**********Draft**********

Comparing SpinalHDL to Autofpga
When a PLL is required
Lesson Learned
03/22/21

**********Draft**********

The catzip project was created for the CatBoard using software packages (Autofpga, Icestorm, Yosys, Nextpnr, Verilator and ZipCPU).
The main clock on the Catboard is 100MHz. As the logic in FPGA is increased the speed needs to be reduced to obtain a working system. In the catzip.bin require a PLL to reduce the input freqency to 40Mhz.

The first steps in creating the catzip.bin was using Autofpga to generate a toplevel.v and main.v in the folder ~/testbuilds/catzip/rtl/catzip.

Autofpga requires a file ~/testbuilds/catzip/auto-data/clockpll40.txt with the following. Autofpga uses the tags

```
@$CLKFREQHZ=40000000
@PREFIX=clk
@CLOCK.NAME=clk
@TOP.DEFNS=
        wire            s_clk, s_reset;
@TOP.INSERT=
        assign  s_reset = 1'b0; // This design requires local, not global resets
@TOP.PORTLIST=i_clk
@TOP.IODECL= input  i_clk;
@REGDEFS.H.DEFNS=
#define         CLKFREQHZ@$CLKFREQHZ
@BDEF.DEFN=
#define         CLKFREQHZ@$CLKFREQHZ
@CLOCK.NAME=clk
@CLOCK.FREQUENCY= @$CLKFREQHZ
```

```
##
##
@$CLKFREQHZ=40000000
@PREFIX=clk
@CLOCK.NAME=clk
@TOP.DEFNS=
        wire            s_clk, s_reset;
@TOP.INSERT=
        assign          s_reset = 1'b0; // This design requires local, not global resets

`ifdef VERILATOR
        assign          s_clk = i_clk;
`else
        wire   clk_40mhz, pll_locked;
        SB_PLL40_CORE #(
```

```
                .FEEDBACK_PATH("SIMPLE"),
                .DELAY_ADJUSTMENT_MODE_FEEDBACK("FIXED"),
                .DELAY_ADJUSTMENT_MODE_RELATIVE("FIXED"),
                .PLLOUT_SELECT("GENCLK"),
                .FDA_FEEDBACK(4'b1111),
                .FDA_RELATIVE(4'b1111),
                .DIVR(4'b0100),          // DIVR =  4
                .DIVQ(7'b0011111),              // DIVQ =  31
                .DIVF(3'b100),           // DIVF =  4
                .FILTER_RANGE(3'b010)    // FILTER_RANGE = 2
        ) plli (
                .REFERENCECLK    (i_clk       ),
                .PLLOUTCORE    (clk_40mhz    ),
                .LOCK          (pll_locked  ),
                .BYPASS        (1'b0        ),
                .RESETB        (1'b1        )
        );
        //SB_GB global_buffer(clk_40mhz, s_clk);
        assign         s_clk = clk_40mhz;
`endif
@TOP.PORTLIST=i_clk
@TOP.IODECL= input  i_clk;
@REGDEFS.H.DEFNS=
#define      CLKFREQHZ    @$CLKFREQHZ
@BDEF.DEFN=
#define      CLKFREQHZ    @$CLKFREQHZ
@CLOCK.NAME=clk
@CLOCK.FREQUENCY= @$CLKFREQHZ
```

The file /home/devel/testbuilds/catzip/auto-data/Makefile requires the following:

```
DATA := global.txt bkram.txt buserr.txt clockpll40.txt          \
      pic.txt pwrcount.txt version.txt                          \
      hbconsole.txt gpio.txt dlyarbiter.txt zipbones.txt sdramdev.txt \
      mem_sdram_bkram.txt sdramscope.txt
```

