This tutorial is made to explain how to use BitMan for partial reconfiguration.

Overview

BitMan is a bitstream manipulation tool for Xilinx FPGAs, built in the University of Manchester [1]. It has below features:

- Report resource usage (LUT, BRAM, DSP, clock, wires…) on Xilinx FPGAs
- Relocate a partial module into new position on FPGA fabric or compose a partial bitstream for further partial reconfiguration
- Modify LUT's values
- Support Xilinx Virtex-6, 7-Series, UltraScale, and UltraScale+ families
- Compatible with ISE and Vivado design flows

Installation

BitMan can run from a single executable binary bitman.exe without any prior installation on Windows machines.

```
bitman.exe –h
```

Usage

Use the help feature to explore console functionality

```
bitman.exe -h
```

BitMan - a configuration bitstream analizing tool for Xilinx FPGAs

Usage: bitman options [input_bitfile_1] [input_bitfile_2] option [output_file]

Options	***Features***
-v [input_bitfile_1]	Verbose (print out Xilinx's configuration commands)
-c [input_bitfile_1]	Print CLB info including bitstream encoding
-x COL1 ROW1 COL2 ROW2	Cut out an FPGA region from COL1 ROW1 (bottom-left) COL2 ROW2 (top-right)
-m COL1 ROW1 COL2 ROW2 [input_bitfile_1] [input_bitfile_2]	Cut out an FPGA region from COL1 ROW1 (bottom-left) COL2 ROW2 (top right) in ***full bitstream bitfile_1*** and merge it into the ***full bitstream bitfile_2***
-r COL1 ROW1 COL2 ROW2 COL3 ROW3 [input_bitfile_1]	relocate an FPGA region from COL1 ROW1 (bottom-left) COL2 ROW2 (top-right) to a new position which starts at COL3 ROW3 (bottom-left)
-d COL1 ROW1 COL2 ROW2 COL3 ROW3 [input_bitfile_1]	duplicate an FPGA region from COL1 ROW1 (bottom-left) COL2 ROW2 (top-right) to a new position which starts at COL3 ROW3 (bottom-left)
-S COL ROW LUT value_h value_l	Set value (value_h, valuel) in the 1 of 8 LUTs at the COL ROW
-F [output_file]	Write (linked) full bitstream into file
-M COL3 ROW3 [output_file]	Write module bitstream into partial file in the location start at COL3 ROW3

Coordination system

The coordination system used by BitMan for its (COLx, ROWx) arguments is according to the Xilinx "interconnection tile grid". In the XC7Z020 (ZedBoard), the COL and ROW values of a resource column are the X-Y coordinator of its Switchbox.

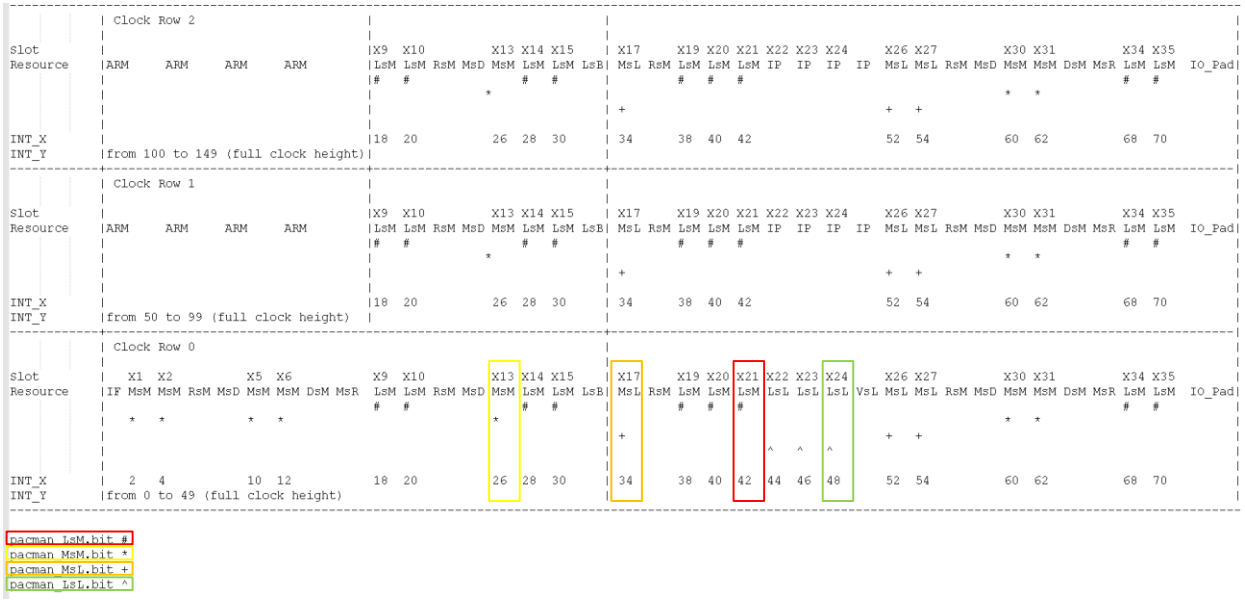

Figure 1. Possible PACMAN placements on XC7Z020 (ZedBoard) chip

Examples for PACMAN demo

Figure 1 shows an abstract of resources on the XC7Z020 (ZedBoard) chip. It also gives an example of possible placements for different PACMAN modules on the chip.

