One-Step -- Logic Automatic Translation For FPGA Applications

Allen HSU[1] and Steve YANG[2]

1. Doosan HF Controls Corp., 1624 W. Crosby Road, #124, Carrollton, TX 75006, USA (allen.hsu@doosan.com)
2. Doosan HF Controls Corp., 1624 W. Crosby Road, #124, Carrollton, TX 75006, USA (steve.yang@doosan.com)

Abstract: Field Programmable Gate Arrays (FPGAs), as programmable logic devices (PLDs) have gained interests for implementing safety Instrumentation and Control (I&C) applications in nuclear power plants (NPPs) owing to the FPGAs' potential advantage over the currently more common microprocessor-based digital I&C applications. Generic I&C platforms using FPGA have been developed for safety applications such as Reactor Protection System (RPS) and Engineered Safety Features Actuation System (ESFAS). The RPS and ESFAS implementation starts with plant specific requirements and licensing basis, which are translated into system requirements specification and followed by the system architectural design. The system design and architecture implementation is then realized via the generic FPGA platform, where the RPS and ESFAS applications logics are executed.

To execute the RPS and ESFAS applications logics, logic drawings or diagrams are first translated into Hardware Description Languages (HDLs). Then, using an electronic design automation tool, a technology-mapped netlist is generated. This process is called synthesis, in which the HDL or schematic design is translated to logic gates, memory units, registers and connections. The netlist can then be implemented by the FPGA manufacturer's proprietary software to fit to the actual FPGA architecture. This includes translation, map and place-and-route processes. The designer as well as V&V engineers will verify the map, place-and-route results via timing analysis, simulation, and other verification methodologies. Once the design and verification process is complete, the programming file generated (also using the FPGA manufacturer's proprietary software) is employed to (re)configure the FPGA. This file is transferred to the FPGA via a serial interface (JTAG).

The process of translating logic drawings into the HDLs is laborious and errors prone, especially for a system with a great number of inputs and outputs. It is therefore desired to automate the translation process, and achieve errors-free in implementing and executing safety logic algorithms. For this reason, One-Step software tool for FPGA applications has been developed. One-Step not only automates the logic translations but also creates dynamic images of a CAD (Computer-Aided Design) drawing for display on the workstations so that logic drawings can be dynamically monitored with the system on-line. The One-Step tool has been evaluated in accordance with the relevant industry standards (i.e., guidance as provided in IEEE Std 7-4.3.2-2003).

Keyword: FPGA, One-Step, software tool, logic translation, lifecycle, microprocessor, nuclear I&C, V&V

1 Introduction

Field Programmable Gate Arrays (FPGAs) have gained interest for implementing safety I&C applications in nuclear power plants (NPPs). There are many benefits in using FPGA technology for safety I&C applications owing to the FPGAs' potential advantage over the currently more common microprocessor-based nuclear digital I&C applications. FPGAs, from software perspective, can be made simpler, less reliant on complex software such as operating systems, which should make FPGAs easier to qualify for nuclear safety applications. FPGAs are less vulnerable to cyber-attacks when FPGAs implement the I&C systems that do not contain high-level, general purpose software that may be easily subjected to malicious modifications. FPGAs can process separate functions independently and in parallel on the same integrated circuit, which makes FPGAs highly efficient in their performance. FPGAs can bring cost reduction in an I&C digital upgrade

because FPGAs can provide simpler licensing process than microprocessor-based digital I&C, and FPGAs can be implemented more efficiently. These benefits provide reasonable justification for the implementation of FPGA-based applications in NPPs, which is expected to grow significantly in the near future.

However, there are challenges for the application of FPGA in the nuclear industry. The use of FPGA technology in the nuclear industry is relatively new and therefore in general there are not many application cases for reference as experience and lessons-learn. The FPGA applications to the nuclear I&C need specialized expertise of design engineers, who are not readily available in the nuclear industry. The HDL compiler is FPGA manufacturer specific and the correctness of the end downloadable file needs extra verification and validation efforts. Finally, for complicated numerical calculations, FPGA loses its advantages to microprocessors because of the limited resource that FPGA has.

Despite these challenges, the advantages of the FPGA are very attractive. Therefore, the nuclear digital I&C industry should take advantages of the FPGA benefits while realizing presence of the challenges.

When using FPGA to implement nuclear I&C applications, two approaches have been considered. First of all, FPGA is used as Microprocessor's Emulator. This is to emulate the operation of microprocessors and to interpret the execution of their CPU processes. In this scheme, existing software of communication networking and control algorithms can be adopted. A recent upgrade using FPGA to replace a Single Board Controller in the YGN Unit 3&4 nuclear power plant has been a successful case where the obsolete Intel 8085 microprocessor was replaced by an FPGA chip [1][2].