/home/devel/icestorm/icefuzz/tests/sb_pll40_core.v

```
module top(
        input   REFERENCECLK,
        output  PLLOUTCORE,
        output  PLLOUTGLOBAL,
        input   EXTFEEDBACK,
```

```
        input   [7:0] DYNAMICDELAY,
        output  LOCK,
        input   BYPASS,
        input   RESETB,
        input   LATCHINPUTVALUE,

        //Test Pins
        output  SDO,
        input   SDI,
        input   SCLK
);
        SB_PLL40_CORE #(
                .FEEDBACK_PATH("DELAY"),
                // .FEEDBACK_PATH("SIMPLE"),
                // .FEEDBACK_PATH("PHASE_AND_DELAY"),
                // .FEEDBACK_PATH("EXTERNAL"),

                .DELAY_ADJUSTMENT_MODE_FEEDBACK("FIXED"),
                // .DELAY_ADJUSTMENT_MODE_FEEDBACK("DYNAMIC"),

                .DELAY_ADJUSTMENT_MODE_RELATIVE("FIXED"),
                // .DELAY_ADJUSTMENT_MODE_RELATIVE("DYNAMIC"),

                .PLLOUT_SELECT("GENCLK"),
                // .PLLOUT_SELECT("GENCLK_HALF"),
                // .PLLOUT_SELECT("SHIFTREG_90deg"),
                // .PLLOUT_SELECT("SHIFTREG_0deg"),

                .SHIFTREG_DIV_MODE(1'b0),
                .FDA_FEEDBACK(4'b1111),
                .FDA_RELATIVE(4'b1111),
                .DIVR(4'b0000),
                .DIVF(7'b0000000),
                .DIVQ(3'b001),
                .FILTER_RANGE(3'b000),
                .ENABLE_ICEGATE(1'b0),
                .TEST_MODE(1'b0)
        ) uut (
                .REFERENCECLK   (REFERENCECLK   ),
                .PLLOUTCORE     (PLLOUTCORE     ),
                .PLLOUTGLOBAL   (PLLOUTGLOBAL   ),
                .EXTFEEDBACK    (EXTFEEDBACK    ),
                .DYNAMICDELAY   (DYNAMICDELAY   ),
                .LOCK           (LOCK           ),
                .BYPASS         (BYPASS         ),
```

```
                .RESETB       (RESETB       ),
                .LATCHINPUTVALUE(LATCHINPUTVALUE),
                .SDO          (SDO          ),
                .SDI          (SDI          ),
                .SCLK         (SCLK         )
        );
endmodule
```

Using SpinalHDL to create pll Verilog.