On this chip, currently we have implemented PACMAN on 4 different tuples of resources: LsM, MsM, MsL and LsL. They come from separated designs and therefore separated bitstreams. In this example, we have another bitstream for the only static design.

Bitstream	***Feature***
PAC_007_STC_4.bit	Only static bitstream
pacman_LsM.bit	PACMAN module implemented on LsM resources
pacman_MsM.bit	PACMAN module implemented on MsM resources
pacman_MsL.bit	PACMAN module implemented on MsL resources
pacman_LsL.bit	PACMAN module implemented on LsL resources

To be short, a specific PACMAN module must be placed on a region correspondent to its resources (i.e. pacman_LsM on LsM slots and so on). Otherwise, the configuration will be corrupted and cause unknown consequences on the chip, including permanently hardware damaged.

Placing a module and generating a statically linked bitstream

For example, we want to merge the PACMAN with the LsM resource into the slot 21. According to Figure 1, we have its bottom-left corner at (42, 0).

In this task, we need to do 2 actions by merging PACMAN module from the `pacman_LsM.bit` into the static `PAC_007_STC_4.bit` (`-m` option) and writing new configuration data into another ***full bitstream*** (`-F` option). Combined with above coordinators, we have the below command.

```
bitman.exe –m 42 0 43 49 pacman_LsM.bit PAC_007_STC_4.bit –F merged_pacman_LsM_42_0.bit
```

Duplicating a module and generating a new bitstream

In this example, we want to copy a normal connection signals from the slot 19 into a region that has gaps in signals due to the placement of connection macro with the LsM resource in the slot 20. According to Figure 1, we have its bottom-left corners of slot 19 and 20 at (38, 0), and (40, 0) respectively.

In this task, we duplicate from the region in slot 19 (i.e. copy the configuration data) to the slot 20 in the same static `Static_original_with_connection_macros.bit` ***full bitstream*** (`-d` option), and then write new data into another ***full bitstream*** (`-F` option). Therefore, we have the below command.

```
bitman.exe -d 38 0 39 49 40 0 Static_original_with_connection_macros.bit -F
Static_filled_row_1.bit
```

Placing a module and generating a partial bitstream

In this example, we want to cutout the PACMAN with the MsM resource from its ***full bitstream*** and generate a ***partial bitstream*** so that we could place it on the running static at slot 31. Figure 1 shows us the bottom-left corner of that slot 31 is (62, 0).

This task implies 2 other actions by cutting out PACMAN module from the `pacman_MsM.bit` (`-x` option) and composing a partial bitstream which has an appropriate Frame Address Register (FAR) value to load module onto correct position (`-M` option).

We have the below command

```
bitman.exe –x 26 0 27 49 pacman_MsM.bit –M 62 0 pacman_MsM_clk_0_slot_31.bit
```

Downloading the bitstream from the command line

A TCL script, called load_bitstream.tcl, is prepared in order to download a bitstream from the command line. Steps are described as below:

```
REM Source the Vivado run-time environment
c:\Xilinx\Vivado\2017.1\.settings64-Vivado.bat

REM Set the bitfile to be downloaded
set bitfile=PAC_007_STC_4.bit

REM Call Vivado to execute the load_bitstream.tcl script to download the
bitstream onto the FPGA
call vivado -nolog -nojournal -notrace -mode batch -source load_bitstream.tcl -quiet
```

Appendix A will show all commands for the demo to partially reconfigure 6 PACMAN modules with different resources on the running static system.

References

[1] Khoa Dang Pham, Edson Horta, Dirk Koch, “BITMAN: A tool and API for FPGA bitstream manipulations”, in *DATE* 2017.

