PRESENT CIPHER
documentation

Krzysztof Gajewski
and opencores.org

www.opencores.org

gajos@opencores.org

Change History

Rev.	Chapter	Date	Description	Reviewer
0.1	all	2014/02/01	First draft	K. Gajewski

Contents

1 Introduction

Present is ``ultra-lightweight´´ block cipher developed by A. Bogdanov et al. and proposed in 2007 [1]. It uses 64 bit data block and 80 bit or 128 bit key. This cipher consists of 32 rounds, during which:

- round key is added to plaintext
- plaintext goues through sBoxes (substitution boxes)
- plaintext after sBoxes goes through pLayer (permutation layer)
- round key is updated

After that, ciphertext feeds out the output. Briefly algorithm was shown in Fig. 1 In this project

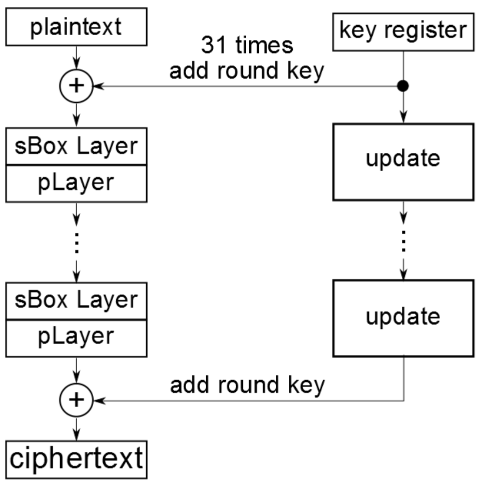

Figure 1: Briefly block scheme of the PRESENT block cipher

Present block cipher works with 80 bit key. Target was Xilinx® Spartan 3E XC3S500E [2] on Spartan 3E Starter Board [3] made by Digilent®.

2 Interface

Top level component of present was shown in Fig. 2. All inputs and outputs are synchronous except `reset` signal and sampled at rising edge of the clock. Type for all signals is `STD_LOGIC` or `STD_LOGIC_VECTOR`.

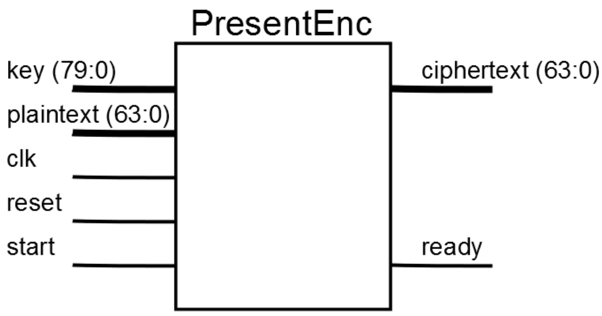

Figure 2: Top level component of Present component

Signal name	Width	In/Out	Description
`key`	80	in	secret key used for input data encoding.
`plaintext`	64	in	input data which have to be encoded.
`clk`	1	in	clock signal for the component
`reset`	1	in	*Asynchronous* reset signal.
`start`	1	in	signal which starts encoding process.
`ciphertext`	64	out	encoded text output.
`ready`	1	out	signal informing about end of encoding process. "0" - wait until end of data encoding. "1" - data at the `ciphertext` output are valid, you can read them.

Table 1: Input/Output signals of Present component

3 State machine workflow

Overall internal structure of Present component is similar to the structure shown in [1]. Suitable control logic was added in state machine added to the core. It was shown in Fig. 3.

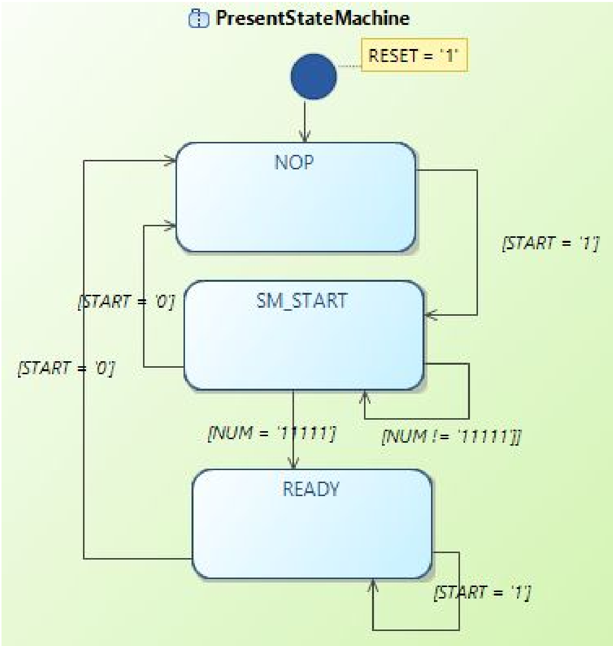

Figure 3: State machine of the Present component

State machine consist of three states `NOP`, `SM_START` and `READY`. Some control signal of used multiplexers, registers and counter was omitted. `NOP` is default state after resetting the core. This state is active as long as `START` = '0'.