The file /home/devel/SpinalKeyboardtest/src/main/scala/testcode/toplevel.scala is what determines the what is placed in the file /home/devel/SpinalKeyboardtest/toplevel_pll.v The first is used to separate the code into a package. The green highlted area are just comments.

```
package testcode

import spinal.core._
/*
`ifdef VERILATOR
        assign          s_clk = i_clk;
`else
        wire    clk_40mhz, pll_locked;
        SB_PLL40_CORE #(
                .FEEDBACK_PATH("SIMPLE"),
                .DELAY_ADJUSTMENT_MODE_FEEDBACK("FIXED"),
                .DELAY_ADJUSTMENT_MODE_RELATIVE("FIXED"),
                .PLLOUT_SELECT("GENCLK"),
                .FDA_FEEDBACK(4'b1111),
                .FDA_RELATIVE(4'b1111),
                .DIVR(4'b0100),             // DIVR =  4
                .DIVQ(7'b0011111),                  // DIVQ =  31
                .DIVF(3'b100),              // DIVF =  4
                .FILTER_RANGE(3'b010)       // FILTER_RANGE = 2
        ) plli (
                .REFERENCECLK     (i_clk        ),
                .PLLOUTCORE       (clk_40mhz    ),
                .LOCK             (pll_locked   ),
                .BYPASS           (1'b0         ),
                .RESETB           (1'b1         )
        );
        //SB_GB global_buffer(clk_40mhz, s_clk);
        assign          s_clk = clk_40mhz;
`endif
```

```
  SB_PLL40_CORE #(
    .FEEDBACK_PATH("SIMPLE"),
    .DIVR(4'b0010),
    .DIVF(7'h16),
    .DIVQ(3'b110),
    .FILTER_RANGE(3'b010)
  ) plli (
    .REFERENCECLK    (io_CLK_100       ), //i
    .PLLOUTCORE      (plli_PLLOUTCORE  ), //o
    .LOCK            (plli_LOCK        ), //o
    .BYPASS          (_zz_1            ), //i
    .RESETB          (_zz_2            )  //i
  );
  assign _zz_1 = 1'b0;
  assign _zz_2 = 1'b1;
*/

package testcode

import spinal.core._
case class SB_PLL40_CORE() extends BlackBox {
  val REFERENCECLK: Bool = in Bool
  val PLLOUTCORE, LOCK = out Bool
  val BYPASS = in Bits()
  val RESETB = in Bits()
  addGeneric("FEEDBACK_PATH", "SIMPLE")
  addGeneric("DELAY_ADJUSTMENT_MODE_FEEDBACK", "FIXED")
  addGeneric("DELAY_ADJUSTMENT_MODE_RELATIVE", "FIXED")
  addGeneric("PLLOUT_SELECT","GENCLK")
  addGeneric("FDA_FEEDBACK",B(15,4 bits))
  addGeneric("FDA_RELATIVE",B(15,4 bits))
  addGeneric("DIVR", B(0,4 bits))
  addGeneric("DIVQ", B(4,3 bits))
  addGeneric("DIVF", B(7,7 bits))

  addGeneric("FILTER_RANGE", B(5,3 bits))

  //addGeneric("BYPASS", B(0,1 bits))
  //addGeneric("RESETB", "RESET")
  //addGeneric("EXTFEEDBACK","()")
}
case class main() extends BlackBox {
        val iClk: Bool = in Bool
        val iRX: Bool = in Bool
        val oTX: Bool = out Bool
```

```
}
class toplevel_pll() extends Component {
  val io = new Bundle() {
  val CLK_100 = in Bool
  val uartRX = in Bool
  val uartTX = out Bool

  }
  val plli: SB_PLL40_CORE = new SB_PLL40_CORE()
  plli.REFERENCECLK <> io.CLK_100
  plli.BYPASS <> B(0,1 bit)
  plli.RESETB <> B(1,1 bit)

  val main = new main()
  main.iClk <> plli.PLLOUTCORE
  main.iRX <> io.uartRX
  main.oTX <> io.uartTX
}
//Generate the toplevel_pll Verilog
object toplevel_pllVerilog {
  def main(args: Array[String]) {
    SpinalVerilog(new toplevel_pll)
  }
}
```

Goal: To complete the SIMPLE pll using SpinalHDL. In addition to include a Blackbox to include main.v
a MyHDL modude which sends HELLOWORLD at 2MBaud. The PLL will reduce the 100MHz and main was reduced to
115200 Baud.

Status as of 03/22/21: The SIMPLE pll appears to be complete, but the HELLOWORLD at 115200 Baud is now
working. main.pcf -> toplevel_pll.pcf. Yosys, Nextpnr, icetime, and icepack are all working.
main require a active lo iRst which was provided by pwruprst.py

```
  wire      [0:0]   _zz_1;
  wire      [0:0]   _zz_2;
  wire              plli_PLLOUTCORE;
  wire              plli_LOCK;
  wire              main_1_oTX;

  SB_PLL40_CORE #(
```

```
    .FEEDBACK_PATH("SIMPLE"),
    .DELAY_ADJUSTMENT_MODE_FEEDBACK("FIXED"),
    .DELAY_ADJUSTMENT_MODE_RELATIVE("FIXED"),
    .PLLOUT_SELECT("GENCLK"),
    .FDA_FEEDBACK(4'b1111),
    .FDA_RELATIVE(4'b1111),
    .DIVR(4'b0000),
    .DIVQ(3'b100),
    .DIVF(7'h07),
    .FILTER_RANGE(3'b101)
  ) plli (
    .REFERENCECLK    (io_CLK_100       ), //i
    .PLLOUTCORE      (plli_PLLOUTCORE  ), //o
    .LOCK            (plli_LOCK        ), //o
    .BYPASS          (_zz_1            ), //i
    .RESETB          (_zz_2            )  //i
  );
  main main_1 (
    .iClk    (plli_PLLOUTCORE  ), //i
    .iRX     (io_uartRX        ), //i
    .oTX     (main_1_oTX       )  //o
  );
  assign _zz_1 = 1'b0;
  assign _zz_2 = 1'b1;
  assign io_uartTX = main_1_oTX;
```