Appendix A – PACMAN demo

```
@REM c:\Xilinx\Vivado\2017.1\.settings64-Vivado.bat
@REM Load the static
@ECHO
################################################################################
@ECHO Load the static bitstream
@ECHO
################################################################################

@REM defines global variables
@set X_LsM=42
@set X_LsM_1=43
@set X_MsM=26
@set X_MsM_1=27
@set X_MsL=34
@set X_MsL_1=35

@set clock_region=0
REM Y = clock_region * 50; Y_1 = Y + 1
@set /a Y=%clock_region%*50
@set /a Y_1=%Y%+49

@REM defines the static bitstream
@set bitfile=PAC_007_STC_4.bit

call vivado -nolog -nojournal -notrace -mode batch -source load_bitstream.tcl
-quiet
REM Load the first pacman in slot 21 - LsM

REM defines slot's variables
@set slot=21
@set slot_type=LsM
REM X_slot = slot * 2; X_slot_1 = X_slot + 1
@set /a X_slot=%slot%*2
@set /a X_slot_1=%X_slot%+1

@ECHO
################################################################################
@ECHO
################################################################################
@ECHO Press enter for creating a partial pacman module to slot %slot%
@ECHO Slot %slot% has the resource footprint "%slot_type%" therefore, we
relocate bitstream
@ECHO pacman_%slot_type% to the target position %slot%
@ECHO
################################################################################
@ECHO
################################################################################
pause
@ECHO
################################################################################
@ECHO
################################################################################
@ECHO running BitMan to relocate bitstream to target position
```

```
REM NOTE: make sure the X_LsM and X_LsM_1 being used for the slot_type LsM
@ECHO bitman.exe -x %X_LsM% %Y% %X_LsM_1% %Y_1% pacman_%slot_type%.bit -M
%X_slot% %Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
@REM will execute bitman.exe -x %X_LsM% %Y% %X_LsM_1% %Y_1%
pacman_%slot_type%.bit -M %X_slot% %Y%
pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
bitman.exe -x %X_LsM% %Y% %X_LsM_1% %Y_1% pacman_%slot_type%.bit -M %X_slot%
%Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
set bitfile=pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
@ECHO
################################################################################
@ECHO
################################################################################
@ECHO Press enter for loading a partial pacman module to slot %slot%
pause
call vivado -nolog -nojournal -notrace -mode batch -source load_bitstream.tcl
-quiet

REM Load the second pacman in slot 13 - MsM

REM defines slot's variables
@set slot=13
@set slot_type=MsM
REM X_slot = slot * 2; X_slot_1 = X_slot + 1
@set /a X_slot=%slot%*2
@set /a X_slot_1=%X_slot%+1

@ECHO
################################################################################
@ECHO
################################################################################
@ECHO Press enter for creating a partial pacman module to slot %slot%
@ECHO Slot %slot% has the resource footprint "%slot_type%" therefore, we
relocate bitstream
@ECHO pacman_%slot_type% to the target position %slot%
@ECHO
################################################################################
@ECHO
################################################################################
pause
@ECHO
################################################################################
@ECHO
################################################################################
@ECHO running BitMan to relocate bitstream to target position
REM NOTE: make sure the X_MsM and X_MsM_1 being used for the slot_type MsM
@ECHO bitman.exe -x %X_MsM% %Y% %X_MsM_1% %Y_1% pacman_%slot_type%.bit -M
%X_slot% %Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
bitman.exe -x %X_MsM% %Y% %X_MsM_1% %Y_1% pacman_%slot_type%.bit -M %X_slot%
%Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
set bitfile=pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
@ECHO
################################################################################
@ECHO
################################################################################
@ECHO Press enter for loading a partial pacman module to slot %slot%
pause
```

```
call vivado -nolog -nojournal -notrace -mode batch -source load_bitstream.tcl
-quiet

REM Load the third pacman in slot 20 - LsM

REM defines slot's variables
@set slot=20
@set slot_type=LsM
REM X_slot = slot * 2; X_slot_1 = X_slot + 1
@set /a X_slot=%slot%*2
@set /a X_slot_1=%X_slot%+1

@ECHO
################################################################################
@ECHO
################################################################################
@ECHO Press enter for creating a partial pacman module to slot %slot%
@ECHO Slot %slot% has the resource footprint "%slot_type%" therefore, we
relocate bitstream
@ECHO pacman_%slot_type% to the target position %slot%
@ECHO
################################################################################
@ECHO
################################################################################
pause
@ECHO
################################################################################
@ECHO
################################################################################
@ECHO running BitMan to relocate bitstream to target position
REM NOTE: make sure the X_LsM and X_LsM_1 being used for the slot_type LsM
@ECHO bitman.exe -x %X_LsM% %Y% %X_LsM_1% %Y_1% pacman_%slot_type%.bit -M
%X_slot% %Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
bitman.exe -x %X_LsM% %Y% %X_LsM_1% %Y_1% pacman_%slot_type%.bit -M %X_slot%
%Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
set bitfile=pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
@ECHO
################################################################################
@ECHO
################################################################################
