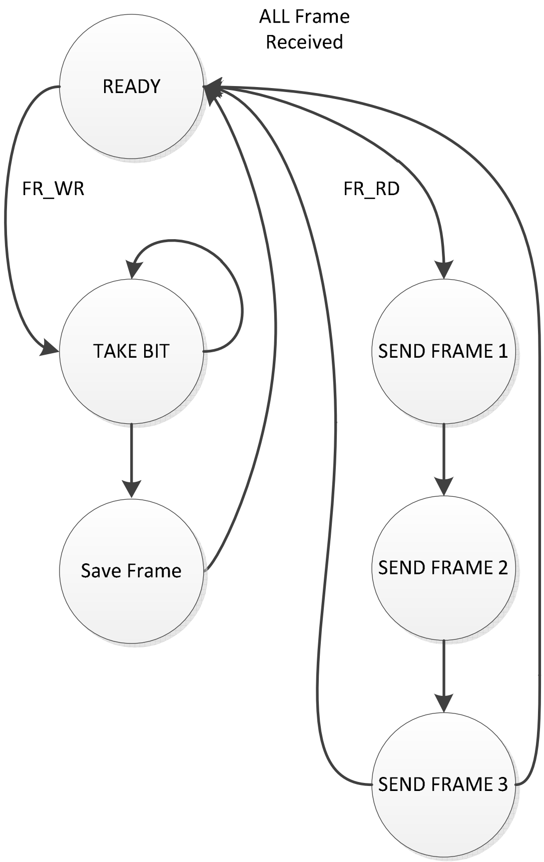


Figure 3-17 FSM of Channel Receiver (RX FIFO)

Caches sizes are calculated when sonde acquisition speed, and local communications speed are known. That calculation is not done here, therefore cache sizes are put as 128 frames of 32 bits.

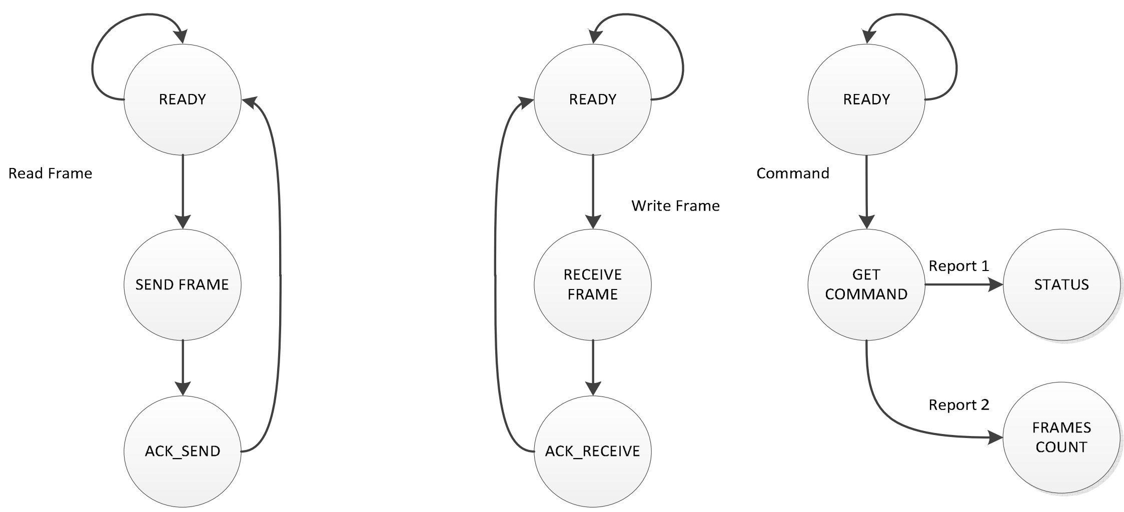


Figure 3-18 Node Caches – Caches Read / Write Tasks

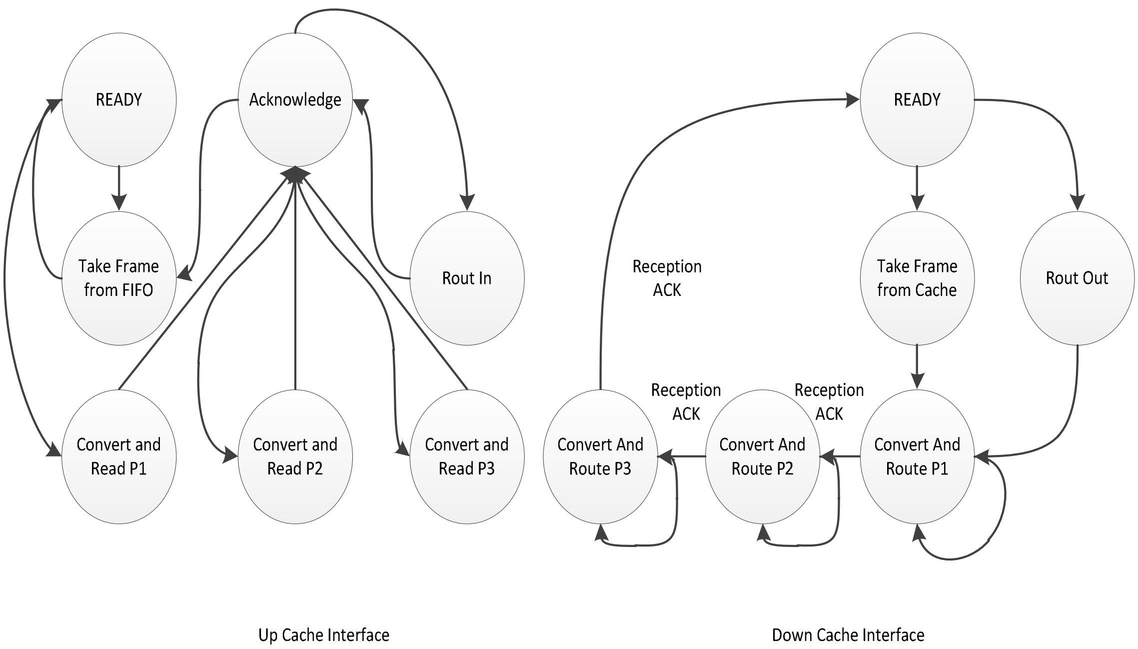


Figure 3-19 Local Caches – Caches Interfaces FSM

Local Controller: Figure 3-20 shows FSM of node's local controller. It is connected to the main processor, via a 32 bits bus, and to caches, sonde channels and sonde controller. It receives errors signals from channels and emergency channel, only when prompted to send, during status report gathering cycle, that takes place at start up, or when required by node's main processor.

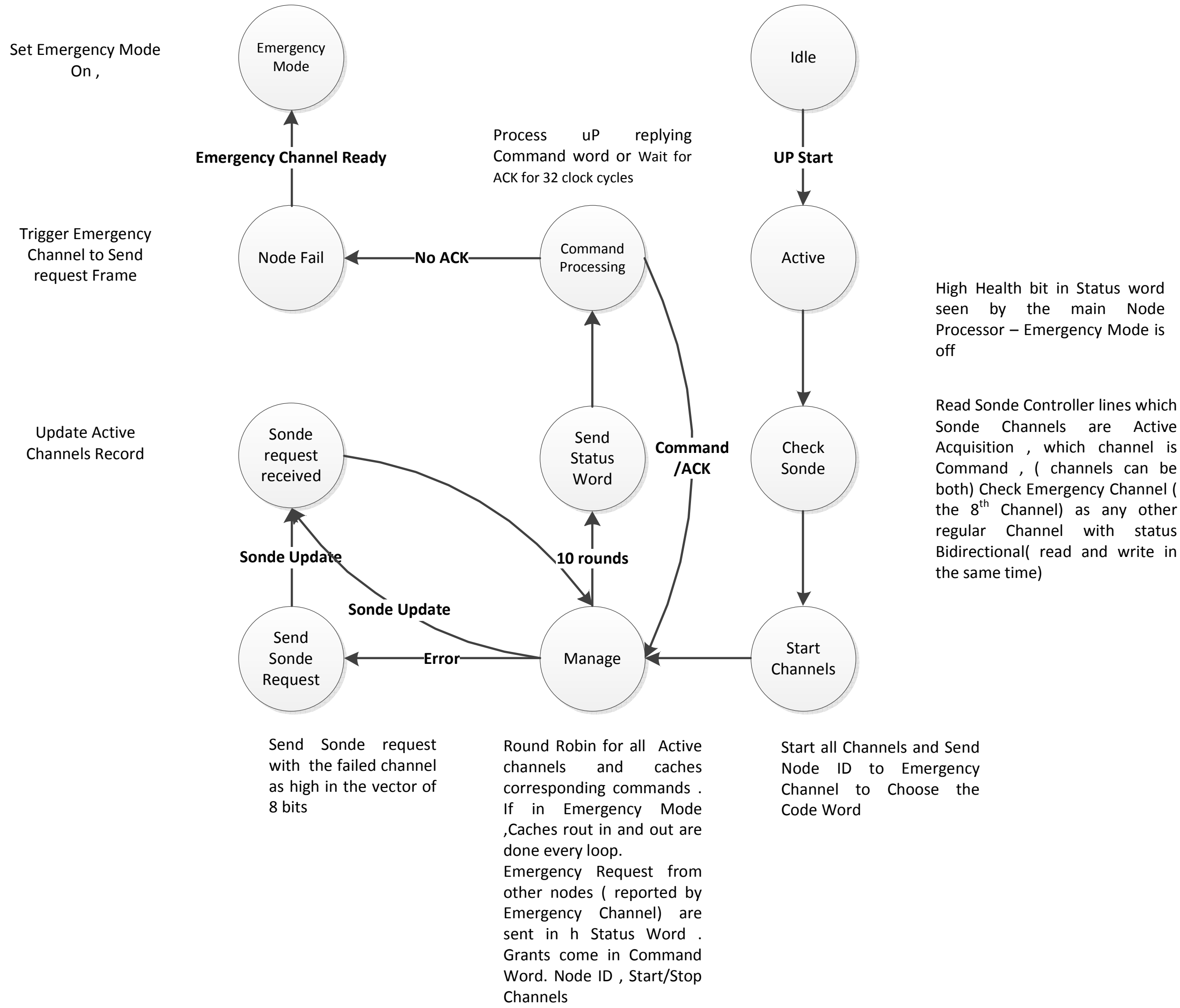


Figure 3-20 Local Controller – FSM

Controller’s Role: the local controller in each sonde interface in each node is responsible for managing inputs to universal caches; assign a “go” command to channels, and to rout emergency flow. The local controller and main controller communicate through status word / command 32 bit lines. Table 3-3 shows status/command word mapping. The local controller sends status word after a round robin cycle in which it sends reading/writing permits to all active channels, and waits for command word. If both are down, the main controller send “interface fail” flag to the surface. When a routed data frames from another “failed” node take their place in the cache in a manner arranged by the controller, the same sequence in issuing frames from the cache will now represents the other node’s data.

Byte 3		Byte 2	Byte 1		Byte 0	
bit 7	7 bits	8 bits	4 bits	4 bits	8 bits	Description
			From Node's Main Processor			
1	Node ID	00000001	10101010		10101010	Set Node ID
1	Node ID	00001011	00000000		11111111	Blank Acknowledgement
1	Node ID	00001001	11111111		00000000	Clear To Send Status Report
1	Node ID	00001100	00000000		11111111	CH0, CH8 Statuses ACK
1	Node ID	00001101	00000000		11111111	CH1,CH9 Statuses
1	Node ID	00001110	00000000		11111111	CH2,CH10 Statuses
1	Node ID	00001111	00000000		11111111	CH3,CH11 Statuses
1	Node ID	00010000	00000000		11111111	CH4,CH12 Statuses S
1	Node ID	00010001	00000000		11111111	CH5,CH13 Statuses
1	Node ID	00010010	00000000		11111111	CH6,CH14 Statuses
1	Node ID	00010011	00000000		11111111	CH7,CH15 Statuses
1	Node ID	00010100	00000000		11111111	EM_TX,EM_RX Statuses
1	Node ID	00010101	11111111		00000000	Send Node Status Request
1	Node ID	00010110	11111111		00000000	Monitor Channels Status Request
1	Node ID	00010111	Failed Node ID		11110000	Rescue Permission Is Granted
1	Node ID	00010111	Failed Node ID		00001111	Rescue Permission Is Denied
1	Node ID	00011000	11111111		00000000	Start EM Channel
1	Node ID	00011001	11111111		00000000	Stop EM Channel
1	Node ID	00011010	00000000		11111111	Blank ACK
			From Node's Local Controller			
1	Node ID	10000001	1010101		1010101	SET NODE ID ACK
1	Node ID	10001100	CH0 (TX)	CH8 (RX)	00000000	CH0,CH8 Statuses Ready
1	Node ID	10001101	CH1 (TX)	CH8 (RX)	00000000	CH1 ,CH9 Statuses Ready
1	Node ID	10001110	CH2 (TX)	CH8 (RX)	00000000	CH2,CH10 Statuses Ready
1	Node ID	10001111	CH3 (TX)	CH8 (RX)	00000000	CH3,CH11 Statuses Ready
1	Node ID	10010000	CH4 (TX)	CH8 (RX)	00000000	CH4,CH12 Statuses Ready
1	Node ID	10010001	CH5 (TX)	CH8 (RX)	00000000	CH5,CH13 STATUSES READY
1	Node ID	10010010	CH6 (TX)	CH8 (RX)	00000000	CH6,CH14 Statuses Ready
1	Node ID	10010011	CH7 (TX)	CH8 (RX)	00000000	CH7,CH15 Statuses Ready
1	Node ID	10010100	EM (TX)	EM (RX)	00000000	Emergency Channels Statuses
1	Node ID	10010101	Active Channels		Node Status	Send Node Status ACK
1	Node ID	10010110	Active Channels		00000000	Monitor Channels Status Ready
1	Node ID	10010111	Failed Node ID		Active Channels	Rescue Permission Request
1	Node ID	10011000	00001111		00000000	Up Cache Is Full

Table 3-3 Status / Command Words

Emergency Channel: Emergency Channel commands come from the Node local Controller, Channel Modem, and other Nodes Requests. It consists of one transmitter, and one receiver channels, similar to those used with the sonde (16 channels) but with 32 frames of 32 bits. They are part of the communication loop that goes over all channels (16 regular plus 2 emergencies) but, instead of 3 steps frame transition between caches and channels' FIFOs, the Emergency channels take 4 cycles, that is because they deal with the all the frame, including Node's number. When a frame doesn't carry the node's Id received in the downlink cache, it is routed to the emergency transmitter channel. Emergency channels don't only carry Acquisition frames, but also they carry control frames, when the rescuing node's main processor, sends control commands to the failed node's local controller. Table 3-3 shows the different commands formats. The two emergency channels work when the node is in "rescuing "mode , that is when it accepts the routed traffic from another failed node.