Testing 100MHz and BAUDRATE 2e6

mainwkg1.png

mainwkg2.png

devel@mypi3-21:~/myhdl-relook/rs232tests $ sudo config_cat main.bin

mainwkg3.png

~/SpinalKeyboardtest/rs232-MyHDL-SpinalHDL

```
diff rs232sig.py ~/myhdl-relook/rs232tests/rs232sig.py
3,4c3,4
< Clk_f=50e6 #100 Mhz
< BAUDRATE=115200
---
> Clk_f=100e6 #100 Mhz
```

> BAUDRATE=2e6

In main.py uncoment
133 #convert_main(hdl='Verilog')

python main.py create main.v

~/SpinalKeyboardtest/
sbt "runMain testcode.toplevel_pllVerilog"

[info] welcome to sbt 1.3.13 (Raspbian Java 11.0.9.1)
[info] loading settings for project spinalkeyboardtest-build from plugins.sbt ...
[info] loading project definition from /home/devel/SpinalKeyboardtest/project
[info] loading settings for project spinalkeyboardtest from build.sbt ...
[info] set current project to SpinalTemplateSbt (in build file:/home/devel/SpinalKeyboardtest/)
[warn] Multiple main classes detected. Run 'show discoveredMainClasses' to see the list
[info] running (fork) testcode.toplevel_pllVerilog
[info] [Runtime] SpinalHDL v1.4.3   git head : adf552d8f500e7419fff395b7049228e4bc5de26
[info] [Runtime] JVM max memory : 944.0MiB
[info] [Runtime] Current date : 2021.03.22 06:27:57
[info] [Progress] at 0.000 : Elaborate components
[info] [Progress] at 0.361 : Checks and transforms
[info] [Progress] at 0.605 : Generate Verilog
[info] [Warning] 1 signals were pruned. You can call printPruned on the backend report to get more informations.
[info] [Done] at 0.912
[success] Total time: 7 s, completed Mar 22, 2021, 6:27:59 AM

~/SpinalKeyboardtest/rs232-MyHDL-SpinalHDL
rm -f toplevel_pll.v

cat ../toplevel_pll.v main.v > toplevel_pll.v

yosys -l simple.log -p 'synth_ice40 -abc9 -blif toplevel_pll.blif -json toplevel_pll.json' toplevel_pll.v
=== toplevel_pll ===

*   Number of wires: 319*
*   Number of wire bits: 526*
*   Number of public wires: 319*
*   Number of public wire bits: 526*
*   Number of memories: 0*

```
  Number of memory bits:          0
  Number of processes:            0
  Number of cells:              501
    SB_CARRY                    116
    SB_DFFE                      73
    SB_DFFER                     46
    SB_DFFESR                     4
    SB_DFFR                       4
    SB_DFFS                       1
    SB_DFFSR                     32
    SB_DFFSS                      2
    SB_LUT4                     222
    SB_PLL40_CORE                 1
```

```
nextpnr-ice40 --hx8k --pcf toplevel_pll.pcf --json toplevel_pll.json --asc
toplevel_pll.asc
```

```
icetime -d hx8k -c 50 toplevel_pll.asc
// Reading input .asc file..
// Reading 8k chipdb file..
// Creating timing netlist..
// Timing estimate: 9.39 ns (106.46 MHz)
// Checking 20.00 ns (50.00 MHz) clock constraint: PASSED.
```

```
icepack toplevel_pll.asc toplevel_pll.bin
```

```
md5sum toplevel_pll.bin
```

```
endmodule
```