Secondary, FPGA is used for generic platform for nuclear safety application, where requirements of nuclear safety I&C including fundamental design principles, calculation power, response time, etc. are considered. The fundamental design principles include single failure criteria (e.g., single random failure and single event upset); determinism; architectural simplicity; diversity and defense-in-depth; physical, electrical and communication independence; and fail-safe considerations.

The above two design approaches are considered in HFC-FPGA platform development. With proper application configured, the HFC-FPGA platform can be used in the nuclear safety and non-safety applications including RPS, ESFAS and DPS (Diverse Protection System).

The HFC-FPGA platform is the FPGA version of the HFC-6000 platform, which has obtained the US NRC SER in April 2011. With successful completion of Equipment Qualification tests, the HFC-FPGA platform was submitted to the US NRC for review and evaluation in August 2017.

When using FPGA for RPS or ESFAS applications, the RPS and ESFAS implementation starts with plant specific requirements and licensing basis, which are translated into system requirements specification and followed by the system architectural design. The system design and architecture implementation is then realized via the generic FPGA platform, where the RPS and ESFAS applications logics are executed.

To execute the RPS and ESFAS applications logics, logic drawings or diagrams are first translated into Hardware Description Languages (HDLs). Then, using an electronic design automation tool, a technology-mapped netlist is generated; this synthesis process translates HDL or schematic design to logic gates, memory units, registers and connections. The netlist can then be implemented by the FPGA manufacturer's proprietary software to fit to the actual FPGA architecture. This includes translation, map and place-and-route processes. The designer as well as V&V engineers will verify the map, place-and-route results via timing analysis, simulation, and other verification methodologies. Once the design and verification process is complete, the programming file generated (also using the FPGA manufacturer's proprietary software) is employed to (re)configure the FPGA. This file is transferred to the FPGA via a serial interface (JTAG).

The process of translating logic drawings into the HDLs is laborious and errors prone, especially for a system with a great number of inputs and outputs. It is therefore desired to automate the translation process, and achieve errors-free in implementing and executing safety logic algorithms. For this reason, based upon its microprocessor version of the software tool, One-Step for FPGA applications has been developed. The One-Step software tool not only automates the logic translations but also creates dynamic images of a CAD (Computer-Aided Design) drawing for display on the workstations so that logic drawings can be dynamically monitored with the system on-line. The One-Step tool has

been evaluated in accordance with the relevant industry standards (i.e., guidance as provided in IEEE Std 7-4.3.2-2003[3], and IAEA Nuclear Energy Series, No. NP-T-3.17[4]).

One-Step for a microprocessor based platform has been developed and used widely in the NPP I&C applications since early 2000; and it has been a proven tool for logic automation since then[5]. Because of this, One-Step for microprocessor based applications provides a valid and proven reference to assess the correctness of the One-Step for FPGA based applications.

2 One-Step Tool Developments

2.1 Development Lifecycle and Design Architecture

In the development of One-Step software tool, the lifecycle process as defined IEEE Std 1074-2006[6], which is endorsed by the US Regulatory Guides 1.173[7], was followed.

One-Step software tool Requirements Specification was developed, which defines the functions, capabilities, and limitations of the tool. The One-Step software tool Design Description was followed, which shows how the tool is structured to satisfy the requirements identified in the tool Requirements Specification. The Design Description is a translation of requirements into a description of tool structure, tool module components, interfaces, and data necessary for the implementation of the tool. The tool Design Description becomes a detailed blueprint for the implementation activity.

During the tool implementation and testing process, one of the key elements is the assurance of the manufacturers' FPGA software libraries including its third party libraries and tools. To reach high confidence of the manufacturers' and third party libraries, thorough code review and testing (functional, structural, integration and system testing) of the application logics (its translated HDL code and end downloadable file) were performed, and results were evaluated and analyzed.

Requirements traceability analysis was conducted. Each identifiable design element in the Design Description was traced backwards to the tool requirements. Each identifiable requirement was written so that it is also forward traceable to subsequent design outputs. Each requirement was traced to one or more design entities. Forward traceability to all documents spawned by the Requirements Specification includes verification and validation (V&V) materials. A forward trace exists from each requirement in the Requirements Specification to the specific inspections, analyses, or tests used to confirm that the tool requirements have been met.

One-Step software architecture is composed of two software modules that perform separate functions: one that generates an HDL logic file suitable for compiling into downloadable files for the target FPGA chip and one that generates graphic file for static and dynamic display purpose.

The two software modules are implemented using different approaches. Specifically, an Object Oriented Programming technique is adopted to generate the HDL file. A Top-Down approach is used to produce the static and dynamic graphical display.

The program module that generates the output HDL file and the program module that produces the dynamic graphic file are separate but integral parts of the One-Step software. Figure 1 illustrates the logic flow of One-Step software.

One-Step is developed in a Windows 7 environment using Visual C++. It is a standalone program that is activated by clicking a single button (or selecting a single menu option) in the HFC program list. When the software begins operation, a dialog box will appear and prompt users to select the control loop or loops to be translated. Section 3 illustrates the use of the One-Step from a simple logic automation example.