@ECHO Press enter for loading a partial pacman module to slot %slot%
pause
call vivado -nolog -nojournal -notrace -mode batch -source load_bitstream.tcl
-quiet

REM Load the 4th pacman in slot 17 - MsL

REM defines slot's variables
@set slot=17
@set slot_type=MsL
REM X_slot = slot * 2; X_slot_1 = X_slot + 1
@set /a X_slot=%slot%*2
@set /a X_slot_1=%X_slot%+1

@ECHO
################################################################################
```

```
@ECHO
################################################################################
@ECHO Press enter for creating a partial pacman module to slot %slot%
@ECHO Slot %slot% has the resource footprint "%slot_type%" therefore, we
relocate bitstream
@ECHO pacman_%slot_type% to the target position %slot%
@ECHO
################################################################################
@ECHO
################################################################################
pause
@ECHO
################################################################################
@ECHO
################################################################################
@ECHO running BitMan to relocate bitstream to target position
REM NOTE: make sure the X_MsL and X_MsL_1 being used for the slot_type MsL
@ECHO bitman.exe -x %X_MsL% %Y% %X_MsL_1% %Y_1% pacman_%slot_type%.bit -M
%X_slot% %Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
bitman.exe -x %X_MsL% %Y% %X_MsL_1% %Y_1% pacman_%slot_type%.bit -M %X_slot%
%Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
set bitfile=pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
@ECHO
################################################################################
@ECHO
################################################################################
@ECHO Press enter for loading a partial pacman module to slot %slot%
pause
call vivado -nolog -nojournal -notrace -mode batch -source load_bitstream.tcl
-quiet

REM Load the 5th pacman in slot 31 - MsM

REM defines slot's variables
@set slot=31
@set slot_type=MsM
REM X_slot = slot * 2; X_slot_1 = X_slot + 1
@set /a X_slot=%slot%*2
@set /a X_slot_1=%X_slot%+1

@ECHO
################################################################################
@ECHO
################################################################################
@ECHO Press enter for creating a partial pacman module to slot %slot%
@ECHO Slot %slot% has the resource footprint "%slot_type%" therefore, we
relocate bitstream
@ECHO pacman_%slot_type% to the target position %slot%
@ECHO
################################################################################
@ECHO
################################################################################
pause
@ECHO
################################################################################
@ECHO
################################################################################
```

```
@ECHO running BitMan to relocate bitstream to target position
REM NOTE: make sure the X_MsM and X_MsM_1 being used for the slot_type MsM
@ECHO bitman.exe -x %X_MsM% %Y% %X_MsM_1% %Y_1% pacman_%slot_type%.bit -M
%X_slot% %Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
bitman.exe -x %X_MsM% %Y% %X_MsM_1% %Y_1% pacman_%slot_type%.bit -M %X_slot%
%Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
set bitfile=pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
@ECHO
################################################################################
@ECHO
################################################################################
@ECHO Press enter for loading a partial pacman module to slot %slot%
pause
call vivado -nolog -nojournal -notrace -mode batch -source load_bitstream.tcl
-quiet

REM Load the 6th pacman in slot 27 - MsL

REM defines slot's variables
@set slot=27
@set slot_type=MsL
REM X_slot = slot * 2; X_slot_1 = X_slot + 1
@set /a X_slot=%slot%*2
@set /a X_slot_1=%X_slot%+1

@ECHO
################################################################################
@ECHO
################################################################################
@ECHO Press enter for creating a partial pacman module to slot %slot%
@ECHO Slot %slot% has the resource footprint "%slot_type%" therefore, we
relocate bitstream
@ECHO pacman_%slot_type% to the target position %slot%
@ECHO
################################################################################
@ECHO
################################################################################
pause
@ECHO
################################################################################
@ECHO
################################################################################
@ECHO running BitMan to relocate bitstream to target position
REM NOTE: make sure the X_MsL and X_MsL_1 being used for the slot_type MsL
@ECHO bitman.exe -x %X_MsL% %Y% %X_MsL_1% %Y_1% pacman_%slot_type%.bit -M
%X_slot% %Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
bitman.exe -x %X_MsL% %Y% %X_MsL_1% %Y_1% pacman_%slot_type%.bit -M %X_slot%
%Y% pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
set bitfile=pacman_%slot_type%_clk_%clock_region%_slot_%slot%.bit
@ECHO
################################################################################
@ECHO
################################################################################
@ECHO Press enter for loading a partial pacman module to slot %slot%
pause
call vivado -nolog -nojournal -notrace -mode batch -source load_bitstream.tcl
-quiet
```

```
REM Homework: Could you try to put more modules onto empty slots? We could
place 14 modules totally for this static design with 3 slot types LsM, MsM
and MsL.
REM Hint: Refer to the file ZedBoard_partial_slots.txt for other available
slots and their slot types, from slot 20 and afterwards.
REM NOTE: use the command "taskkill /IM vivado.exe /F" to kill the Vivado
process if it hangs due to failure of reconfiguration, usually due to wrong
bitstream settings.
```