Emergency port: When the Node fails, all its main processor's traffic, either acquisition, from caches, or commands/acknowledgements to/from the local controller, are routed to another section in the emergency section, which is the Emergency port. The Emergency port examines incoming frames to assign them to either the local controller, or the caches, depending on the nature of the 32 bits frame received.

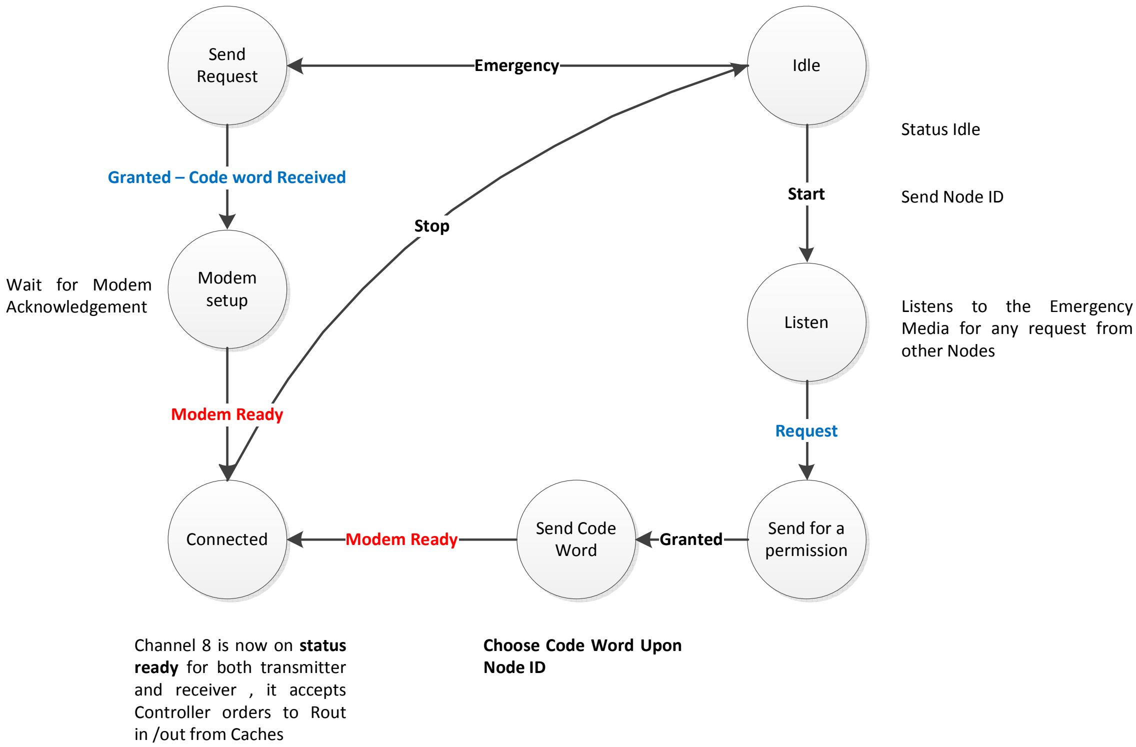


Figure 3-21 Emergency Channel – FSM

3.4 Higher level acquisition processes

Tasks achieved by Sonde-Interface FPGA, are issued, and reported to the acquisition process running on the main processor. It does the higher level traffic control tasks , and takes a decision to whether or not give help to a failed node , to be its rescuer node , depending on the input request from the failed node , that carries the number of active channels , and the feedback from other similar processes running on the other nodes.

3.5 Summary

In order to build a generic system, a configurable hardware is needed, and customization is driven in loadable software processes.

Multicore is built to run processes, or threads that are parallel enough to get the benefit of the independent cores.

Logging tasks can be rearranged into software processes that run in parallel. A communication mean between the processes is easily achieved through the shared memory space seen by processes running on the same Multicore.

Telemetry process is needed in only one node, to send collected data from all other nodes on the cable. Cable telemetry is done by using DMT modulation.

DMT based on sending data symbols on a group of carriers. Data symbols are generated by a constellation encoding process in order to decrease BER over the media.

The main node’s processor deal with the acquired data through a Sonde Interface FPGA that consists of communication channels that use FIFO to buffer acquired frames from and to sonde. Channels send and receive data frames from the Node’s caches, which send and receive data frames from the main processor directly.

Node’s local Controller manages data channels and emergency requests from other channels, and monitors the main processor in order to make sure that the node is alive.

When no respond from the main processor, Nodes broadcasting a distress request, other nodes receive the failed node’s request, send the request to the main processor.

Higher level acquisition process, which runs on the main processer, received emergency request from other nodes, analyses the situation, and consult other nodes to get the one which should respond to the emergency request. When the request is granted, node’s local controller sends to the emergency section in sonde interface FPGA, which respond by sending the Attached Modem an 8 bit code word , to send to the failed node in order to start receiving its frames , either acquisition or control frames.

CHAPTER 4

SIMULATIONS AND RESULTS

This chapter presents the implemented parts, in test scenarios that verify the proper responses of each part.

4.1 Telemetry Procedure results on a Prototyping Station

The Integrated process of DMT modem is responsible for cable modulation task , it takes place at every node of the tool , in startup stage , then one node is assigned to do it on behave of all other nodes in the tool string.

4.1.1 **Test scenario 1**: Telemetry process with Base band signal Parameters

The following example shows a Base band signal being modulated via the implemented Modem. All ones bit stream.

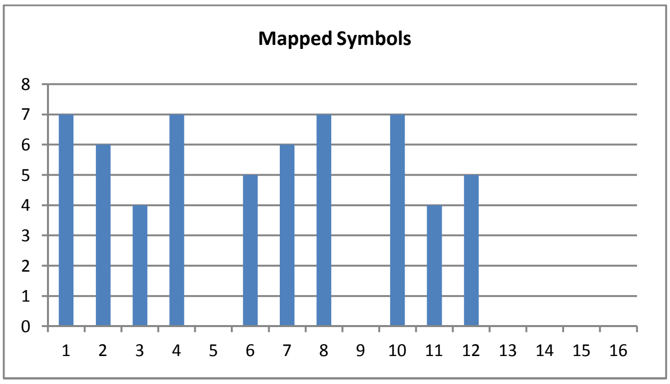
Channel Update:

Channel Height	: 4	Number of Symbols output of each sub channel
Channel Width	: 11	Number of bits input to each sub channel
Channel Number	: 4	Number of sub channels
Total Width	: 44	Total Input bits per one Sample Time
Division	: 3	Hz, between adjacent tones.

Tones are accepted with even numbers.
Additional Tone is added in the middle of the spectrum (here Frequency 0 Hz)

Tones (Hz):

Tone [0] = -24	Tone [8] = 0
Tone [1] = -21	Tone [9] = 3
Tone [2] = -18	Tone [11]= 9
Tone [3] = -15	Tone [12]= 12
Tone [4] = -12	Tone [13]= 15
Tone [5] = -9	Tone [14]= 18
Tone [6] = -6	Tone [15]= 21
Tone [7] = -3	Tone [16]= 24

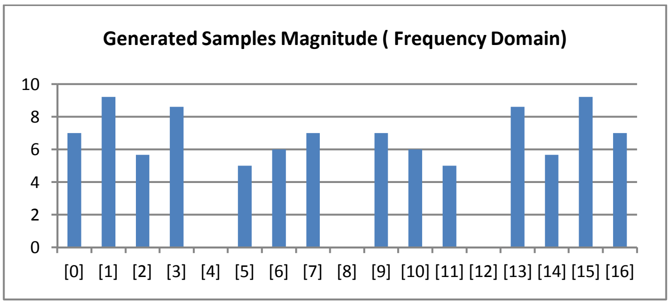


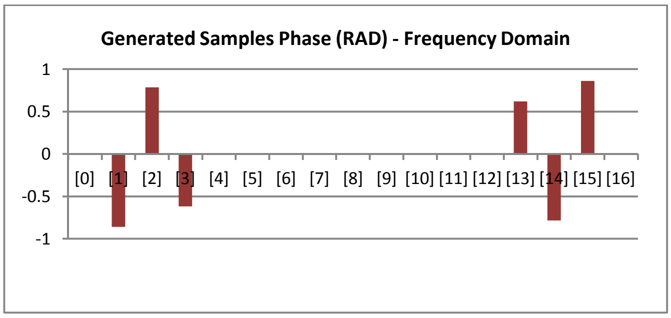
11111111011: Equals: 1791 Sub Channel 1 input
7
6 Mapped symbols of Sub channel 1
4
7
01000001110: Equals: 898 Sub Channel 2 input
0
5 Mapped symbols of sub channel 2
6
7

11000101110: Equals: 931 Sub Channel 3 input
0
7 Mapped symbols of sub channel 2
4
5
00000000000: Equals: 0 Sub Channel 4 input
0
0 Mapped symbols of sub channel 2
0
0

Before IDFT

Frequency
[f=0 KHz]= 7|_ 0
[f=0.003 KHz]= 9.21954|_-0.86217
[f=0.006 KHz]= 5.65685|_ 0.785398
[f=0.009 KHz]= 8.60233|_-0.62025
[f=0.012 KHz]= 0|_ 0
[f=0.015 KHz]= 5|_ 0
[f=0.018 KHz]= 6|_ 0
[f=0.021 KHz]= 7|_ 0
[f=0.024 KHz]= 0|_ 0
[f=0.027 KHz]= 7|_ 0
[f=0.03 KHz]= 6|_ 0
[f=0.033 KHz]= 5|_ 0
[f=0.036 KHz]= 0|_ 0
[f=0.039 KHz]= 8.60233|_ 0.62025
[f=0.042 KHz]= 5.65685|_-0.785398
[f=0.045 KHz]= 9.21954|_ 0.86217
[f=0.048 KHz]= 7|_ 0
[f=0.051 KHz]= 0|_ 0
[f=0.054 KHz]= 0|_ 0
[f=0.057 KHz]= 0|_ 0
[f=0.06 KHz]= 0|_ 0
[f=0.063 KHz]= 7|_ 0
[f=0.066 KHz]= 9.21954|_ 0.86217
[f=0.069 KHz]= 5.65685|_-0.785398
[f=0.072 KHz]= 8.60233|_ 0.62025
[f=0.075 KHz]= 0|_ 0
[f=0.078 KHz]= 5|_ 0
[f=0.081 KHz]= 6|_ 0
[f=0.084 KHz]= 7|_ 0
[f=0.087 KHz]= 0|_ 0
[f=0.09 KHz]= 7|_ 0
[f=0.093 KHz]= 6|_ 0
[f=0.096 KHz]= 5|_ 0
[f=0.099 KHz]= 0|_ 0
[f=0.102 KHz]= 8.60233|_-0.62025
[f=0.105 KHz]= 5.65685|_ 0.785398
[f=0.108 KHz]= 9.21954|_-0.86217
[f=0.111 KHz]= 7|_ 0



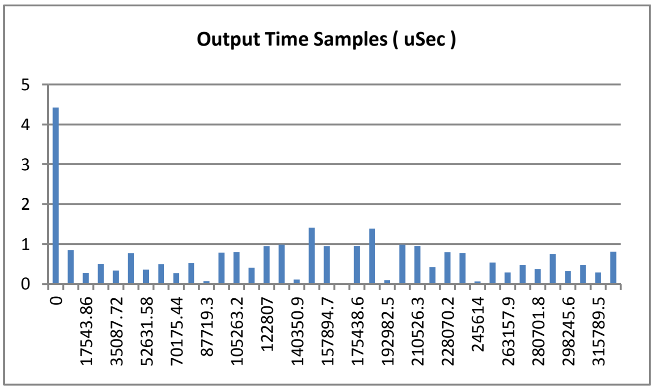


All Symbols / Tones:	After Conversion to Complex: Frequency	
	[0]=	7|_ 0
	[1]=	9.21954|_-0.86217
7	[2]=	5.65685|_ 0.785398
6	[3]=	8.60233|_-0.62025
4	[4]=	0|_ 0
7	[5]=	5|_ 0
0	[6]=	6|_ 0
5	[7]=	7|_ 0
6	[8]=	0|_ 0
7	[9]=	7|_ 0
0	[10]=	6|_ 0
7	[11]=	5|_ 0
4	[12]=	0|_ 0
5	[13]=	8.60233|_ 0.62025
0	[14]=	5.65685|_-.785398
0	[15]=	9.21954|_ 0.86217
0	[16]=	7|_ 0
0		

The Input Signal is padded with: Pins Before: 0
Padding Pins are added.

