An On-chip Glitchy-clock Generator
Quick Start Guide
- Version 1.0 -

October 17, 2011

Graduate School of Information Sciences, Tohoku University

0 Introduction

This document describes the first steps to use the on-chip glitchy-clock generator. The source code of the glitchy-clock generator comes with a sample program which demonstrates the function of the generator. You can try out glitchy-clock instantly with the program. We have confirmed correct operation of the sample program on Windows XP.

1 Setting up your SASEBO board

Before setting up the on-chip glitchy-clock generator and running its test program, follow the steps of the SASEBO Quick Start Guide, which is available at
`http://staff.aist.go.jp/akashi.satoh/SASEBO/pdf/SASEBO_QuickStartGuide_Ver1.0_English.pdf`.
The "SASEBO Tester" program which is required in the SASEBO Quick Start Guide is also available at the page above.

2 Software installation

The sample program is written in Python. Python 2.6(Python environment) and API which supports USB or serial port are required for execution. Python 2.6 is available at
`http://www.python.org/download/releases/2.6.6/`. The following software products might also be required. If you use a serial port to connect the SASEBO board to your PC, install pySerial 2.5 available at `http://pypi.python.org/pypi/pyserial`. If you use a USB connection, install pyUSB 1.6 available at `http://bleyer.org/pyusb/`.

3 FPGA configuration

The configuration steps are the same as those of in the SASEBO Quick Start Guide. To reprogram the flash ROM (XCF16P) for the control FPGA, use the provided mcs file `Control_SASEBO.mcs` for the SASEBO (-G), or `Control_SASEBO-R.mcs` for SASEBO-R. In the SASEBO(-G), reprogram the flash ROM (XCF8P) for the cryptographic FPGA with the provided mcs file `RSA_xv2vp7.mcs`. If you use a SASEBO prototype board, change the TCK Speed/Baud rate to 3 MHz before the following steps (Fig.1). You can find this option in the Output → Cable Setup menu in the

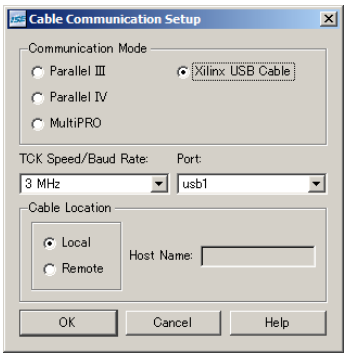

Figure 1 Cable Setup dialog.

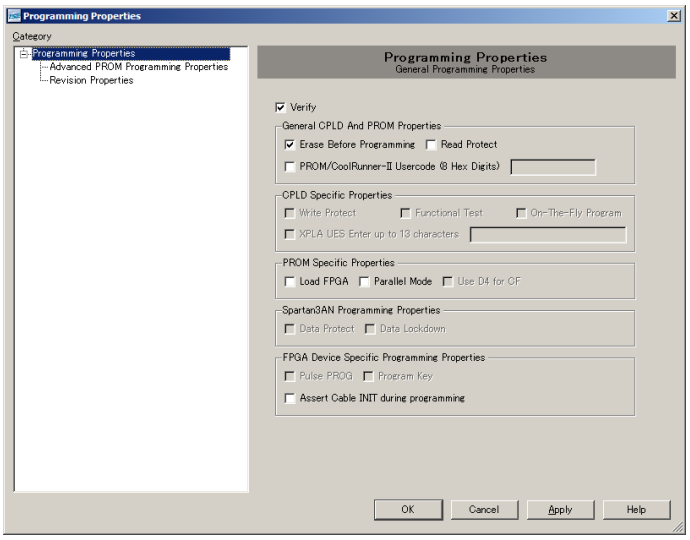

Figure 2 Properties dialog.

iMPACT programming software. On the programming property screen of the iMPACT, as shown in Fig. 2, check Verify and Erase, and uncheck Parallel Mode. Push "CONFRST" button to configure the FPGA before running the sample program.

4 Running the sample program

The sample program communicates with the control FPGA on SASEBO through the USB interface by default. If you use a serial port for communication, modify the device number in the sample program. Replace "COM3" with the name of your PC's serial port.

```
#Instantiate the cipher module
#cm = CipherModuleRSA(interface = "USB")
# If you use serial port, specify the port.
cm = CipherModuleRSA(interface = "COM3")
```

After installing the software, run the sample program using the commands below:

```
cd /path/to/sample.py
C:\Python26\python.exe sample.py
```

You will get on output as shown in Fig. 3. You can verify whether the result is correct by comparing the result with the correct ciphertext.

```
 Set RSA core to mode  0
 CRT mode was set to  0
 result             = 0x119cc7894073cfb531aa0b7c4dc6604035ff2f9e572f04e68686f1637e322ace34c
04edb881847643b10e40a91fcec5116ecfaba43dcc12dd9811a34a3fc5f6f
 correct ciphertext = 0x119cc7894073cfb531aa0b7c4dc6604035ff2f9e572f04e68686f1637e322ace34c
04edb881847643b10e40a91fcec5116ecfaba43dcc12dd9811a34a3fc5f6fL
```

Figure 3 Output of sample program.

5 Configuring the glitchy-clock generator

The glitchy-clock generator is configured in the sample program. Fig. 4 presents the glitch parameters. "Position of glitch" is the interval between the positive edge of the EXEC signal and the glitchy-clock cycle. There are two registers to configure the glitch position, `POSITION` and `POS_FINE`. These registers specify how many cycles the generator waits before it injects the glitch. The position is defined as follows:

$$[\text{Position of glitch}] = \texttt{POSITION} \times 577 + \texttt{POS\_FINE} \tag{1}$$

The coefficient 577 can be changed in the source file `DelayedTrigger.v`.