When `START` = '1' encoding process starts. Proper `key` and `plaintext` must fed the input before start encoding. `SM_START` state is active as long as `START` = '1'. Change of this signal to '0' automatically stops encoding process.

After 32 clock cycles (counter reach `'11111'` value), when encoding process is ended, state machine automatically change its state to `READY`. This informs user by setting `READY` output to '1'. Then `ciphertext` output contains proper data, which can be readed by user. This state are active as long as `START` = '1'. Change this signal to '0', change state machine to `NOP` state. Core is ready for next data encoding.

4 FPGA implementations

The component has only been verified on a Xilinx® Spartan 3E XC3S500E FPGA in FG320 package and synthesized with Xilinx ISE 14.2. Appropriate setup files was prepared with use of ISE Project Navigator, but Makefile scripts was also written. Suitable files was stored in `./Pure/syn/XC3ES500/` directory. Implementation in FPGA device was done in another sub-project called `PureTesting`. Makefile was tested in Windows 8 with use of Cygwin for 64-bit Windows.

Synthesis results was given in Fig. 4

Xilinx ®Spartan 3E XC3S500E FPGA in FG320 package			
Parameter	**Used**	**Available**	**Utilization**
Number of Slices	248	4656	5%
Number of Slice Flip Flops	151	9312	1%
Number of 4 input LUTs	296	9312	3%
Number of bonded IOBs	212	232	91%
Number of GCLKs	1	24	4%
Minimum period	5.035ns	-	-
Maximum Frequency	198 MHz	-	-

Table 2: Synthesis results for Spartan 3E XC3S500E

Possible change in used FPGA device may be possible in steps given below[1]:

1. Copy `./Pure/syn/XC3ES500/` directory to another one like `./Pure/syn/YOUR_FPGA_SYMBOL/`
2. Go to `./Pure/syn/XC3ES500/` directory.
3. In `PresentEnc.xst` file modify the line `-p xc3s500e-5-fg320` to `-p YOUR_FPGA_SYMBOL`
4. In `Makefile` file modify the line `PLATFORM=xc3s500e-fg320-5` to `PLATFORM=YOUR_FPGA_SYMBOL`

[1]This solution was not tested and is based on my own observations. Additional care should be taken with *.UCF files. You can make this modifications on your own risk

5 Simulation

Self-checking test bench were provided to the components used for Present encoder. They are stored in `./Pure/bench/vhdl` directory. Suitable configuration files and Makefile used for running test bench was stored in `.Pure/sim/rtl_sim/bin` directory. Appropriate test vectors was taken from [1].

Makefile was prepared to make "manual run" of tests. If You want to perform it without gui, remove `-gui` option in Makefaile.

6 Troubleshooting

During work with Windows 8 64-bit and and Xilinx® ISE 64-bit some problems may occur:

1. Xilinx may be unable to open projects in Project Navigator.
2. When you run `make` in Cygwin and perform testbench it would be unable to open ISIM gui.
3. When you run ISIM gui (*.exe test bench file) it hangs out or anti virus protection opens.

To solve problems listed above you have to perform steps listed below:

1. You have to rename libraries `libPortabilityNOSH.dll` to `libPortability.dll` from `nt64` directories (http://www.gadgetfactory.net/2013/09/having-problems-installing-xilinx-ise-on-windows-8-64bit-here-is-a-fix-video-included/)
2. Firstly, install Cygwin X11 (http://stackoverflow.com/questions/9393462/cannot-launch-git-gui-using-cygwin-on-windows)
3. Temporary switch off anti virus protection.

7 License and Liability

References

[1] A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C. Vikkelsoe, “Present: An ultra-lightweight block cipher,” in *Cryptographic Hardware and Embedded Systems - CHES 2007*, ser. Lecture Notes in Computer Science, P. Paillier and I. Verbauwhede, Eds. Springer Berlin Heidelberg, 2007, vol. 4727, pp. 450–466. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74735-2_31

[2] Xilinx. (2014, Feb.) Spartan-3e fpga family data sheet. [Online]. Available: http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf

[3] Digilent. (2014, Feb.) Spartan 3e starter board. [Online]. Available: http://www.digilentinc.com/Products/Detail.cfm?Prod=S3EBOARD