Time
[t=0.00 uSec]= 4.4210|_ 2.83832e-009
[t=8771.93 uSec]= 0.853|_-1.04379
[t=17543.86 uSec]= 0.277083|_-2.37694
[t=26315.79 uSec]= 0.504802|_-0.220268
[t=35087.72 uSec]= 0.332422|_-2.22007
[t=43859.65 uSec]= 0.771601|_-0.0663403
[t=52631.58 uSec]= 0.356552|_ 2.84371
[t=61403.51 uSec]= 0.496734|_-0.405636
[t=70175.44 uSec]= 0.267954|_ 2.97642
[t=78947.37 uSec]= 0.526694|_ 2.36132
[t=87719.30 uSec]= 0.0728533|_-2.75713
[t=96491.23 uSec]= 0.782839|_-1.1286
[t=105263.20 uSec]= 0.799388|_-0.0128673
[t=114035.10 uSec]= 0.409907|_ 2.35878
[t=122807.00 uSec]= 0.945464|_ 0.0477311
[t=131579.00 uSec]= 0.985799|_ 0.779887
[t=140350.90 uSec]= 0.108627|_-0.526286
[t=149122.80 uSec]= 1.40762|_ 0.369416
[t=157894.70 uSec]= 0.946854|_ 1.90519
[t=166666.70 uSec]= 0.0171804|_ 0.731967
[t=175438.60 uSec]= 0.954007|_-1.35987
[t=184210.50 uSec]= 1.38765|_-3.08311
[t=192982.50 uSec]= 0.0935568|_ 2.16896
[t=201754.40 uSec]= 0.98526|_-2.99397
[t=210526.30 uSec]= 0.952609|_ 2.39893
[t=219298.30 uSec]= 0.425854|_-1.80716
[t=228070.20 uSec]= 0.796949|_ 2.0275
[t=236842.10 uSec]= 0.774315|_ 0.733813
[t=245614.00 uSec]= 0.0608541|_-1.01707
[t=254386.00 uSec]= 0.53434|_-2.42778
[t=263157.90 uSec]= 0.28829|_-2.0219
[t=271929.80 uSec]= 0.478109|_ 0.800464
[t=280701.80 uSec]= 0.378245|_-2.44515
[t=289473.70 uSec]= 0.75372|_ 0.819912
[t=298245.60 uSec]= 0.327326|_-1.59153
[t=307017.50 uSec]= 0.48038|_ 0.337725
[t=315789.50 uSec]= 0.289228|_-2.13666
[t=324561.40 uSec]= 0.811127|_-0.904462

Inverse Fourier transform is completed
Time Spectrum is produced
Fundamental Frequency = 3 Hz
Sampling Frequency = 114 Hz
Sampling Period = 8771.93 us
Number of Time Samples = 38



The previous example is an algorithm test that modulates 44 input bits with carrier frequency 0, on 16 Tones within a Band width of 48 Hz, using 4 states Convolutional Encoder. That was the input to the SW module implemented. The mentioned parameters are not the real ones to produce the cable spectrum, but it shows the ability of using the same module with other parameters to suit other cables or transmission media, as well as to deal with Cable conditions, noise or interference , as at that level , the control section can modify the send/receive parameters.

4.1.2 **Test scenario 2**: Telemetry Process with Parameters used in Wireline

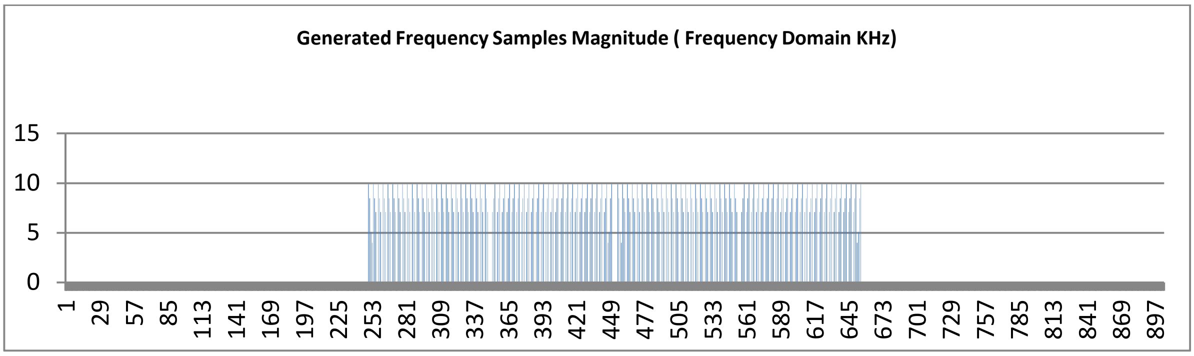
In this test the following parameters are fed the DMT Modem Module:

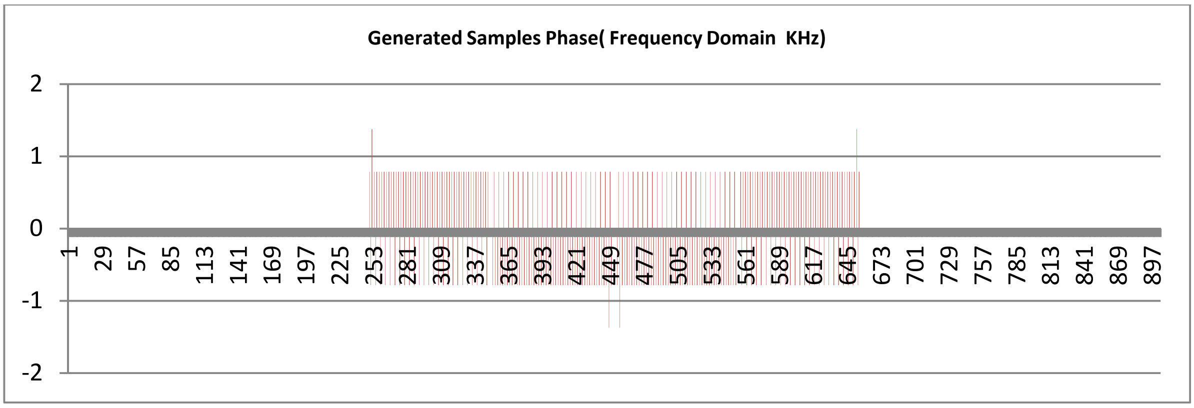
- All Ones input bit stream.
- 550 bit vector input.
- 200 Tones.
- 200 KHz Bandwidth.
- 250 KHz Carrier.
- 4 States Convolutional Encoder.
- 4 Dimensions 12 bit QAM Constellation.

As number of samples is large in is test, they won't be printed; only spectrums and time samples diagrams are presented.

Channel Update

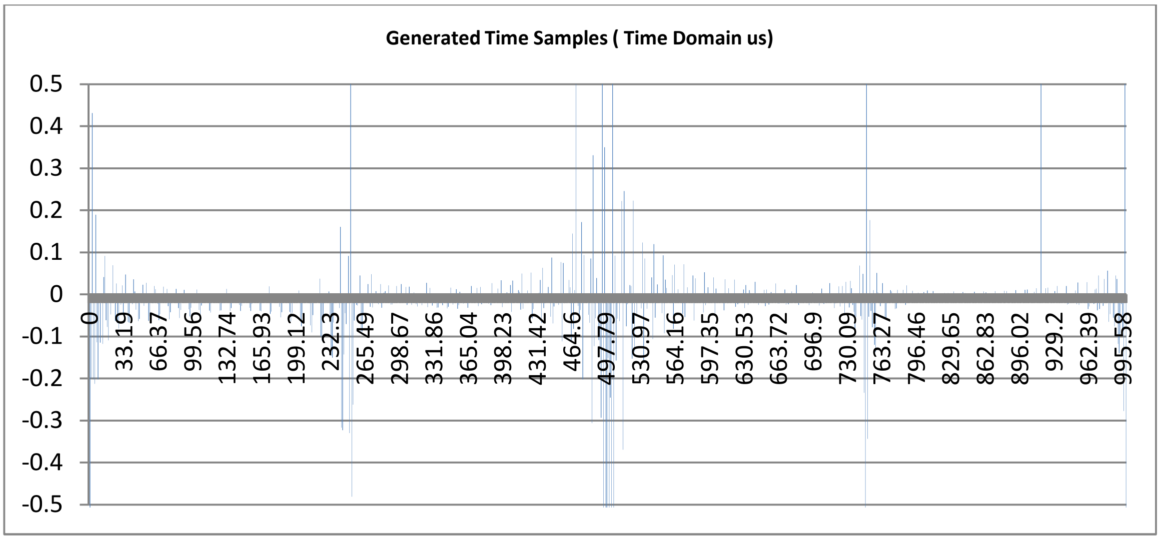
Channel Height	: 4
Channel Width	: 11
Channel Number	: 50
Total Width	: 550
Division	: 1000





Inverse Fourier transform is completed
Time Spectrum is produced

Fundamental Frequency	= 1000 Hz
Sampling Frequency	= 904000 Hz
Sampling Period	= 1.106195 us
Number of Time Samples	= 904



Conclusion

Building DMT Module that smoothly configured gives the benefit of easily responding to transmission media conditions, widens the range of used cables types, and uses the same modulation with different bands and carriers which could be implemented inside the tool string on the local communication path. Digital modulation performed by this module would need only gain controlled digital to analog converter and time switch with settable switching time interval that makes the analog HW simpler.

4.2 Acquisition FPGAs’ Simulation Results

This part shows the simulation results of the Node to Sonde Interface HW, it is as mention in chapter 3, is chosen to be implemented by an FPGA. Figure 4-1 shows a block diagram, showing the main parts and buses widths between them. Chapter 3 discusses the state diagrams and codes used in intercommunications between parts.

The simulation is done By ALTERA-ModelSim. Tests are presented in Test Scenarios. Chapter 3 shows the function of the synthesized FPGA, which is to carry the acquired data from Sonde to the main processing unit, and to sense and switch traffic in case of emergency. Three test scenarios are presented here. The first is in case of normal operation, the second, is when traffic from a failed node is routed to the node under test, and the third one, when the node itself fails.

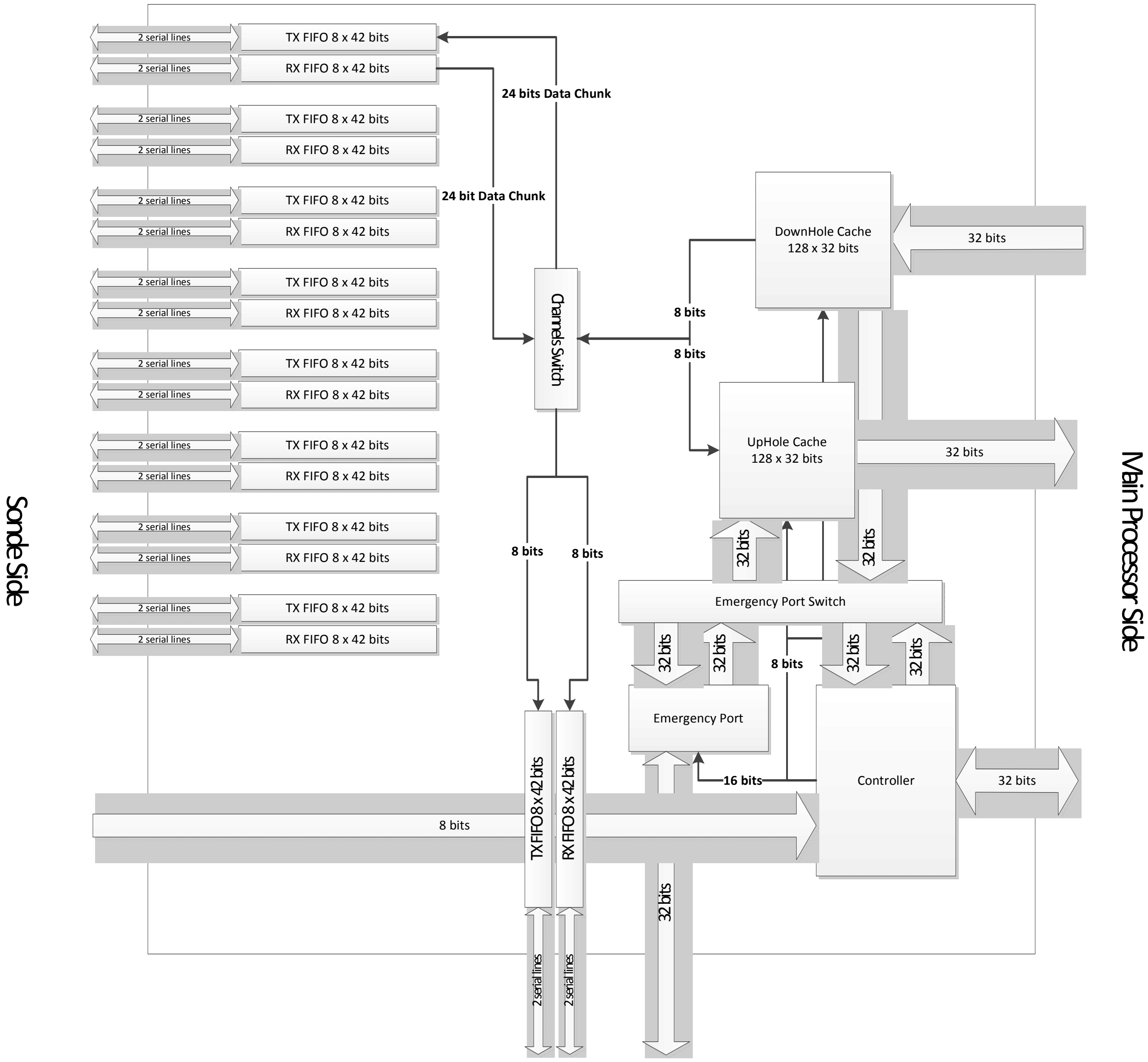


Figure 4-1 Node-Sonde Interface FPGA

4.2.1 **Test Scenario 1:** Normal Operation

A Sonde Interface node, connected to node's microprocessor, and to a sonde part. The node is in normal operation. Main processor and sonde part will exchange data, and commands. Sonde channels are configured as follows:

- 4 Acquisition Channels , 2 command channels

  Channel 1: **Acquisition**. Faces **Ch0RX** in Node-Sonde Interface FPGA.