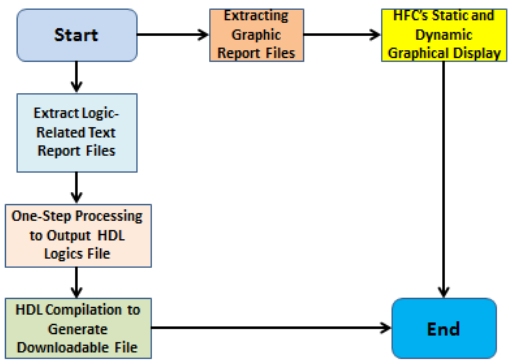

Fig. 1 One-Step software architecture process control flow chart

2.2 Module for Generating HDL File

This module is based on the Object Oriented

Programming technique. The concept of a user-defined type – class is adopted, so that data become private. The procedures that access the data move inside the class and become properties and methods. Classes not only facilitate software reuse but also contribute to fault containment because errors can be localized within the class.

A control system is defined as a Project, which consists of a number of Installations that represent control loops.

A hierarchy is established in which the project class that represents a controller system contains the installation class that represents loop numbers, which in turn contain the page class that depicts each drawing. The page class contains symbol classes, which are the lowest level. The symbol class defines all control logic symbols in the CAD drawings and connecting lines including connection information between symbols, point IDs (controller cards and I/O cards, and logic data types), and symbol types (logic gates AND, OR, and NOT etc.) are attributes of the class.

The module scans data extracted from the Promis-E database and uses the hierarchical class structure with defined attributes to generate a symbol array. The module then uses the symbol array to create an HDL file. The HDL logics are eventually compiled into the downloadable files by using FPGA manufacturers' libraries (including the third party libraries). The module also writes the report text file to the project database.

Different from the microprocessor based One-Step, where the generated logic equations can be compiled into binary files using the HFC proprietary Equation Interpreter, the FPGA based One-Step has rely on the manufacturer's libraries for the complication process. These libraries may be third parties that are incorporated into the manufacturers' compiling process. These tools or libraries developers may not be able to provide detailed evidence of their configuration management for their tools/libraries or the level of error tracking and quality assurance needed to support and maintain their tools/libraries as safety grade software for nuclear power plant applications. Therefore, it is critical to ensure the correctness of the end results (the downloadable files) in safety applications via rigorous code review of the application logics and its converted HDL file, and thorough testing of the end FPGA downloadable file. Section 4 describes V&V approaches to ensure the One-Step produce correct results and does not introduce or mask unwanted errors during the FPGA applications.

2.3 Module for Producing Static and Dynamic Graphic Display

This module uses Top-To-Down procedural design approach. The module produces an output file based on a graphic report file produced by Promis-E. The display format of the graphic file matches the layout of the CAD logic diagram. The graphics contains dynamic data that correspond to inputs and outputs of the logic functions.

This module produces static and dynamic graphic displays from the extracted graphic report file. Properties such as line, poly-line, circle, arc, ellipse, and text from the graphic report input are reformatted to serve both static and dynamic aspects of the graphic representation. Status parameters are clearly defined so that static and dynamic displays are accurately presented by the HFC MCRT utility.

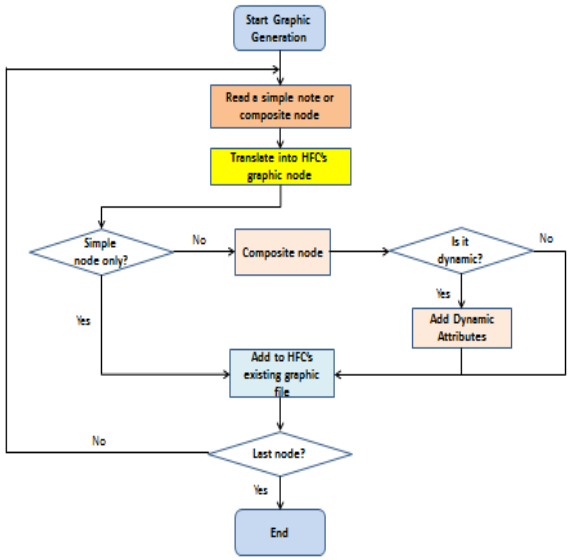

Fig. 2 Graphic generation process flow

Figure 2 shows the One-Step graphics generation process. One-Step graphic generation will result in the graphics as in the application logics with dynamic features added for facilitating parameters displays and monitoring as well as system diagnostics.

3 Illustration of One-Step FPGA Downloadable File Generation Process

The One-Step software tool enables the user to generate the application code executable in the

HFC-FPGA controller from Promis-E drawings. It also generates graphic files compatible with HFC MCRT software from the same Promis-E drawings. Thus, the engineers can create and modify control logic application codes of a loop controller in a completely graphic programming environment. The application code generated can be loaded directly into an FPGA as the application firmware of a controller card. The graphic files can be displayed by MCRT software for monitoring the dynamic operation of application logics.

