**Acknowledgement**

The project undertaken by our team was not just a task but an arduous journey undertaken by us which was filled with experiences that were unique ultimately will prove to be fruitful for us in the near future. Undertaking this project was a massive commitment and a responsibility bestowed upon us by our mentors during the limited amount of time we were present here.

We would like to extend our sincerest thanks to our mentors Mr. Rites Ranjon, Mr. Abhilash Dikshit and Mr. K.K. Jha (Refueling Technology Division) for giving us this opportunity and showing faith in us where experience and professionalism counted the most. We would also like to appreciate Bhabha Atomic Research Centre for giving us this unique opportunity to work on a project with such unparalleled uniqueness and complexity.

The support we received during the entire period of training was very encouraging which helped us to think out of the box and to come up with some innovative solutions which ultimately gave us a optimized final product.

**Executive summary**

The project began on 4/5/19 at hall 7 electronics lab. First day of project work included familiarizing ourselves with the theory of the fuelling machine and its working. In conjunction to this we began our research into the theoretical approach towards optimization of instructions to hardware and began learning/revising the concepts of VHDL which is short for Very High Speed Integrated Circuit Hardware Description Language. The next part in our task was to understand the basic concepts of MSL or Manual Safety Logic/Locks. MSl is nothing but a set of instructions which are supposed to be the parent level barriers to prevent any anomalous inputs which will result in catastrophic consequences. As the theoretical parts were completed the project moved towards the implementation stage. A document was prepared listing all the conditions that should be satisfied for the safe and efficient functioning of the fuelling machine assembly. Our objective was to understand these conditions and code the given conditions using VHDL. A total of 6 modules with subsections were provided and had different requirements. Each module was divided into tasks and divided between team members. Inputs were taken and discussed and finally the gate level diagram was drawn for reference which could also be considered as the architecture of the system. This was repeated with all the modules and then verified. With the gate level diagrams as reference our team coded the requirements with precision and very less margin of error. The next stage was the observation of test bench waveforms which helped us verify whether the given logic was faithful or not. After obtaining the test bench waveforms, the next stage was the hardware testing stage where in we were provided the “TEST JIG” and an FPGA (Field Programmable Gate Array). A “TEST JIG” is nothing but a set of switches which will simulate all the conditions for the program/logic stored in the FPGA. Before testing our team had to go through the documentation of the FPGA and the test kit to map the appropriate I/O according to what we had assigned in the program. It is to be noted that the test bench waveforms of all the modules were successfully obtained but the hardware level testing of only one module (module 5a) was finished because of a lack of adequate testing kits.

**M&SL TEST JIG**

The M&SL test jig is a state of the art device which lets a user test any VHDL logic uploaded to an FPGA in real time. The test jig used in this project is manufactured by RATO Communications (RACE) and supports testing of one FPGA board at a time. The test jig in consideration here provides three I/O lines. Each I/O has up to 62 individual switch I/Os which have two states (high/low) and can be controlled manually so that synthetic testing conditions can be created. The test jig supports up to 24 V DC supply with probe type inputs and also a standard plug and socket input. Each I/O on the test jig can be manually mapped by the user with the help of the mapping tables. Table below shows the I/O mapping for testing of module 5a:

|  |  |
| --- | --- |
| **I/O names in program** | **Corresponding I/O on test jig** |
| AI 57 | DI 1-1 |
| DI 13 | D1 1-2 |
| DI 14 | DI 1-3 |
| DI 15 | DI 1-4 |
| DI 16 | DI 1-5 |
| DI 86 | DI 1-6 |
| DI 87 | DI 1-7 |
| DI 88 | DI 1-8 |
| DI 89 | DI 1-9 |
| DI 90 | DI 1-10 |
| DI 91 | DI 1-11 |
| DI 92 | DI 1-12 |
| DI 93 | DI 1-13 |
| DI 94 | DI 1-14 |
| DI 95 | DI 1-15 |
| DI 98 | DI 1-16 |
| DI 99 | DI 1-17 |
| Sense\_Finger\_error | DI 1-18 |
| Y5\_a | DO-1 |

\* **DI 1 & DI 19 weren’t mapped by the software because of redundancy in the logic where these pins were involved.**

\* **To obtain pin reports and mappings done by libero IDE click on design flow , in the bottom click on export pin reports . After that in the work window click on project summary under that click on \*project name\* reports.**