  Channel 2: **Command**. Faces **Ch0TX** in Node-Sonde Interface FPGA.

Channel 3: **Acquisition**. Faces **Ch4RX** in Node-Sonde Interface FPGA.

Channel 4: **Acquisition**. Faces **Ch5RX** in Node-Sonde Interface FPGA.

Channel 5: Command. Faces **Ch5TX** in Node-Sonde Interface FPGA.

Channel 6: Acquisition. Faces **Ch7RX** in Node-Sonde Interface FPGA.

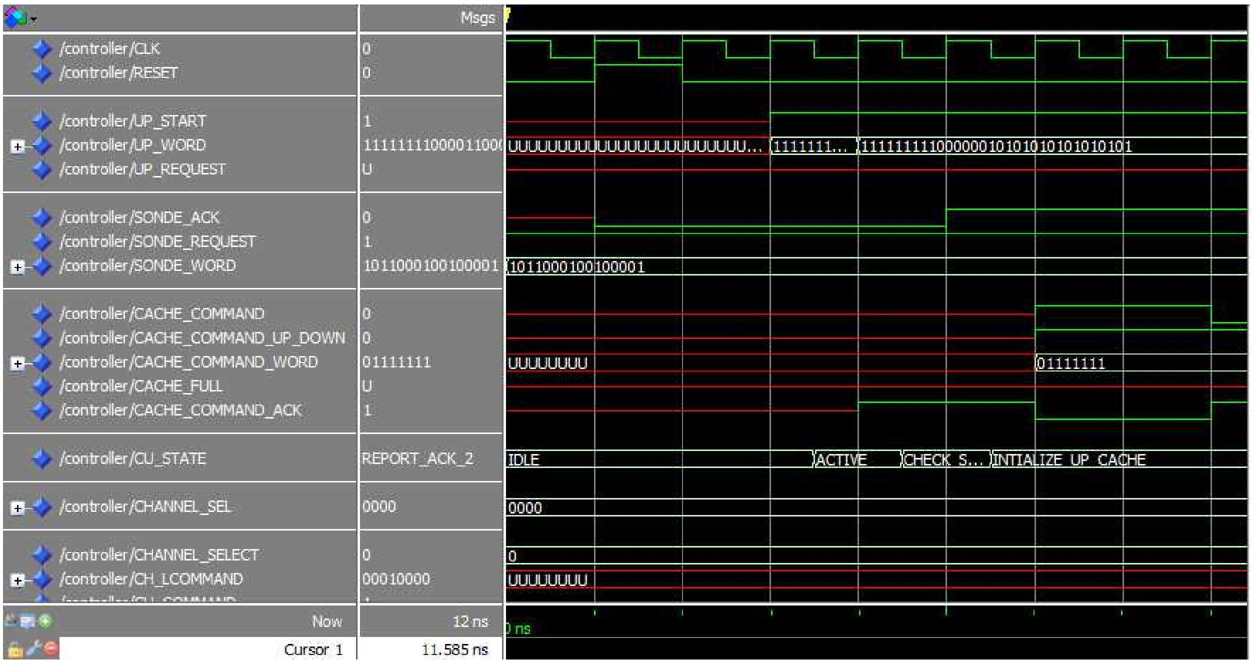
- Channels carry random sequences of bits.

Step 1: Initialization

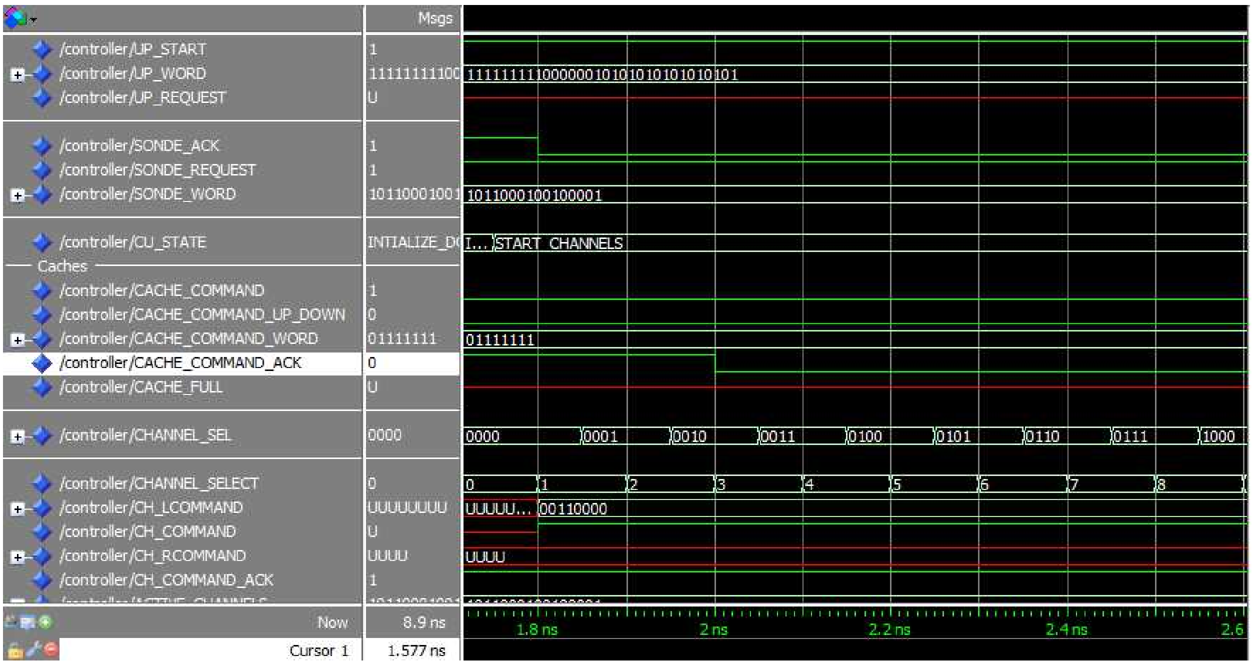
1. The node controller gets start command from the main microprocessor.
2. The main processor sets up Node ID with the local Controller.
3. The local controller listens to Sonde's controller word that carries the working channels, as each channels TX, and RX are considered two separate channels with different IDs. Upon channel's ID it could receive dedicated traffic, and stamp its own. Stamping traffic packets is done at sonde side (not presented here) as it is the one assigns channels. Channels IDs are kept the same even if the node's main processor fails. Still traffic is being routed upon to the same channels, as will be discussed later in the emergency node scenario.
4. Channels report is prepared and the main processor is notified.
5. The main processor asks for channels reports by acknowledging the previous. It can ask any time after this for any certain channel's status separately.

The following as simulation results, showing the local controllers traffic and command/reply exchanges between the Node's main processor and itself, corresponding to the previous steps.

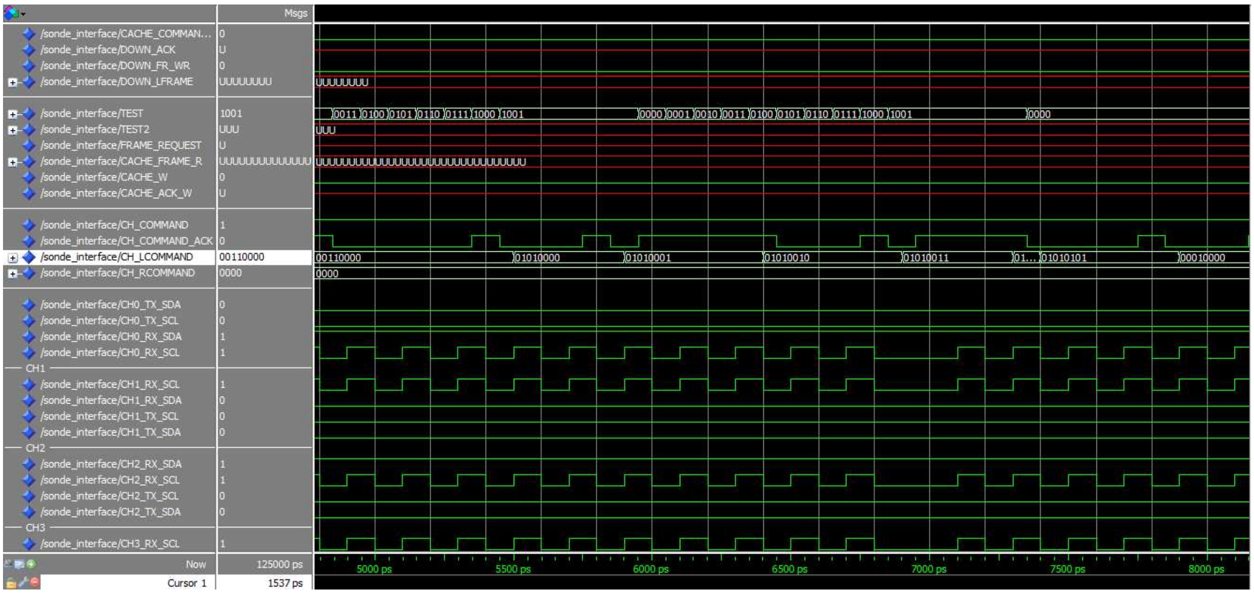
Simulation results:



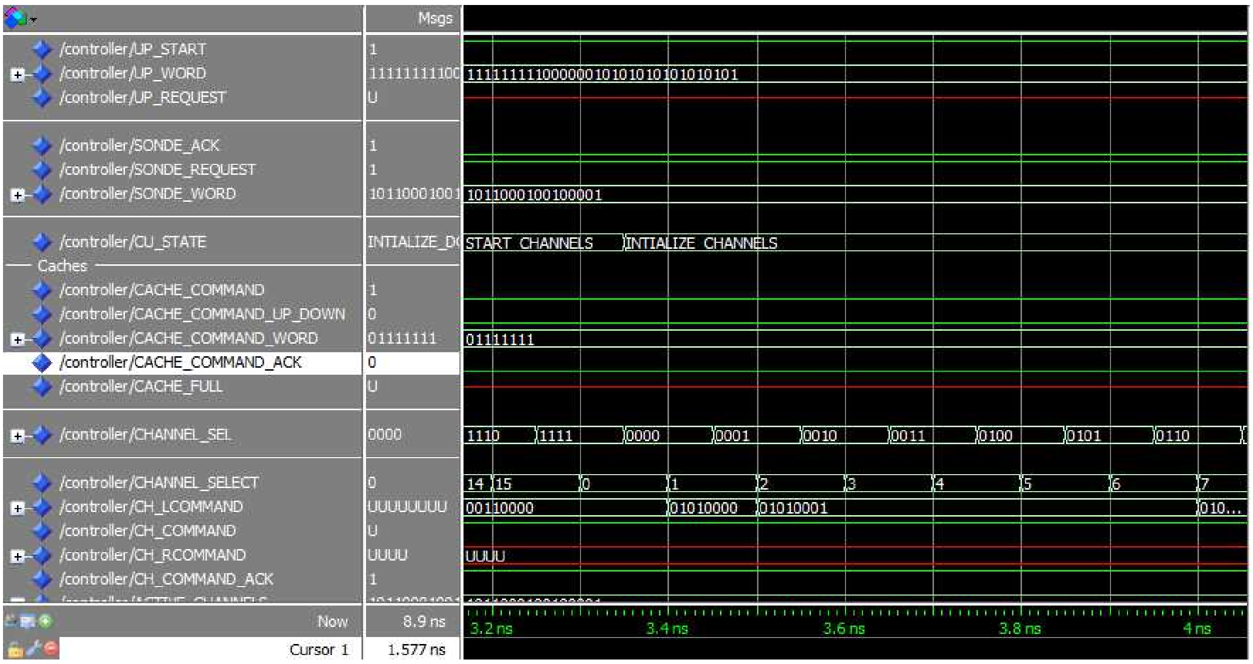
Step 1- Initialization: Clock is 100 ps, with 50% duty cycle. Simulation starts at 0 ps, and ends at 0.83 ns (items 1, 2 and 3 of Step 1)