The One-Step is a utility of HFC EWS/MCRT software suite. The Promis-E software is a commercial software package that can be used with HFC-defined macros and symbols to generate Promis-E drawings and report data files. The report files supply the source of data used by One-Step to generate its output.

Application engineers use the Promis-E software to define control loop algorithms in the form of Promis-E drawings. The Promis-E software runs as an Add-on to AutoCAD and allows users to set up various attributes for the drawings and their components. The information for each Promis-E project is stored in an associated database, and it is extractable from Promis-E.

As shown in Figure 3, the process of generating control algorithms with One-Step code generation includes making Promis-E drawings, extracting Promis-E reports, generating application code, and installing that code in an FPGA. Performing each step properly is essential for obtaining reliable application code from the original control algorithms specification.

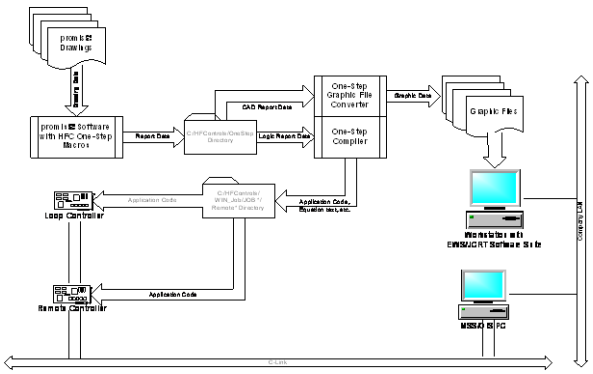

Fig. 3 One-Step operation and process flow

For simplicity, an example illustration is provided below to show how the One-Step tool is used in generating the FPGA programming file to be downloaded into the FPGA based controllers.

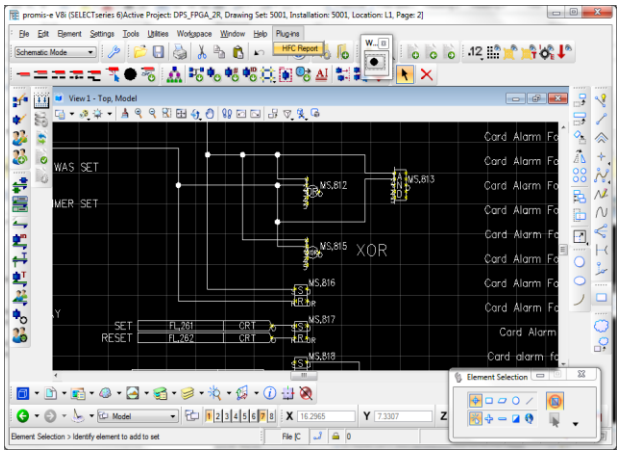

Fig. 4 AND and OR examples simple logics

In this example, AND and OR gates logics were created using Promis-E tool with logic connecitons (Figure 4). Use Promis-E tool to make the logic connections. After logic connections are completed, click on menu item Plug-ins and select HFC Report.

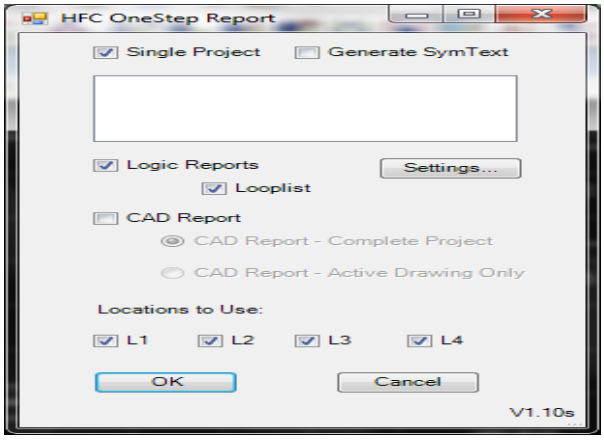

Fig. 5 Reporting generation selection

HFC Report can make Promis-E output files for logic and graphics (Figure 5). We only make output files for logic in this illustration.

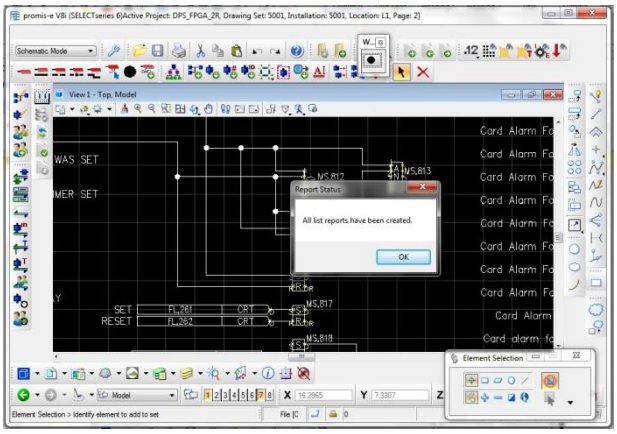

Fig. 6 Message indication

After the plug-in is completed, a message will indicate if the reports have been created (Figure 6).

