How to do readback capture on the Zynq UltraScale+ MPSoC

Introduction

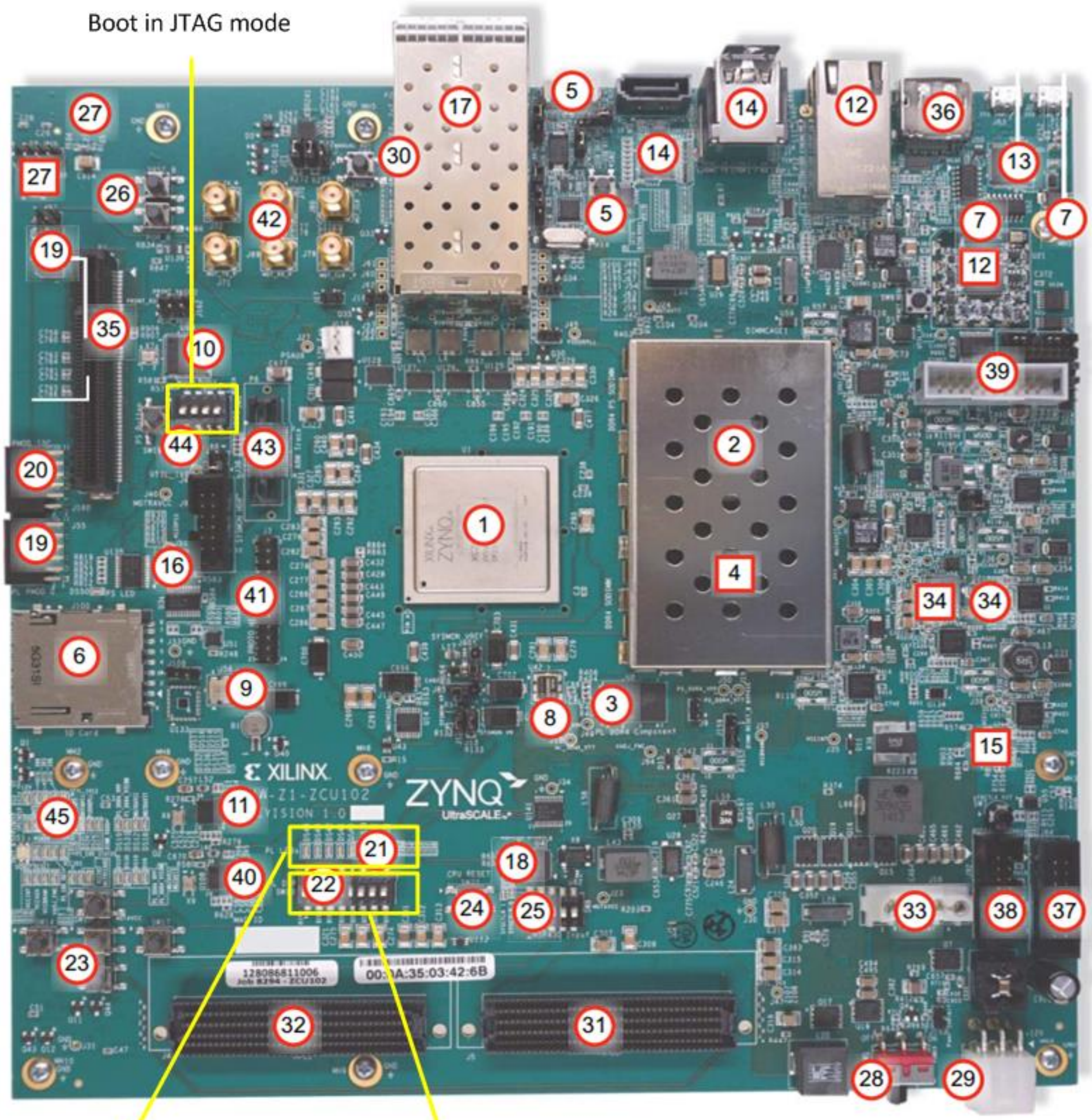
This document adopts the method from the XAPP1230 for doing readback capture on Xilinx UltraScale devices and shows how to migrate the same task to Zynq UltraScale+ MPSoC with several noticeable differences.