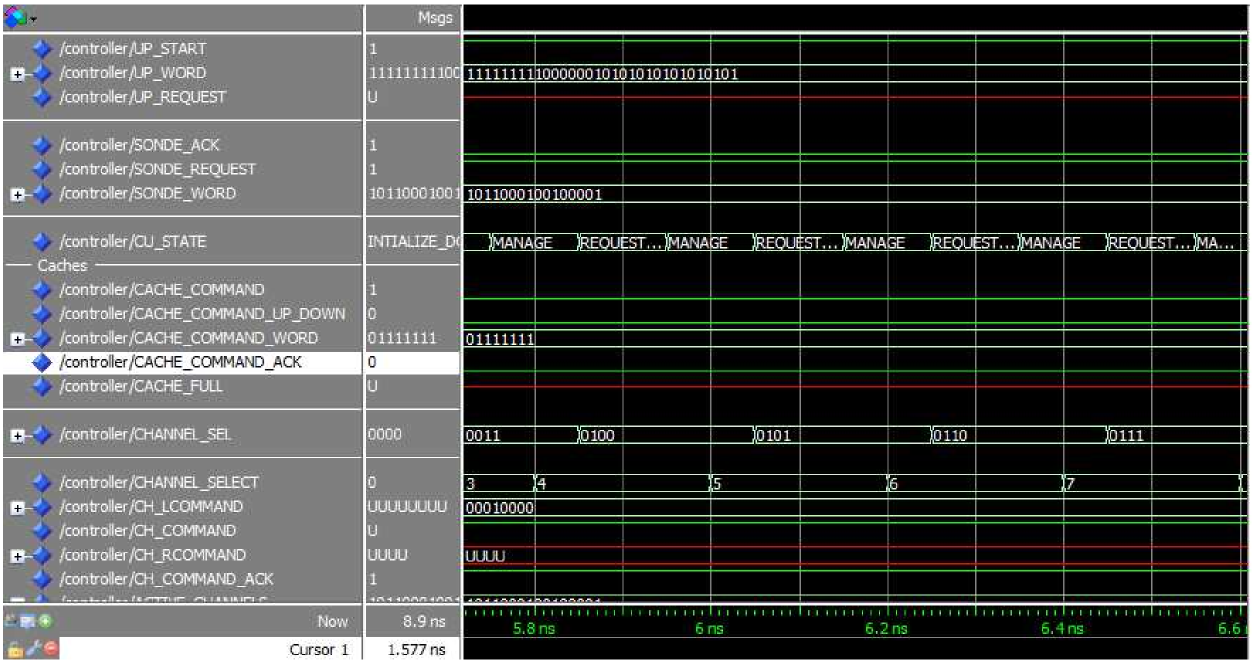
Step 1-3 (Sonde Active Channels are requested) – Initialization



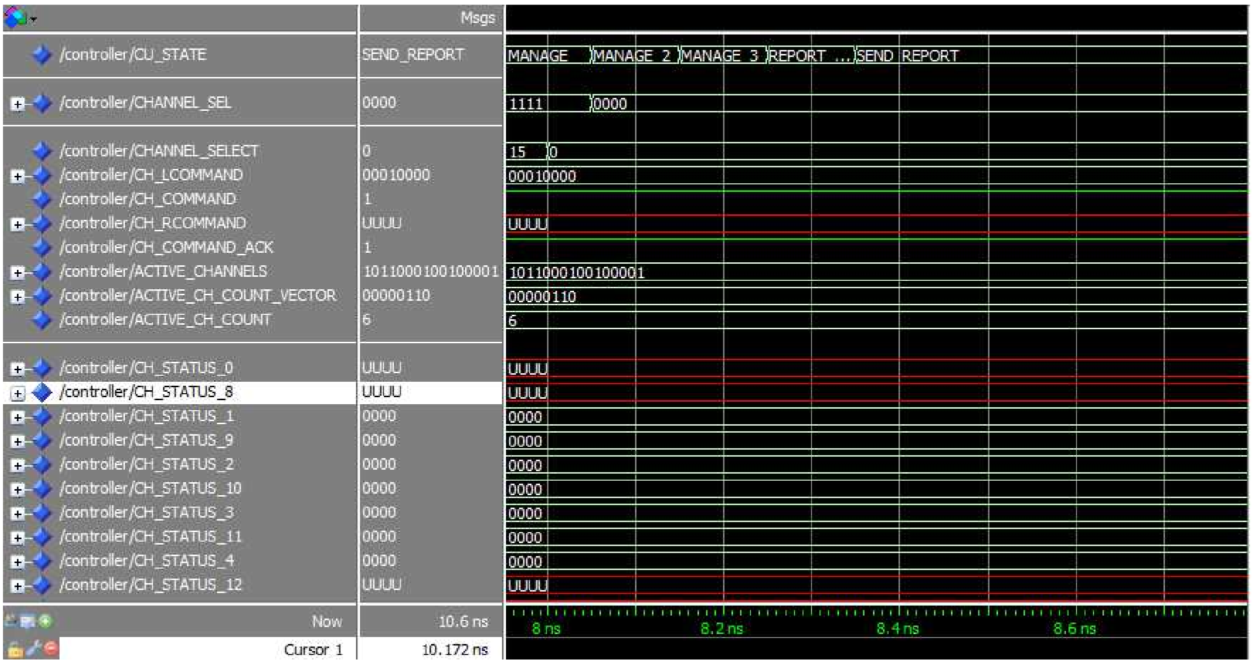
Step 1-3 (only Sonde Active channels are initialized)



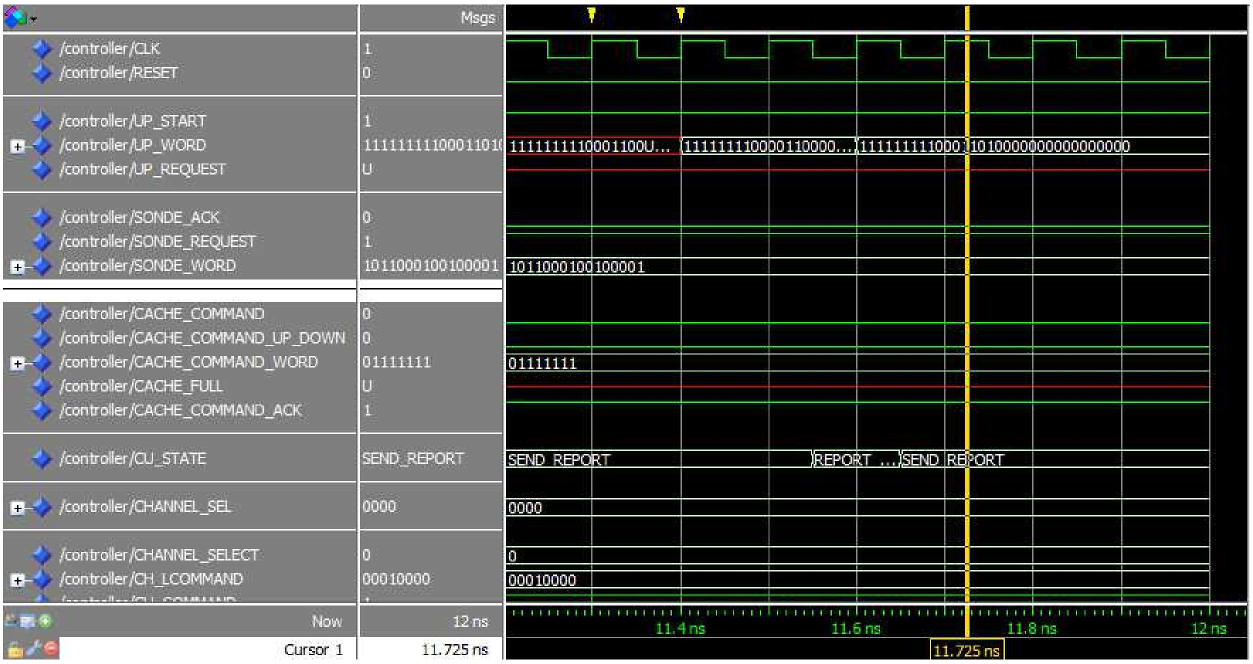
Third task in step one, as mentioned in Step 1- Initialization (continued)



Fourth task in step one, as mentioned in Step 1- Initialization (Collecting channels status reports)



Fourth task in step one, as mentioned in Step 1- Initialization (continued)



Fifth task in step one, as mentioned in Step 1- Initialization.

Step 2: Traffic

1. All channels start, only channels that sonde section had declared in the previous section will be working.
2. The main processor sends commands, or delivers surface commands to the node. Command frames are taken by Downhole cache.
3. The main processor asks to read what had been acquired. Acquired frames are stored in Uphole cache. The control over caches is not done by local controller. When the caches are full it refuse new traffic, and so channels queues gets full and stops sonde acquisition. The local controller is notified by the cache and so it send to the main processor. No further action is taken on this level.
4. Acquisition channels, detects start of frame symbol on its serial lines, starts accumulating a frame, then store it in its FIFO queue, when a frame is asked for by the up cache, it is sent on three acknowledged steps of 8 bits each, except for emergency channel which send, which sends the frame in 4 steps, as it sends the whole 32 bits frame, as shown in the next test scenario. Each channel, including emergency channels takes a share to send their frames (one frame at a time) to the Uphole cache. All channels pause for a while (put as 4 clock cycles) , during which , the main processor reads from Uphole cache

5. Command channels receive frames of 24 bits from the downhole cache. According to its channel Id, the sent frames, which are stamped with the same channel Id in the second half of the first 8 bits, are accepted by the corresponding channels. Frames are passed in 3 chunks of 8 bits. They are stored in the transmitter's FIFO. Figure 4-1 shows the sizes of FIFO queues, and caches.

Test parameters:

- Node ID: 1111111.
- Main processor input is repeated frames on the form:

01111111 00001110 00001110 00001110

- The frames are sent to channel 0 TX, which channel ID, is set to 0000 (Step 1-3).
- Downhole cache receives the sent frames , and rout them to the designated channel to be saved as data chunks on the form:

00001110 00001110 00001110

- Receiver channels are receiving a sequence of 1s, all 0s and. which is to be sent to Uphole cache as frames on the form:

01111111 11111111 11111111 11111111

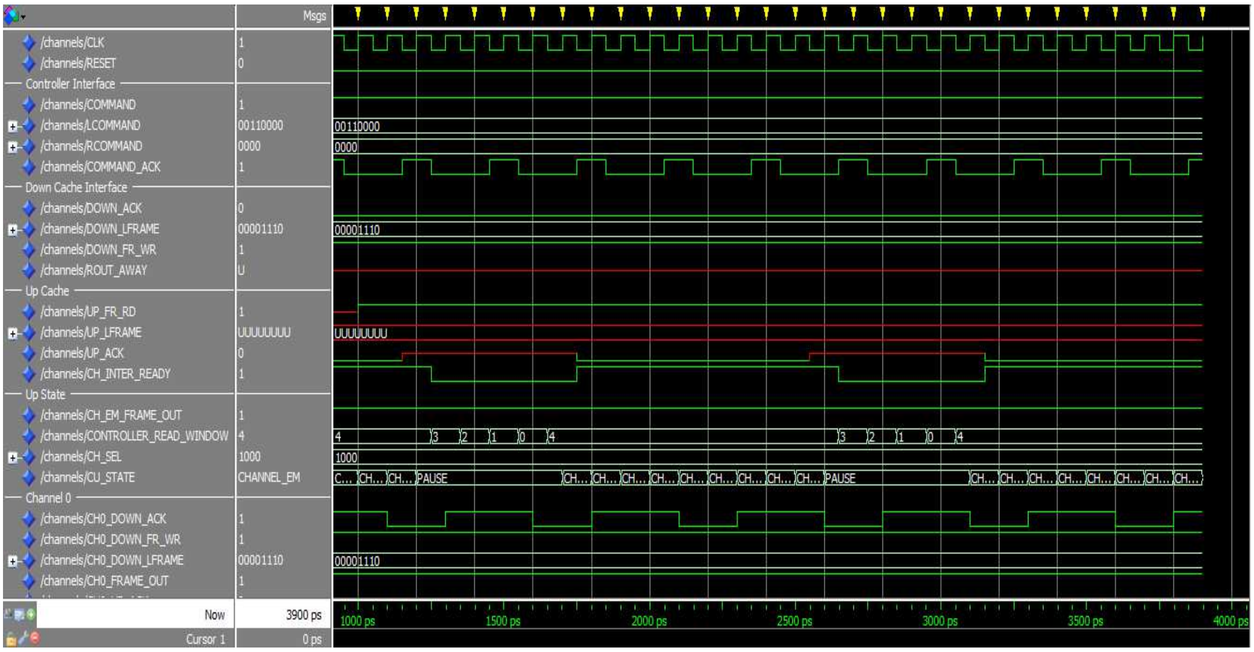
Simulation results:

```
00000000 |111111111111111111111111 111111111111111111111111 111111111111111111111111 111111111111111111111111
00000004 |UUUUUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUU
```