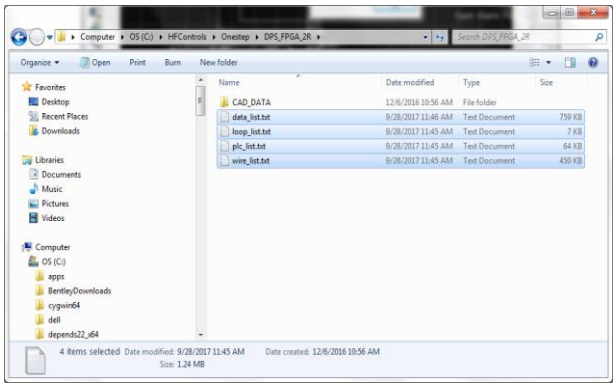

Fig. 7 shows the output files.

The Promis-E output files are located on the EWS PC hard drive. There are only 4 files for logic output (Figure 7).

Now run the One Step Control program as shown in Figure 8 by clicking P (or G) button.

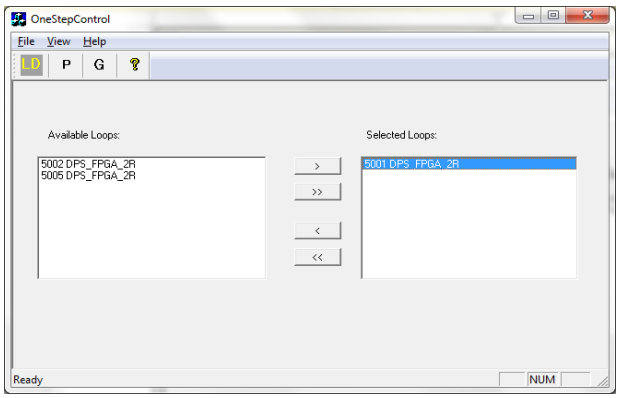

Fig. 8 Loop selection and processing the logics

Select the loop or loops to compile and click the "P" button to Process the loops. The One-Step will then process the Promis-E output files. The One-Step populates an Access database with project information, makes loadable files that can be downloaded into the FPGA controller, makes a HDL file for input into the Libero FPGA compile tool for eventual generation of FPGA programming file and thus downloadable file.

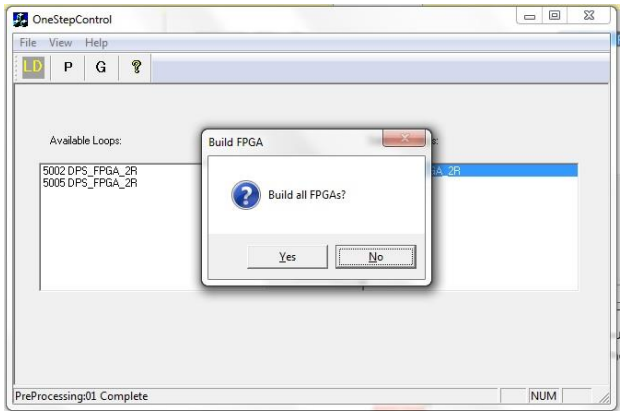

Fig. 9 Pop-up Message Selection to Invoke Generation of Downloadable File

A message box pops up to let the user to decide if the FPGA files are to be created (Figure 9). If yes, the One-Step then calls on the Libero tool (third party) to spawn two processes that compiles the HDL code into downloadable FPGA files. One process is spawned for the Control FPGA build and another process is spawned for the Diagnostic FPGA build.

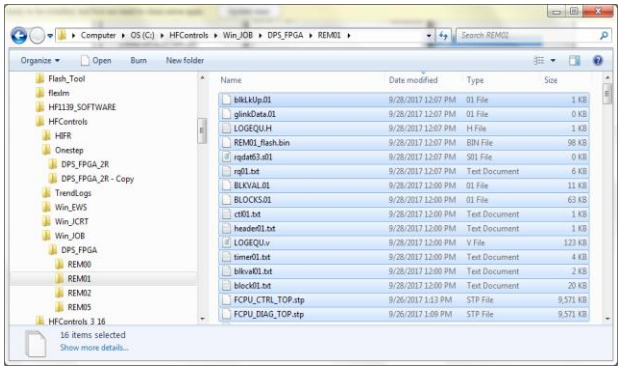

Fig. 10 Loadable programming file generation

One-Step makes one loadable file for programming the flash memory on the HFC-FCPUX. This file is called REM01_flash.bin. The Libero processes spawned create two loadable files which are called FCPU_CTRL_TOP.stp and FCPU_DIAG_TOP.stp (Figure 10).

The LOGEGU.V file is the HDL file output.