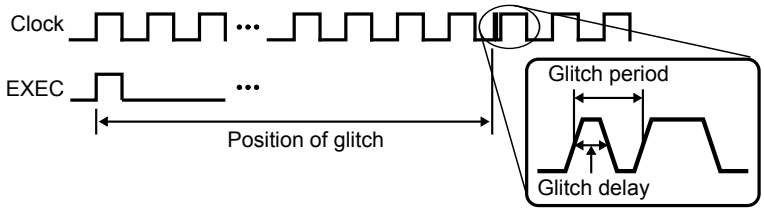

Figure 4 Glitch parameters.

The glitch delay and glitch period are configured through registers `DELAY` and `PERIOD`, respectively. The time unit of these register is 0.195 ns in the Virtex-II Pro FPGA.

Fig. 5 shows a sample code that configures the glitchy-clock generator. You can change the parameters by the numbers in the function call `cm.write_param`. You will receive a faulty ciphertext with the initial settings because a glitch is injected. When you change the number from "1" to "0" on the last line in the code shown in Fig. 5, you will get the correct ciphertext.

```
# Period of glitch = 15
cm.write_param(addr_list["DELAY"], 15)
# Period of glitch = 30
cm.write_param(addr_list["PERIOD"], 30)
# Position of glitch = 14
cm.write_param(addr_list["POSITION"], 14)
# Fine adjustment = 0
cm.write_param(addr_list["POS_FINE"], 0)
# Enable glitch injection
cm.write_param(addr_list["GLITCH_EN"], 1)
```

Figure 5 Code to configure glitchy-clock generator.

If you connect an oscilloscope to the CN11-2 (on SASEBO(-G)) or CN6-2 (on SASEBO-R) pin, you will get the waveform like Fig. 6.

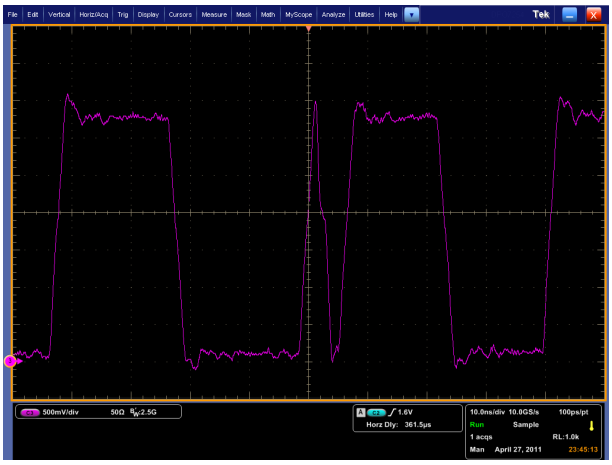

Figure 6 Waveform of glitchy-clock cycle.

6 Using glitchy-clock generator on your HDL source code

The glitchy-clock generator employs the DCM (Digital Clock Manager) to shift the phase of clock signal in the case of Xilinx FPGAs. The DCMs are configured for a variable phase shift. When you compile the source code of the glitchy-clock generator, you need to configure properly the DCM to use the function of DCMs as follows.

First, open the FPGA Editor and find the DCM in the List dialog as shown in Fig. 7. FPGA Editor can be accessed from Implement Design → Place & route → View/Edit Routed Design (FPGA Editor). You can easily find the DCM by clicking the column "Site." Find the DCMs named u31 and u23, which are used as variable phase shifter. Then, open the Process Properties dialog by right-clicking Generate Programming File and selecting Properties. On the dialog shown in Fig. 8, enter the following command line option into the text box "Other Bitgen Command Line Options."

`-g Centered_<location of a DCM>:0`

This option setting makes the phase shift parameter on the DCM located in `<location of a DCM>` a positive number. In this example, the DCMs are located on `x1y0` and `x1y1`, so enter

"`-g Centered_x1y0:0 -g Centered_x1y1:0`" in the box. You need to modify the location of DCM before the generation of programming file because it changes every time. For more details, see the Xilinx answer at `http://www.xilinx.com/support/answers/15130.htm`.

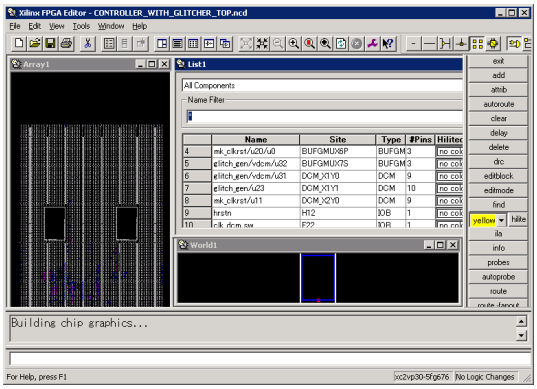

Figure 7 FPGA Editor.

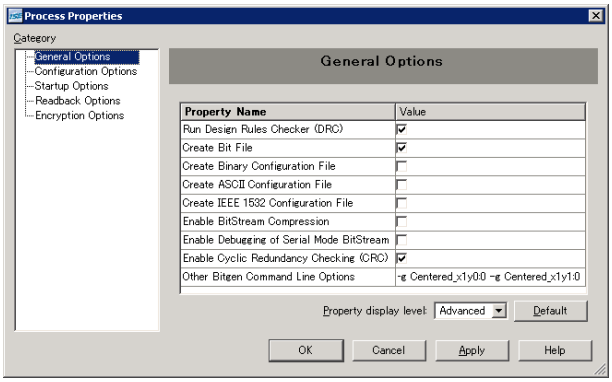

Figure 8 Properties dialog.