Channel 0 RX (CH8) FIFO after 27.5 ns

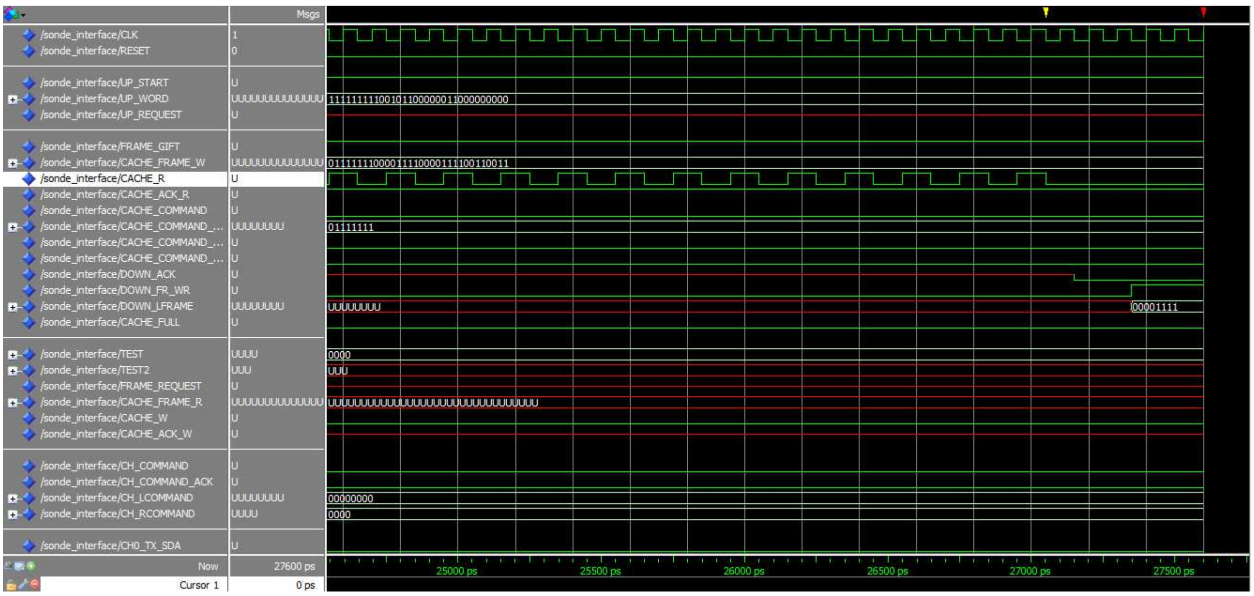
```
00000000 |000000000000000000000000 000000000000000000000000 000000000000000000000000 000000000000000000000000
00000004 |UUUUUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUU
```

Channel 1 RX (CH9) FIFO after 27.5 ns



Step 2-4 Traffic

In the shown waveforms, the down cache has been loaded with frame send by the main processor, and is now sending chunks of 8 bits (DOWN_LFRAME) to all channel transmitters, only channel zero responds as the its id (set is Step 1) matches the second half of Byte number 2, which is the first to ne sent in the chunk (00001110). In 27.5 ns the downhole cache is full and no longer receive frames (no acknowledgements CACHE_R) are sent to the main Processor, as shown in the next shot. As the cache is full, it ignores further request form the main processor, and start sending chunks to the transmitters.

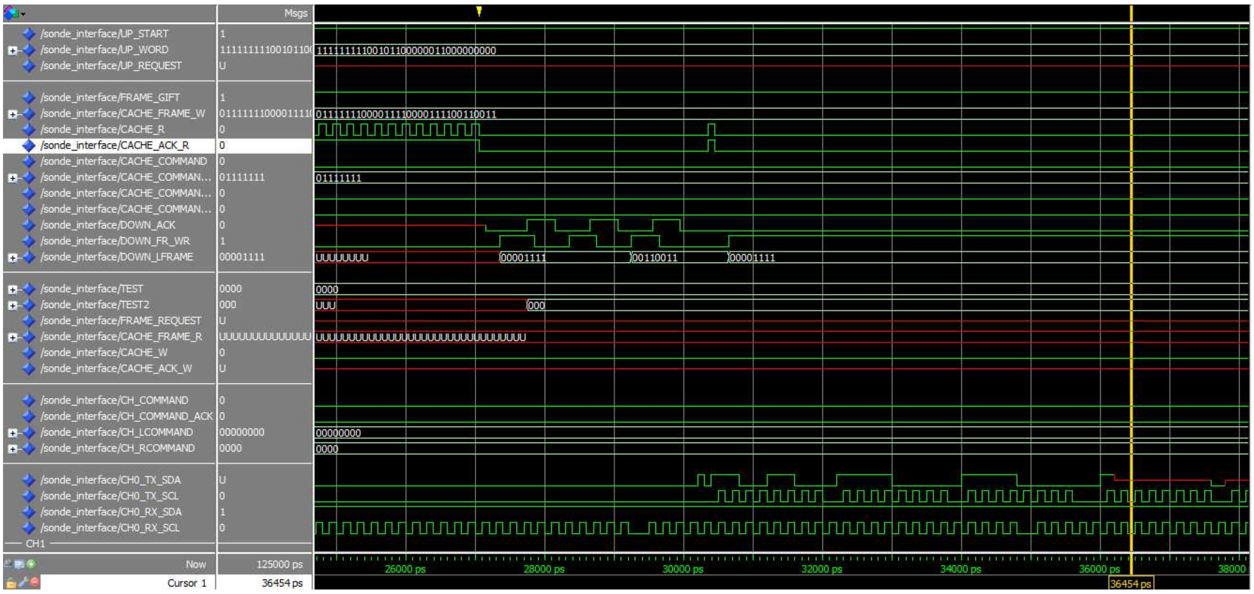


Step 2-5 Traffic

```
0000004c  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000049  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000046  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000043  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000040  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
0000003d  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
0000003a  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000037  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000034  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000031  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
0000002e  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
0000002b  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000028  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000025  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000022  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
0000001f  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
0000001c  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000019  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000016  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000013  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000010  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
0000000d  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
0000000a  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000007  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
00000004  01111111000011110000111100110011 01111111000011110000111100110011 01111111000011110000111100110011
```

Downhole Cache after 27.5 ns

Ch0 TX receives a frame, when done (at 30644 ps), it starts sends it on its serial lines. The sequence starts with Start symbol, then 8 bits at a time, separated by acknowledgements from the receiver side. When a frame is released from the downhole cache, it is now longer full, as so it will accept another frame from the main processor as shown with the Acknowledged frame pulse at 30350 ps. Shown below.



Step 2-5 Traffic

```
00000000  000011110000111100110011 UUUUUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUU
00000004  UUUUUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUU
```

CH0 TX FIFO at 30644 ps

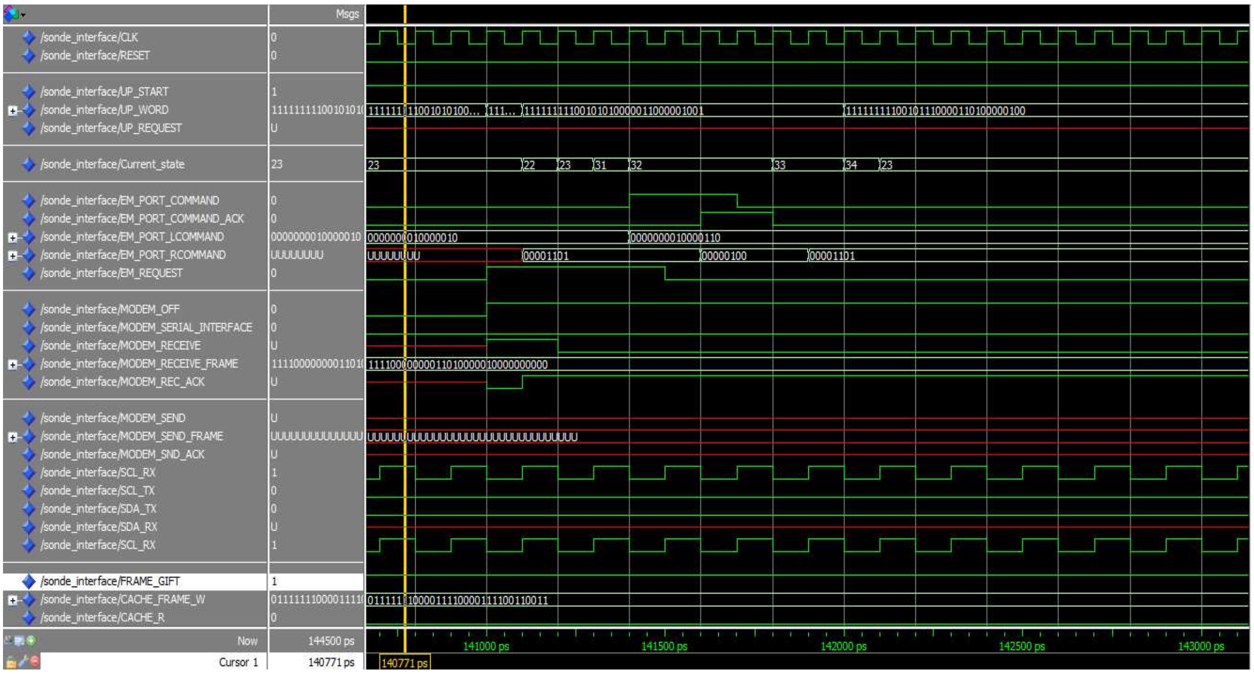
4.2.2 **Test Scenario 2:** Emergency Request from Other Node

A Sonde Interface node, connected to node's microprocessor, and to a sonde part. The node is in normal operation. Main processor and sonde part will exchange data, and commands. Sonde channels are configured as in Test Scenario 1.

- A Node with ID 0001101 fails (Main processor section), it send requests all over the media.
- The modem at Node 1111111 receives a request , and report it as a frame to the emergency section of the node , on the form of a frame as (distress code 11110000 , 0, Requester Node ID, Requester Node's Active channels (4) , 00000000) :

  11110000 0 0001101 00000100 00000000

The following are the simulation results verifying the desired response (correct report to the main processor)



Test Scenario 2, Node fails

- Emergency request is sensed at 141 ns.
- The request is reported by the modem to the emergency port. The report contains the requester Node ID.

- When the local Controller receives the request, it prompts for the number of active channels in the requester node. That happens at 141600 ps.
- The node's local controller reports to the main processor in the form listed in table 3-3. Which is here :

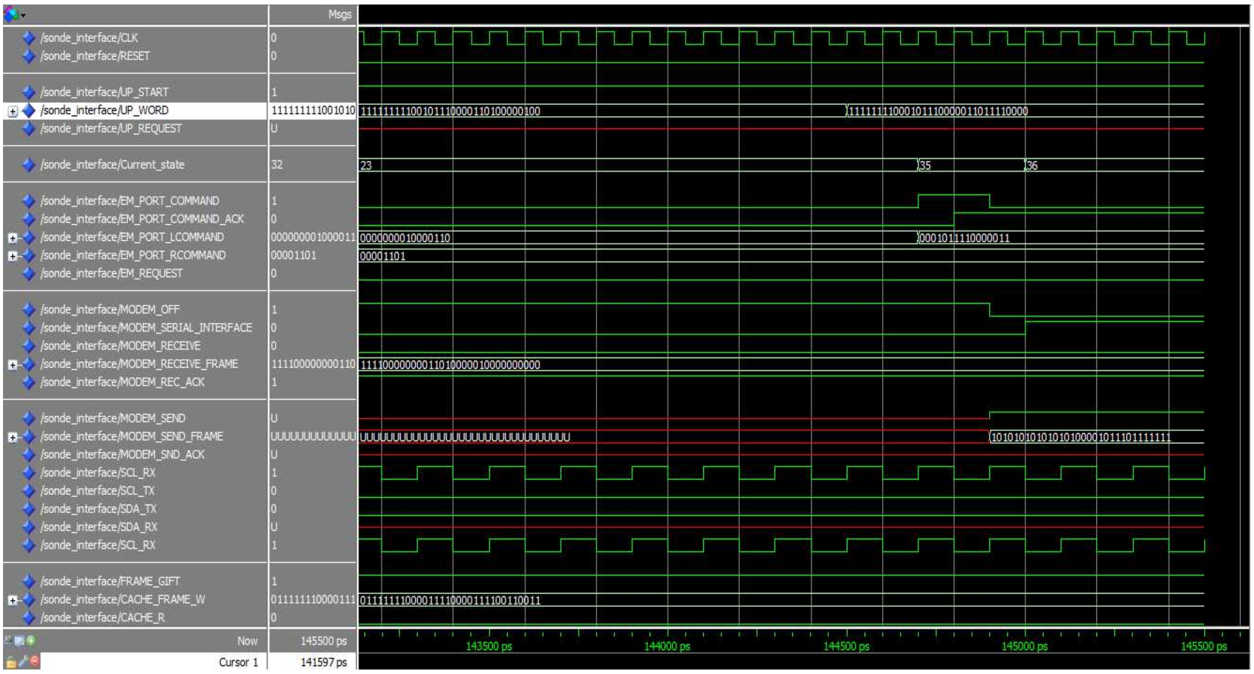
1 1111111 10010101 00000110 00001001

(1, Node ID, 10010101, requester Node ID, requester number of active Channels)

- The Main processor answers with a “Permission Granted” on the form in table 3-3 , which would be :

1 1111111 00010111 00000110 11110000

- When the Local Controller receives “permission granted” from the main Processor, it sends to the emergency port, with the code word part (8 bits) which is a combined with Node ID in the Emergency port, and sent via the modem to be picked by the failed node, and now the Emergency channels are ready to receive the failed node's traffic.