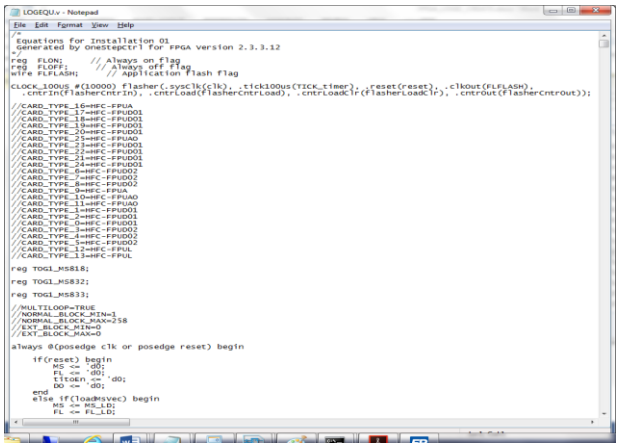

Fig. 11 HDL file that contains application logics

Opening the HDL file shows a readable ASCII text (Figure 11).

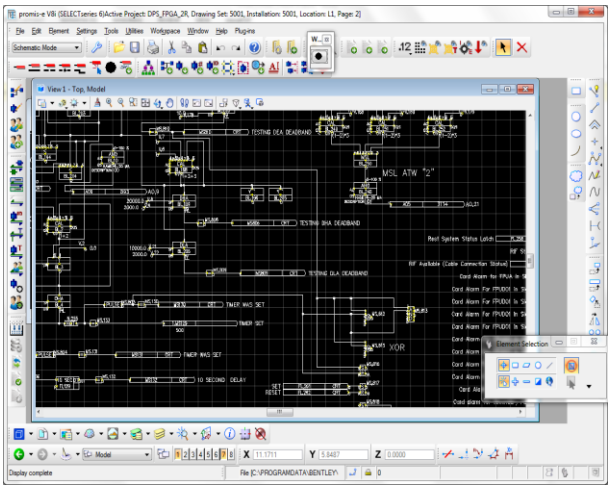

Fig. 12 Expanded AND and OR gates logics

Figure 12 shows an expanded view of the AND and OR gate logic shows the inputs to these gate are MS, 810 and MS, 803.

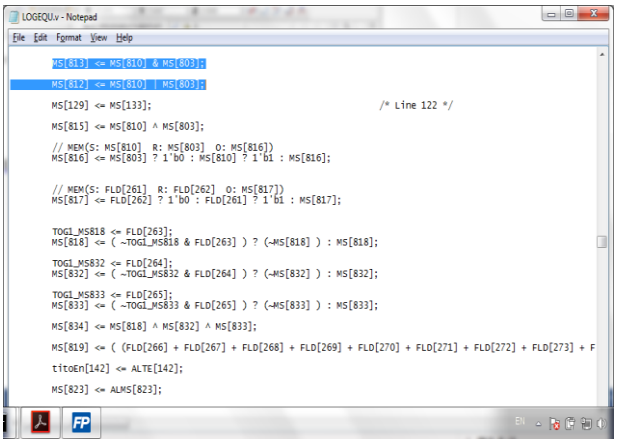

Fig. 13 Logics connections in the HDL code

Figure 13 shows AND gate MS, 813 and OR gate MS, 812 in the HDL code. The HDL shows the inputs to these gate are MS, 810 and MS, 803.

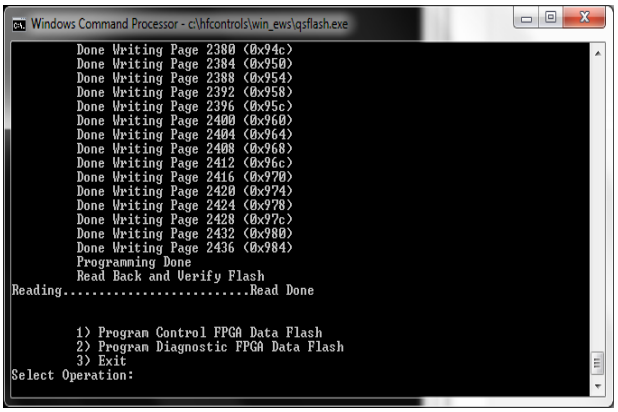

Fig. 14 Programming FPGA chip

Figure 14 shows that One-Step output files are programmed into the HFC-FCPUX. Use the command line program qsflash to download the REM01_flash.bin binary file to both Control and Diagnostic FPGA flash memory. Use a USB programming cable called HFC-FLASH Programmer to connect the PC to the HF-FCPUX.