Demonstration setup

The experiment is taking place on top of the ZCU102 development board with the following settings as shown in **Figure 1**:

- The PL LEDs are used for displaying the 7-bit LED counter. The number of lighting LEDs is up to the number of JTAG clock cycles set by the user as explained in Section “Experimental steps”.
- The DIP7 in the PL DIP is used to reset the LEDs.
- The board is set to boot in JTAG mode.
- The readback is done by Vivado Hardware Manager via JTAG.

Boot in JTAG mode



8 LEDs to display to counter

DIP 7 is used for RST signal

Figure 1. ZCU102 settings for the readback capture experiment.

Reference design description

Design:

- Verilog files: LED_Count.v, cntr.v, and bscan.v are basically identical to the files from the XAPP1230 reference design. There is a minor change in the bscan.v to rename the BSCAN2 instantiation.

- Constrain files: LED_Count.xdc is adopted for the ZCU102 board with the pin settings for input/output according to the aforementioned board setup. Moreover, the ZCU102 reference constrain file is also attached (for reference purpose only).

Demo files:

- Bitstream file: LED_count_zcu.bit/rbt/rbd.

- Logic location file: LED_count_zcu.ll

Tcl script:

- Script file: readback_capture_zcu.tcl is adopted for Zynq UltraScale+ ZU9EG device with new JTAG commands for the readback. The LED_count_zcu.bsd (generated by the *write_bsdl* command in Vivado) is also attached for reference.

Main differences in JTAG commands and frame lengths are listed as below (for JTAG commands of ZU9EG, please refer to the .bsd file for more detail):

- Instruction length:

```
set DEF_IL      6       # for UltraScale device

set DEF_IL      12      # for ZU9EG Zynq UltraScale+ device
```

- Word per frame

```
set DEF_WPF 123         # for UltraScale device

set DEF_WPF 93          # for ZU9EG Zynq UltraScale+ device
```

- Pipeline value to add to frame count

```
set DEF_FDR_PIPE_DEPTH 10     # for UltraScale device

set DEF_FDR_PIPE_DEPTH 25     # for ZU9EG Zynq UltraScale+ device
```

- JTAG instruction opcodes

```
        # for UltraScale device
set DEF_USER4          0x23
set DEF_JCONFIG         0x05
set DEF_JRDBK          0x04
set DEF_BYPASS          0x3F
set DEF_CMD_IDCODE       0x09
```

```
        # for ZU9EG Zynq UltraScale+ device
set DEF_USER4          0x923
set DEF_JCONFIG         0x905
set DEF_JRDBK          0x904
set DEF_BYPASS          0xFFF
set DEF_CMD_IDCODE       0x249
```

Experimental steps

- Create a project with reference design files
- Generate the bitstream:

```
write_bitstream -verbose -raw_bitfile -logic_location_file -readback_file LED_Count_zcu
```

- Programming FPGA via JTAG

```
close_hw_target
```

- Enabling the Count and Stopping the Clock

```
open_hw_target -jtag_mode 1
scan_ir_hw_jtag 12 -tdi 923 # Shift USER4 opcode in the JTAG instruction register
runtest_hw_jtag -tck 174 # LED[7:0] = 10101110
scan_ir_hw_jtag 12 -tdi fff # Shift the BYPASS opcode to ensure the clock is stopped
```

- Readback capture – Tcl script usage

```
source ./readback_capture_zcu.tcl
rdbk_jtag ./LED_Count_zcu.rdbk 71260 1
```

- Repeat the test for other settings

- Reset the CLB register pattern by setting the RST SW to 1. Then issue these commands:

```
scan_ir_hw_jtag 12 -tdi 923
runtest_hw_jtag -tck 1 # Feed the counter with 1 clock cycle only to reset the counter
```