Emergency Permission Granted

4.2.3 **Test Scenario 3**: Node Emergency Mode

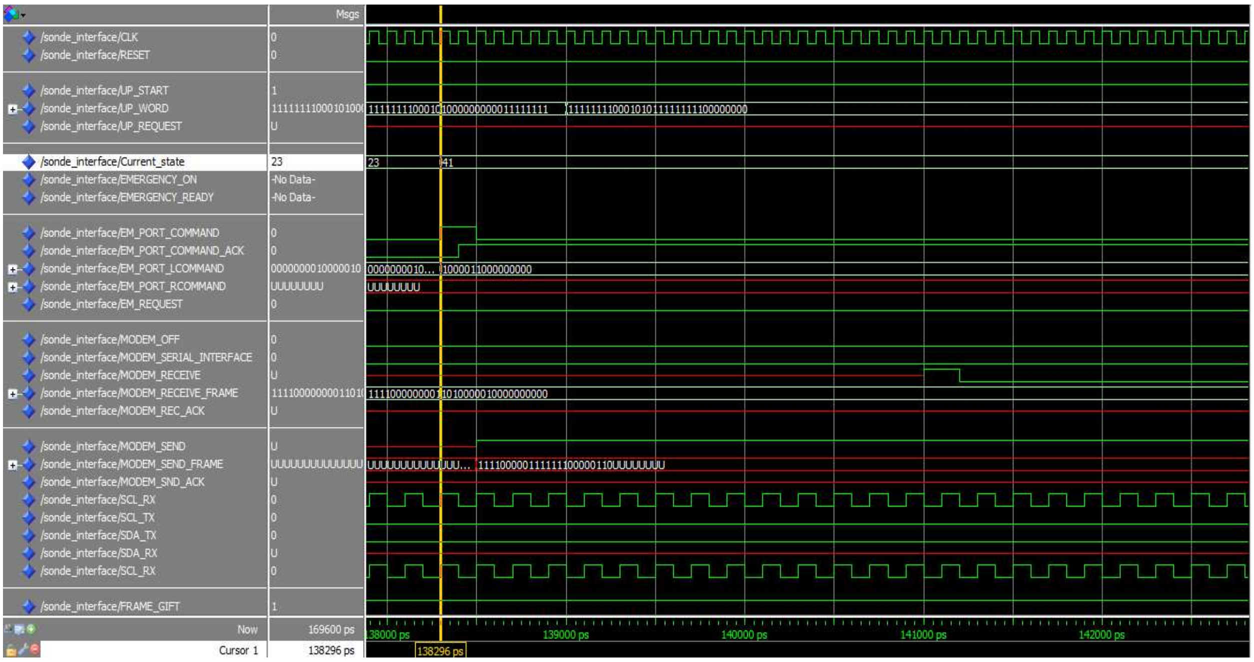
In this scenario, the Node fails. As neither response from the main processor, nor any command, or acknowledge for 120 Clock cycles.

- Local controller goes into Node Fail Mode, and sends to the Emergency port an order to sound the alarm, and broadcast distress signal. That happens at 138269 ps of the start of the simulation.
- Emergency port composes a distress frame and sends it via the modem to all nodes listening.

The distress frame is similar to the one received in test scenario 2, which is now:

11110000 0 1111111 00000110 00000000

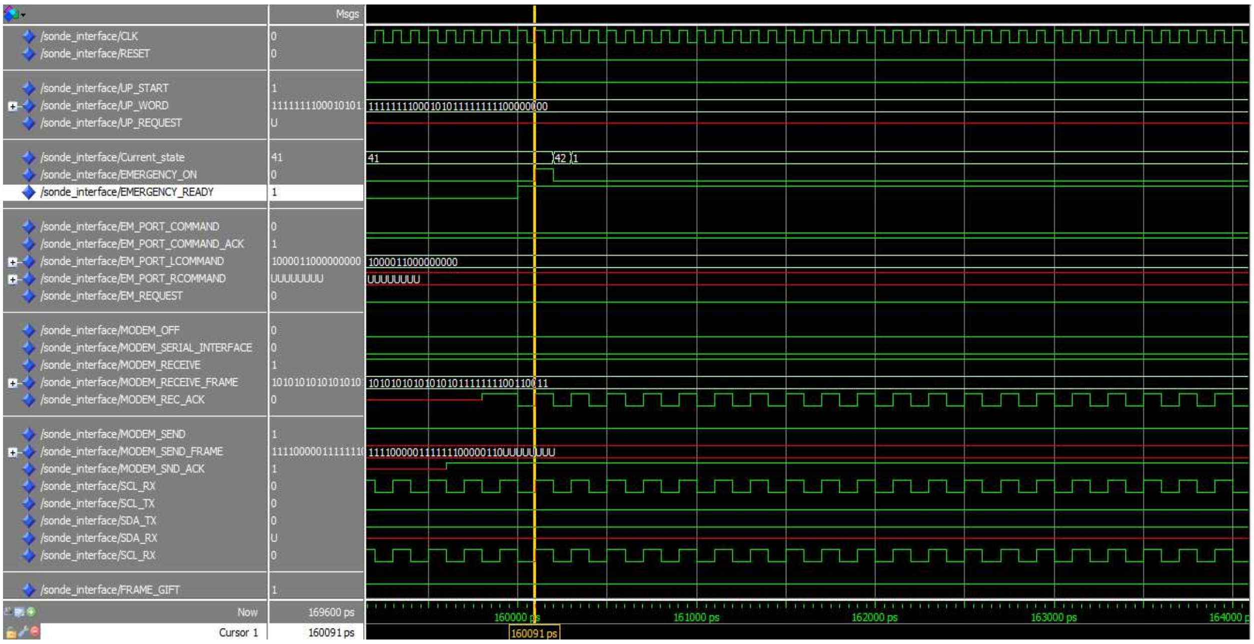
(11110000, 0, Requester Node Id- this node now, number of active channels, any 8 bits sequence)



Node fails and sends a distress signal

- A rescuing node responds by sending a reply frame through the modem with the “Code Word”, which is used by the modem to establish the link.
- The failed node, received the frame via its modem, then rely it to the emergency port, which in turn informs the local controller.

- The local controller now, initializes “Emergency on Signal “so that all other traffic from and to caches, and the local controller as well is routed to the modem Emergency port.



Node in Emergency Mode

The past three test scenarios are to verify the main function achieved by Sonde Interface FPGA.

Many functions haven’t been presented in the three scenarios discussed.

4.3 Summary

The telemetry process can work in various bands with adjustable parameters give the benefit of using more than one media, or cable, type, and can get the best use of the cable.

Cable telemetry proposed process is tested here in doing two differ modes of operation, a base band, and band pass. The output spectrums show the expected frequency range required. They are presented to verify the proper response of the synthesized process in two different conditions.

Sonde Interface section is expressed as an FPGA that interfaces Sonde section, Emergency channels, and the Main processor in each node.

Three test scenarios are presented here, one for the normal operation of a node. The second, when a node provides help to another failing node. The third is when the node itself fails, broadcast an emergency request, and, we accepted, it routs its traffic to the rescuing node.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

This last chapter summaries the previous four chapters, highlights the conclusions, problems, and suggestions for future work.

1.1 Research Summary

Petrophysics is a branch of science that deals with the physical properties of the ground that is related to producing hydrocarbons stored in earth's reservoirs. It started at the beginning of the last century, in France. Petroleum industry relays on Petrophysics in taking engineering decisions, that determines the way to maximize the profit of a well, or a reservoir.

Another related industry that takes over the mission of recording, and monitoring oil wells, in order to gather the desired physical properties of a section in ground formation is well logging.

During the "logging job" in which a section in an oil, or gas well, is monitored, or "logged", special equipment on the surface, and another inside the well, or "downhole" are used. The section of oil field companies responsible for logging is using cables to connect surface equipment, with downhole tools.

Well environment is considered one of the most hostile environments for electronic equipment to work in, because of the high temperature, high pressure, humidity, and corrosive gases that causes unlikely failures in another less hostile environment.

The hard environment causes many simple, or complicated failures in electronic sections responsible for gathering data, digitizing it, do preprocessing and DSP primitive analysis, and send it uphole.

This failures leads to loosing valuable time during the logging job, that causes high financial loss to either the logger or the owner of the well.

Downhole tools are used to measure physical quantities, in the ground formation inside the well, and other measurements to localize the obtained ground measurements, such as position with respect to earth's coordinates, temperature, that is used to correct the obtained measurements via compensating temperature effects of the measuring sensors, and surrounding fluids electrical, or mechanical properties, by with the induced electrical, or mechanical energy of measurement will be affected. In general all acquired data is sent in a digital form, gathered as each tool, which is responsible for measuring certain physical quantity. All processed data from all tools are collected by a telemetry protocol, or a bus protocol, like CAN or EFTP for example. The collected packets are sent by another dedicated telemetry tool over the telemetry cable to the surface. Many protocols are used to send data over the logging cable, one of the most wide spread is DMT.

This study aims to propose a solution to the reliability problem, by which many downhole failures happen, also to provide a way to reduce the maintenance cost of the expensive downhole tools, by changing the high specific purposes tools, into a generic modules that could be exchanged between many different purpose tools, and to confine the customization it to the "Sonde" section, the section carries the application specific sensors, or mechanical gear.

Chapter 3 proposes the solution for this problem by rearranging tasks done in downhole tool string, into sonde section, as mentioned, the physical part responsible for inducing and measure the reflected energy in the formation, a generic modules that do repeated tasks in all tools, and a software section that performs the custom processing on the acquired data according to its nature.

Sonde Section, is where analog measurements are done, digitized and multiplexed as needed, then sent as a row data, on some channels with variable range of baud rate according to how fast or slow the sensors are.

Sonde Interface Section is a Hardware that is command to all tools, contains fixed number of configurable channels that is connected to their counterparts in the sonde section.

Node's Main processor, which is common to all nodes, with the same peripherals, performs preprocessing tasks required by each measurement type.

By getting the previous sections done, the tool string is turned into a network of processing nodes with common peripherals that required a communication mean between each other. That leads to a modular design that uses repeated hardware, and flexible enough to change the task each node does, by loading the proper software routines corresponding to either processing the acquired data, or the communication protocol used to connect the processing nodes together.

By having a processing node that can do more than one separate task evenly, multifunction's downhole tools will be able to be performed by one processing node, not only this , but also another nodes' tasks could be hosted by a capable processing node. in case of a node fail due to an emergency condition happens downhole , that emergency condition , could be as simple as a solder joint , or as complex as a full microprocessor failure , either failure have the same result which is a costly loss time, another processing node can take over its tasks.

So, depending on the unlikely happening condition of all sections fail, the proposed solution offers redundancy sufficient for saving an emergency condition, by:

- Using two Sonde Interface hardware modules in each node, such that one is a backup, choosing between them is the responsibility of the node's main processor.
- Using Sonde Interface hardware in routing traffic to another node, in case of its Node's main processor fails.

Chapter 3 continues with further syntheses of the proposed solution, it starts behaviorally, in which the software processes takes place. According to the nature of the software processes that should run on a single processing node, an how parallel they are, a Multi-core processor is the proposed solution at this point, as it offers both the capability to run parallel tasks , and the ease of communication between the running parallel processes.

In the behavior description of the proposed design, the running parallel processes are presented; one of these processes is the telemetry. Chapter 3 proposes a way to synthesize the telemetry process. The telemetry process is divided into the upper layers protocol, and cable telemetry which is the physical layer, Chapter 3 discusses cable telemetry as a software process, to be run on the Nodes main processor, as this point is meant to be a Multi-core microprocessor.

Chapter 3 also describes sonde interface hardware. It is functionally descried, then on RTL level, and then it is hosted in an FPGA.

The built cable telemetry process, and sonde Interface FPGA, testing results are presented in Chapter 4.

1.2 Conclusion

The proposed solution here gets the advantage of advanced computation capabilities of a Multi-core microprocessor, in an embedded system, to enhance the reliability though providing redundancy in hardware, and flexibility of applying wide range of acquisition, DSP, and communication protocols, in almost no additional cost. And reduce the maintenance cost of downhole logging tools, though providing generic modular sections.

The idea of transforming logging tasks, done by dedicated Hardware into a software processes provide the mentioned flexibility, as seen by the converting Cable telemetry into a process , many control tasks , wide range of operation , and so wide range of communication medias can be used.

The generic sonde interface FPGA, offers a solution to interface the Main Multi-core to a customizable Sonde part.

Rearranging the functions in a logging tool string, results in gaining higher performance, more reliable, and modular that can last longer, get easily updated, and require less time and cost maintenance.

1.3 Future Work

Many sides of the proposed design hasn't been covered here, one the RTL level, design parameters like the time a node considered idle after no response in, cycles that the Sonde interface prepares its report are chosen here, but they could be better adjusted when the complete system, such as the type and speed of the used Multi-core, the fastest response of the connected Sondes are determined.

Multi-core is introduced here as a mean of running parallel processes. It can also, run more advanced DSP, and data manipulations downhole, than currently applied, by redesigning the software processes in parallel threads. Increase speed allows faster logging speed, and so less job time.

Using Multi-core downhole opens another trend in building downhole tools, many problems that face the logging jobs, such as a stuck condition, could be handled by smarter downhole tools , embedded robot arms , can be used to help solving such case.

Advanced sensors and techniques, such as magnetic resonance logging tools, require much of computation power, and currently built in large long and heavy electronic sections. Using as much software in performing the required processes can reduce the size of electronic section s, as well as use the merit of general modular sections in facing failure situations, with no additional Hardware. That application as well is in continuous development firmware. That also can be done easily via the proposed design in this study.

Many problems faces the completeness of the proposed design, one of them is the high temperature downhole. So, circuit level design is facing many challenges in order to take higher power processors such as Multi-cores downhole, currently researches are being done, in Schlumberger labs to provide a cooling technique that can be taken downhole.