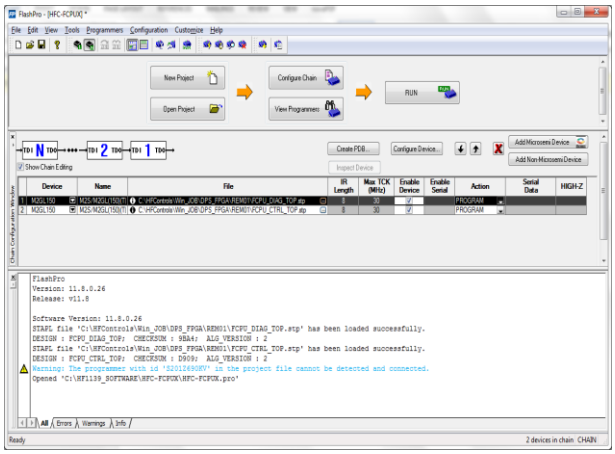

Fig. 15 Downloading the executable file to FPGA controller

Finally, use Actel's FlashPro tool and FlashPro5 USB programming cable to program the STP files to both Control and Diagnostic FPGA via the JTAG port on the HFC-FCPUX (Figure 15).

FPGA code has been downloaded into the target FPGA controller.

4 One-Step Software Tool V&V

HFC implements a V&V program in validating tools for use in FPGA applications. This program is consistent with the guidance provided by the IEEE Std 7-4.3.2-2003 and the V&V methodologies specified in the IEEE Std 1012-2004[8] to evaluate and qualify tools before they are used. The IEEE Std 1012-2004 requires that tools that insert code into the software be verified and validated to the same rigor as the highest software integrity level of the software.

The guidance specified in IEEE Std 7-4.3.2-2003 requires that software tools used to support software development processes and V&V processes shall be controlled under configuration management. One or both of the following methods shall be used to confirm the software tools are suitable for use: 1) A test tool validation program shall be developed to provide confidence that the necessary features of the software tool function as required. And 2) the software tool shall be used in a manner such that defects not detected by the software tool will be detected by V&V activities. Finally, tool operating experience may be used to provide additional confidence in the suitability of a tool, particularly when evaluating the potential for undetected defects.

IAEA Nuclear Energy Series, No. NP-T-3.17[4] provides similar guidance for the use of software tools. Per the IAEA technical report, "Most national regulatory bodies require that all tools that will be used to perform safety functions or that are used in the development process for hardware, software or firmware to perform a safety function be treated as safety software (method II in Fig. 16). This can be a significant regulatory burden for pre-developed software tools used in any digital system development cycle. Tool developers typically do not provide detailed evidence of their configuration management for their tools or the level of error tracking and quality assurance needed to support and maintain their tools as safety grade software for nuclear power plant applications."

"The alternative method that has been found acceptable to most regulatory bodies is to independently review the input and output of tools as part of the development process quality assurance program to ensure that the tool is functioning properly, and that it has not introduced any possible new fault or failure mode into the final digital system or failed to detect an error if that is part of the function of the tool. Owing to the challenges with method II, this method (method I in Figure 16) is more commonly used at the current time.

When tools are used for specific, well bounded parts of the development process, method I has proved to be effective."

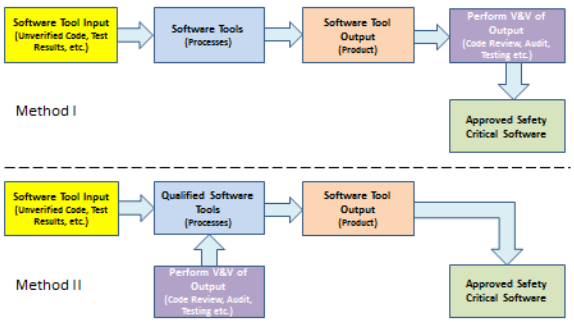

Fig. 16 Two Acceptable methods for tool qualification[4]

Based on these guidance, the qualification of HFC-FPGA platform One-Step consists of the following steps including tool requirements specification development, tool detailed design and implementation, tool V&V program, tool revision control and the use of the tool in nuclear safety I&C

applications.

The tool V&V program includes specific steps as follows:

- Review and Verification of Tool Requirements Specification and Design Implementation,
- Tool Code Review and Walkthrough,
- Tool Code Coverage Testing (complete for all needed logics gates and MACROs),
- Tool Functional Coverage Testing (all logics functions),
- Tool Functional and Timing Simulation Testing (on all required logics and selected examples), and
- Tool Use in the FPGA Circuitry System Testing (on selected typical applications as well as loops logics that have been used in operating NPPs).

After the logic is converted into the HDL file, as part of the One-Step process, the FPGA manufacturer's proprietary software and libraries are used to generate the programming file, which is employed to (re)configure the FPGA. This file is transferred to the FPGA via a serial interface (JTAG). The programmed FPGA based control system is then verified and validated via system testing.

Additionally, the proven microprocessor based One-Step was used to verify and validate the correctness of the end results from the FPGA based applications (See Figure 17).