- Load the CLB register with inverted pattern with these commands:

```
scan_ir_hw_jtag 12 -tdi 923
runtest_hw_jtag -tck 81
scan_ir_hw_jtag 12 -tdi fff
rdbk_jtag ./LED_Count_zcu_inverted.rdbk 71260 1
```

Discussion

The equation to determine the word line for the CLB register value is:

*int(bit offset/32) + pipeline word count + (number of padding frame*number of word per frame) + editor word = word line,*

With: pipeline word count = 25, number of padding frame = 1, and number of word per frame = 93 in the case of ZU9EG device, as shown in **Figure 2**.

For example, the states of Q[3-0] can be found in the .rdbk readback bitfile in the line of 956186 while the states of Q[7-4] in the line of 956207 if the Notepad++ text processing tool is used.

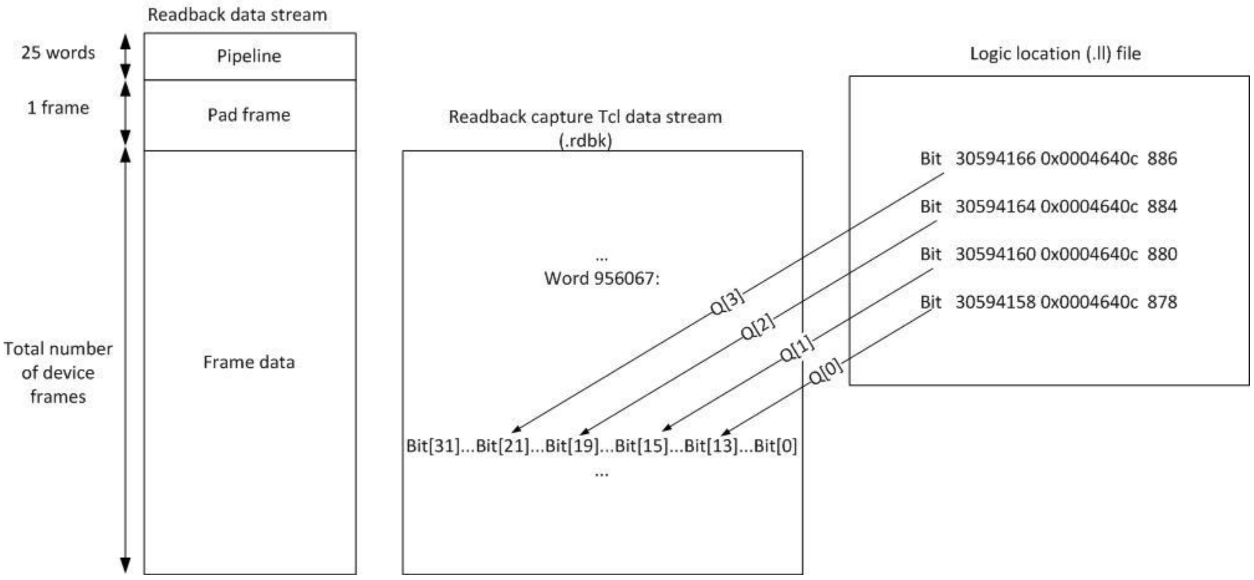

Readback data stream
25 words
Pipeline
1 frame
Pad frame
Total number of device frames
Frame data
Readback capture Tcl data stream (.rdbk)
...
Word 956067:
Bit[31]...Bit[21]...Bit[19]...Bit[15]...Bit[13]...Bit[0]
...
Q[3]
Q[2]
Q[1]
Q[0]
Logic location (.ll) file
Bit 30594166 0x0004640c 886
Bit 30594164 0x0004640c 884
Bit 30594160 0x0004640c 880
Bit 30594158 0x0004640c 878

Figure 2. Example of using the .ll file to identify design element locations.