LIST OF ABBREVIATIONS

BER …………………………………………………………….… **B**it **E**rror **R**ate

CAN …………………………………………………… **C**ontroller **A**rea **N**etwork

DSP …………………………………………………… **D**igital **S**ignal **P**rocessing

DMT …………………………………………… **D**iscrete **M**ulti-**T**one Modulation

EDTS …………………………………………… **E**nhanced **D**igital **T**elemetry **S**ystem

EFTB ……………………………………………………. **E**nhanced **F**ast **T**ool **B**us

FSM …………………………………………………………… **F**inite **S**tate **M**achine

GAPI …… **G**amma Ray of radioactivity of 200 **A**merican **P**etroleum **I**nstitute units

GR ………………………………………………………………….. **G**amma **R**ay

g …………………………………………………………..……… **g**ravity (9.8 m/s)

I/O ……………………………………………………. **I**nput /**O**utput peripherals

LQC …………………………………………………………….. **L**og **Q**uality **C**ontrol

LUT …………………………………………………………… **L**ook **U**p **T**able

OFDM …………………… **O**rthogonal **F**requency **D**ivision **M**ultiplexing

RB …………………………………………………………….. **R**elative **B**earing

RTD ………………………………………….… **R**esistance **T**emperature **D**etector

TLC ……………………………………………..……… **T**ough **L**ogging **C**onveyance

REFERENCES

1. Tiab, Djebbar, Erle C.Donaldson , “ Theory and Practice of Measuring Reservoir Rock and Fluid Transport Properties.” 2nd Edition, Elsevier 2004. pp 21 – 46 , pp 303- 42
2. Manfred G. Prammer , Marian Morys “A High –Resolution LWD Resistivity Imaging Tool – Field Testing in Vertical and Highly Deviated Boreholes” Proceedings of the SPWLA Journal of Formation Evaluation and Reservoir Description, Vol. 50, No1, February 2009.
3. Jeffery R. Hook “ Introduction To porosity “ Proceedings of the SPWLA Journal of Formation Evaluation and Reservoir Description, Vol. 44, No3, May 2003.
4. Darwin Ellis “Formation Porosity Estimation from Density Logs” Proceedings of the SPWLA Journal of Formation Evaluation and Reservoir Description, Vol. 44, No5, September 2003.
5. Jez Lofts , Martin Evans “ A New Microresistivity Imaging Device for use in Non-Conductive and Oil-Based Muds” Proceedings of the SPWLA Journal of Formation Evaluation and Reservoir Description, Vol. 44, No5, September 2003.
6. David Harris , H. E. Weste “ CMOS VLSI Design “ 3rd Edition , Pearson Education , Inc. 2005.
7. Mark Alden , Faisal Arif , Matthew E.Richard “Advancing Downhole Conveyance” Proceedings of the Oil Field Review Magazine , Sugar Land , Texas , Schlumberger , Autumn , 2004.
8. Loveland, Walter D. Morrissey, David J. Seaborg, Glenn T. John “Modern Nuclear Chemistry” Wiley & Sons 2006 , pp 573 – 57.
9. High Temperature Downhole Reservoir Monitoring System – Joseph A. Henfling , Randy AAA Normann- Sandia National Laboratories.
10. Richard G.Lyons. “Understanding Digital Signal Processing “. Prentice Hall PTR, 2001.
11. Schlumberger Oilfield Glossary.
12. Jacky S. Chow, Jerry C. Tu, John M. Cioffi. “A Discrete Multitone Transceiver System for HDSL Applications”- IEEE Journal On Selected Areas In Communications. VOL. 9, NO. 6, August 1991.

13. Lee-Frang Wei. “Terllis-Coded Modulation with Multidimensional Constellations” – IEEE Transactions on Information Theory, VOL IT-33, NO. 4, July 1987.
14. LinG.LotB,”Theory.and.Applications.of.OFDM.and.CDMA.Wideband.Wireless. Communications.” Wiley (2005).
15. Digital Telemetry System DTS High Level Design, Austin Systems Center, Ronald C. Kelly Michael A. Montgomery, December, 1991.
16. Iyad Obeid , Miguel A.L. Nicolelis ,” A multichannel telemetry system for single unit neural recordings”, Journal of Neuroscience Methods 133 (2004) 33–38
17. Max Domeika, “Software Development for Embedded Multi-core Systems: A Practical Guide Using Embedded Intel Architecture”, Elsevier Inc , 2008.
18. John Runia, John Boyes, and Peter Elkington, “Through Bit Logging: A new Method to Acquire Log Data”, Petrophysics, Vol 46, No 4 (August 2005); P. 289-294.
19. Ertan Pcksen and Michael S. Zhdanov, “Apparent Resistivity Correction for Tensor Induction Well Logging in a Deviated Well in an Anisotropic Meduim”, Petrophysics, Vol 44, No 3 (June 2003); P. 196-204.
20. Ollivier Faivre & Gerard Catala, “ Dip Estimation from Azimuthal Laterolog Tools”, SPWLA 36th Logging Symposium (June 1995); P. 26-29.
21. Romain Rrioul, Adam Donald, Randy Koepsell, Zakariae El Marzouki, and Tom Bratton, “ Forward modeling of fracture-induced sonic anisotropy using combination of borehole image and sonic logs”, Geophysics, Vol. 72, No. 4 (July-August 2007) P. E135–E147.
22. Bo Hu, David Conn, and Sandra Reyes,” Smooth Phase Interpolated Keying for Mud Pulse Telemetry”, M&S: Schlumberger J. Of Modeling, Design, and Simulation, Vol. 2, June, 2011.P3-P8.
23. P. Mehta, “Unleash the power of Multi-Core Using a platform Approach”, Multicore Expo, March 2006.
24. Douglas Hamilton “Multi-Core Microprocessors in Embedded Applications”, multicoreWP Rev. 0, 01/200.
25. Georgios Kornaros, “Multi-Core Embedded Systems”, Taylor and Francis Group, 2010.

Appendix A

Appendix A contains Petro physics Terms Description and Logging Industry Related Terms Description.

Term	Definition
Formation dip	The Angle At which a formation bed inclines away from the horizontal. Dip is also used to describe the orientation of bedding planes as specified by the deviation of the upward normal of the plane from the vertical (dip magnitude) and the azimuth of the upward normal (dip azimuth).
Apparent dip	The angle that a plane makes with the horizontal measured in any randomly oriented section rather than perpendicular to strike.
Inclinometry	An instrument used to measure the dip of the Earth's magnetic field.
Rugosity	A qualitative description of the roughness of a borehole wall. Alternatively, the term pertains to a borehole whose diameter changes rapidly with depth. The term usually refers to changes at the scale of logging measurements, a few inches to a few feet, and to the effect this has on logging tool responses. Rugosity can be observed on caliper logs, image logs and by its effect on measurements with a small depth of investigation
Borehole	The wellbore itself, including the open-hole or uncased portion of the well. Borehole may refer to the inside diameter of the wellbore wall, the rock face that bounds the drilled hole.
Open-hole	The uncased portion of a well. All wells, at least when first drilled, have open-hole sections that the well planner must contend with. Prior to running casing, the well planner must consider how the drilled rock will react to drilling fluids, pressures and mechanical actions over time. The strength of the formation must also be considered. A weak formation is likely to fracture, causing a loss of drilling mud to the formation and, in extreme cases, a loss of hydrostatic head and potential well control problems. An extremely high-pressure formation, even if not flowing, may have wellbore stability problems. Once problems become difficult to manage, casing must be set and cemented in place to isolate the formation from the rest of the wellbore. While most completions are cased, some are open, especially in horizontal or extended-reach wells where it may not be possible to cement casing efficiently.
Cased-hole	In Drilling, the portion of the wellbore that has had metal casing placed and cemented to protect the open-hole from fluids, pressures, wellbore stability problems or a combination of these. In Well

Completion, a wellbore lined with a string of casing or liner. Although the term can apply to any hole section, it is often used to describe techniques and practices applied after a casing or liner has been set across the reservoir zone, such as cased-hole logging or Cased-hole testing.

Oil-Base Mud — An emulsified drilling mud in which the continuous phase is oil and the discontinuous aqueous phase occupies less than ten percent of the volume. Electrically nonconductive.

الملخص

صناعة النفط تعتمد على استخدام تقنيات مختلفة للحصول على محتوى المواد الهيدروكربونية القيمة من الأرض. من أجل تحقيق هذا الهدف بوسيلة مربحة ، تنفذ عمليات حيوية لإستخراج بيانات فيزيائية عن المحتوى الصخري للأرض ، بدءً من السطح ، من خلال الاستكشافات الزلزالية ، ثم من خلال تجميع معلومات على طول مسافة الحفر. هذه البيانات من شأنها أن تؤدي إلى إتخاذ القرار الصحيح في الخطوة المقبلة في دورة حياة بئر الإستخراج ، أو لتحديث نماذج تشكيل صخور الأرض لخطط المستقبل. لأن نتائج البينات المجمعة قيمة بما يكفي للتضحية بالوقت الثمين , فقد تستغرق أياماً للحصول عليها. الغرض من هذه الدراسة هو إيجاد حل للتعامل مع ، وهو ليزل يحدث ، الإخفاق في الأقسام الإلكترونية أثناء مهمة التسجيل ، وتسريع العملية عن طريق استخدام أجهزة قادرة على تحقيق المعالجة المطلوبة في أقل الأوقات ، مما يزيد من الحد الأقصى المسموح به لسرعة التسجيل ، والحد من تكلفة صيانة أدوات التسجيل من خلال اقتراح تصميم مكون من وحدات مُكررة ، يمكن أن يكون وسيلة للخروج من خسارة التوقف طويلاً في حال فشل غير متوقع لأجزاء إلكترونية في قاع البئر خلال التسجيل.

عملية التسجيل تبدأ من إنتاج استجابات كهربية من أجهزة الاستشعار المستخدمة ، تحويلها إلي بيانات رقمية ، ثم معالجة البيانات التي حُصل عليها اعتماداً على الكمية الفزيائية المحسوبة وكيف قِيست ، ثم نقل هذه البيانات المجهزة ، على وصلة الإتصال ، إلى السطح.

المعالجات المتقدمة تعد بالعديد من الفوائد التي يمكن الحصول عليها من خلال الطاقة الحسابية العالية التي يمكن أن توفرها. المعالجات متعددة النواة ، والمعالجات المتوازية ، بما توفره من قدرات ، تحتل مكان في كثير من التطبيقات بدءاً من الخوادم و وصولاً الى الأنظمة المدمجة. إنجاز أكثر من مهمة واحدة في وقت واحد يسرع الأداء ، ويسمح لتطبيقات أكثر تعقيداً أن تُنفذ. في هذه الدراسة جوانب أخري للمعالجات متعددة النوى والمعالجات المتعددة تأتي أولا ، هي الموثوقية ، وإستمرار الخدمة ، والتوافق. التطبيق المعروض هنا متعطش لأنظمة يُعتمد عليها و التي يمكن أن تعمل في بيئات قاسية ، مثل آبار النفط. الهدف من الفصل التالي هو تقديم وجهة نظرة موجزة عن مجال صناعة تسجيل معلومات الآبار، وهو التطبيق الرئيسي هنا. الفصل الأول ، أيضاً ، لتوضيح لماذا التسجيل السلكي يعتمد على تجهيز بنية يعتمد عليها (موثوق بها) للعمل في بيئات مثل أسفل البئر، إما بعد الحفر، أو اثناء الأعمال التكميلية ، أو الإنتاج. الفصل الأول يصف كذلك المشكلة المقترحة هنا ومتطلباتها. الفصل الثاني يصف حدود التطبيق ، والحل الذي يمكن أن يُقدم . الفصول التالية تقدم تنفيذاً للحل المقترح. الفصل الأخير يحتوي علي تلخيصاً للموضوع بأكمله ، ويناقش الإستنتاجات ، والعمل المحتمل في المستقبل.

استخدام معالج متعدد النواة لتصميم نظام تجميع المعلومات لأدوات تصوير آبار النفط

إعـداد

محمد محمود الصغير فراج

رسالة مقدمة الي كلية الهندسة ، جامعة القاهرة

كجزء من متطلبات الحصول علي درجة الماجستير في

هندسة الإلكترونيات والإتصالات الكهربية

يعتمد من لجنة الممتحنين :

الأستاذ الدكتورِ/ السيد مصطفي سعد عضوا

الدكتور/ إبراهيم قمر عضوا

الأستاذ الدكتور/ أمين محمد نصار المشرف الرئيسي

الدكتور/ حسام علي فهمي مشرفا

كلية الهندسة ، جامعة القاهرة

الجيزة ، جمهورية مصر العربية

مارس 2012

استخدام معالج متعدد النواة لتصميم نظام تجميع المعلومات لأدوات تصوير آبار النفط

إعـداد

محمد محمود الصغير فراج

رسالة مقدمة الي كلية الهندسة ، جامعة القاهرة

كجزء من متطلبات الحصول علي درجة الماجستير في

هندسة الإلكترونيات والإتصالات الكهربية

تحت إشراف

أ. د. أمين محمد نصار   د. حسام علي فهمي

كلية الهندسة ، جامعة القاهرة

الجيزة ، جمهورية مصر العربية

مارس 2012

استخدام معالج متعدد النواة لتصميم نظام تجميع المعلومات لأدوات تصوير آبار النفط

إعـداد

محمد محمود الصغير فراج

رسالة مقدمة الي كلية الهندسة ، جامعة القاهرة

كجزء من متطلبات الحصول علي درجة الماجستير في

هندسة الإلكترونيات والإتصالات الكهربية

كلية الهندسة ، جامعة القاهرة

الجيزة ، جمهورية مصر العربية

مارس 2012