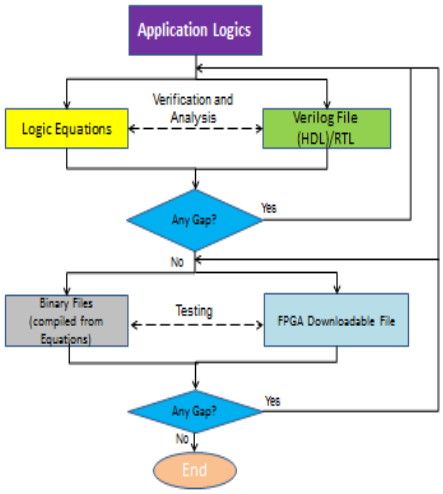

Fig. 17 Using proven microprocessor based One-Step as a reference to verifiy and validate the correctness of the end results from the FPGA based One-Step

This approach for using the proven history of the One-Step for microprocessor based applications provides additional confidence that the One-Step tool for FPGA applications generates correct results and does not mask errors.

5 Summaries and Conclusion

One-Step software tool for FPGA application has been developed based on One-Step for microprocessor-based applications. The software tool was developed using the Object Oriented Programming and Procedural Programming approaches. One-Step for FPGA applications automates the logic translation from CAD drawings to HDL format in implementing and executing loop control algorithms for FPGA applications. One-Step creates the CAD drawings that can be stored in one source electronically. These not only replace manual labors in the logic translation and archiving logic CAD drawings, but human errors are eliminated. One-Step can also create dynamic images of a CAD drawing for display on the workstations so that logic drawings can be dynamically monitored with the system on-line.

The development and V&V of the One-Step software tool for FPGA applications have followed the relevant industry guidance, and as such, the tool is suitable for nuclear safety I&C applications.

Abbreviations and acronyms

CAD – computer assisted drawing
CPU – central processing unit
DPS – Diverse Protection System
ESFAS – Engineered Safety Features Actuation System
EWS – engineers workstation
FPGA – field programmable gate array
HFC – Doosan HF Controls Corporation
HDL – hardware description language
I&C – instrumentation and controls
IAEA – International Atomic Energy Agency
IEEE – Institute of Electrical and Electronics Engineers
I/O – Input/Output
MCRT – Microsoft CRT
NPP – Nuclear Power Plants
NRC – US nuclear regulatory commission
PLD – programmable logic devices
RPS – Reactor Protection System
SER – safety evaluation report

USB – Universal Series Bus
VHDL – Verilog HDL
V&V – Verification and Validation
YGN – Yonggwang NPP

Nomenclature

Actel – Actel Corporation, an FPGA manufacturer, now Microsemi Corporation.
HFC-6000 – HFC safety I&C platform that has been reviewed extensively by the US NRC and obtained the US NRC SER in April 2011.
HFC-FPGA – HFC FPGA platform.
HFC-FCPUX – HFC-FPGA platform controller.
JTAG – a serial interface.
Libero – Microsemi tool for FPGA.
Microsemi – Microsemi Corporation, provides semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets.
Netlist – describes the connectivity of an electronic design. Netlists convey connectivity information and provide instances, nets, and attributes.
One-Step – HFC logic translation automation tool.
Promis-E – an Add-on to AutoCAD and allows users to set up various attributes for the drawings and their components.
Verilog – a HDL used to model electronic systems

Acknowledgement

We would like to thank William Luo and Huaisong Xu for beneficial discussions in the earlier draft preparation of this paper. We would like to thank David Briner for supplying screen captures used for the example illustration of this paper.

References

[1] FPGA Implementation in Nuclear Digital I&C Applications, by A. Hsu, I. Chow and S. Yang, presented at NPIC & HMIT, Charlotte, NC, February 22-26, 2015.

[2] FPGAs Emulate Microprocessors – A Successful Case for HFC NPP Digital I&C Upgrade, by Allen Hsu, Ivan Chow, Carl Reese, Jong Kim, and Steve Yang, ISOFIC/ISSNP 2014, Jeju, Korea, August 24-28, 2014.

[3] IEEE 7-4.3.2-2003, IEEE Standard Criteria for Use of Computers in Safety Systems of Nuclear, the Institute of Electrical and Electronics Engineers, Inc., New York, USA.

[4] IAEA Nuclear Energy Series, No. NP-T-3.17, Application of Field Programmable Gate Arrays in Instrumentation and Control Systems of Nuclear Power Plants, Technical Report, International Atomic Energy Agency, Vienna, 2016.

[5] One-Step: Automating Logic Translations and Creating CAD Drawings for Process Controls, by S. Yang, Proceedings of the 2003 IEEE International Symposium on Intelligent Control, Houston, Texas, October 5-8, 2003.

[6] IEEE Std 1074-2006, IEEE Standard for Developing Software Life Cycle Processes, the Institute of Electrical and Electronics Engineers, Inc., New York, USA.

[7] RG 1.173-2013, Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.

[8] IEEE Std 1012-2004, IEEE Standard for Software Verification and Validation, the Institute of Electrical and Electronics Engineers, Inc., New York, USA.