\* **The entire test jig works in active low or gives out inverted output.**

**ProASIC3 Flash Family FPGAs**

**with Optional Soft ARM Support**

**Features and Benefits**

**High Capacity**

• 15 K to 1 M System Gates

• Up to 144 Kbits of True Dual-Port SRAM

• Up to 300 User I/Os

**Reprogrammable Flash Technology**

• 130-nm, 7-Layer Metal (6 Copper), Flash-Based CMOS

Process

• Instant On Level 0 Support

• Single-Chip Solution

• Retains Programmed Design when Powered Off

**High Performance**

• 350 MHz System Performance

• 3.3 V, 66 MHz 64-Bit PCI†

**In-System Programming (ISP) and Security**

• ISP Using On-Chip 128-Bit Advanced Encryption Standard

(AES) Decryption (except ARM®-enabled ProASIC®3 devices)

via JTAG (IEEE 1532–compliant)†

• FlashLock® to Secure FPGA Contents

**Low Power**

• Core Voltage for Low Power

• Support for 1.5 V-Only Systems

• Low-Impedance Flash Switches

**High-Performance Routing Hierarchy**

• Segmented, Hierarchical Routing and Clock Structure

**Advanced I/O**

• 700 Mbps DDR, LVDS-Capable I/Os (A3P250 and above)

• 1.5 V, 1.8 V, 2.5 V, and 3.3 V Mixed-Voltage Operation

• Wide Range Power Supply Voltage Support per JESD8-B,

Allowing I/Os to Operate from 2.7 V to 3.6 V

• Bank-Selectable I/O Voltages—up to 4 Banks per Chip

• Single-Ended I/O Standards: LVTTL, LVCMOS 3.3 V /

2.5 V / 1.8 V / 1.5 V, 3.3 V PCI / 3.3 V PCI-X† and LVCMOS

2.5 V / 5.0 V Input

• Differential I/O Standards: LVPECL, LVDS, B-LVDS, and

M-LVDS (A3P250 and above)

• I/O Registers on Input, Output, and Enable Paths

• Hot-Swappable and Cold Sparing I/Os‡

• Programmable Output Slew Rate† and Drive Strength

• Weak Pull-Up/-Down

• IEEE 1149.1 (JTAG) Boundary Scan Test

• Pin-Compatible Packages across the ProASIC3 Family

**Clock Conditioning Circuit (CCC) and PLL†**

• Six CCC Blocks, One with an Integrated PLL

• Configurable Phase-Shift, Multiply/Divide, Delay Capabilities

and External Feedback

• Wide Input Frequency Range (1.5 MHz to 350 MHz)

**Embedded Memory**†

• 1 Kbit of FlashROM User Nonvolatile Memory

• SRAMs and FIFOs with Variable-Aspect-Ratio 4,608-Bit RAM

Blocks (×1, ×2, ×4, ×9, and ×18 organizations)†

• True Dual-Port SRAM (except ×18)

**ARM Processor Support in ProASIC3 FPGAs**

• M1 ProASIC3 Devices—ARM®Cortex®-M1 Soft Processor

Available with or without Debug.

**The FPGA which we have used in this project is ProAsic3L A3P600L** – **FG484. Please refer to the datasheet for more details.**

**Conclusion**

The project concluded with hardwired testing of one of the modules amongst several modules concerning the interlocks. In between testing it was found out that some of the interlocks in the core design were overlapping hence were inferred to be redundant. Although redundancy in complex safety systems is given utmost importance, the revelation of this redundancy opens up new windows in hardware optimization in this and similar systems.

During the entire course of the project, exhaustive testing was conducted including some least likely scenarios both on software and hardware level. This exhaustive simulation helped us better understand the system design at a very intricate level which further helped us debug and simplify the circuit and or the entire system.

For testing the entire setup, an in depth study of the entire setup was conducted and each scenario was built from scratch through which all the bugs in the code were traced back. It would be worthwhile to note that the entire setup of interlocks consisted of total twenty modules which needed to be repeatedly flashed into the FPGA and due to lack of time and suitable programming equipment the entire system could not be tested. However, it should also be noted that all the modules were passed through rigorous software tests and all interlocks were seen to be satisfied. It can also be concluded that software simulation gives us opportunities to simulate scenarios which are otherwise very difficult (as well as dangerous) to try on complex and large assemblies.

**Future work**

* All modules should be tested by flashing the same on the ProAsic3L FPGA.
* Software simulation should be enhanced by introduction of more rigorous test benches. *Rigorous* here refers to complex inputs with more than one input making a transition at the same instant of time.
* All modules and sub modules should be simultaneously tested; this would open up more opportunities in the concept of pin and BUS multiplexing due to the large number of inputs (both analog and digital) involved.
* The same methodology and coding style should be followed throughout while developing similar such interlocks in VHDL.
* A similar convention of variable names should be used especially when referring to the same set of variables, parameters, predefined values and constants.
* As specified in the main list of interlocks, the memory organization (non –volatile) for constants should be done in a way so as to facilitate faster access and easier retrieval. Maximal usage of on-board ROM is encouraged in such cases.