

ASYNCHRONOUS FIFO-CLOCK GENERATION USING DCM

Clock Generation using Digital Clock Manager (DCM)

To generate the required clock for the FIFO **Digital Clock Manager (DCM)** wizard is used. DCM can implement a clock delay locked loop, a digital frequency synthesizer, digital phase shifter.

Below mentioned DCM ports are used in this design:

- **CLKIN:** CLKIN pin is the clock input to the DCM and is always enabled. CLKIN provides the source clock to the DCM. In this design frequency of 50 MHz (in Spartan 3 development board) is used as input to the CLKIN.
- **RST (RST_IN):** RST pin is the reset input to the DCM. If RST is not enabled then RST will be tied to GND.
- **CLKDV:** Divide output of the DCM is available at CLKDV pin. An option available in the 'Divide by Value' list determines the output clock CLKDV frequency. In this FIFO design input clock is divided by a divide value of 5 to get 10MHz write clock. Input clock of 50 MHz is directly taken as read clock.
- **CLKFX:** The CLKFX output pin provides fully digital, dedicated frequency synthesizer output to the DCM. The output frequency is a function of the input clock frequency described by M and D, where M is the multiplier (numerator), and D is the divisor (denominator). M and D are calculated for 100 MHz clock frequency. This output can be divided by 2 by using another DCM to get 50MHz of read clock. Instead of doing this, in design, input clock itself is directly used as read clock for the sake of simplicity.
- **CLK0:** Frequency output is same as CLKIN input. This output is also used for on-chip or off-chip synchronization.

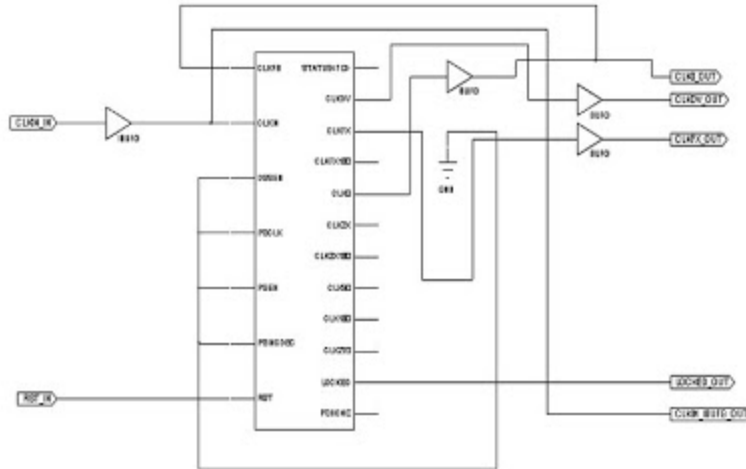


Figure (15) RTL schematic of DCM module

The RTL schematic of the DCM module is shown in the Figure (15). The test bench simulated waveform (post-translate model from Modelsim v.5.8 simulator launched by Xilinx ISE) for the DCM is shown in Figure (16).

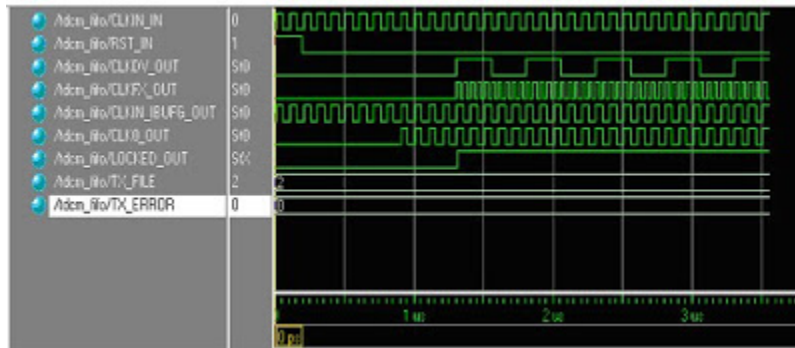


Figure (16) DCM simulation waveforms

Reset is held high for around 20 nS. Hence the CLKDV_OUT and CLKFX_OUT is zero. Then reset is made low. Out put is generated only after 10 input clock cycles. CLKFX_OUT is twice ($50\text{MHz} \times 2 = 100\text{MHz}$) and CLKDV_OUT is 5 times less ($50\text{MHz} / 5 = 10\text{MHz}$) than the input clock frequency (50 MHz).

FIFO with DCM: Synthesis and analysis

The verilog code generated for DCM by the architectural wizard is instantiated in the body of top level module `fifo_top.v`.

```

a_fifo5 a_fifo55(d_out,f_full_flag,f_half_full_flag,f_empty_flag,
f_almost_full_flag,f_almost_empty_flag,d_in,r_en,w_en,CLK0_OUT,CLKDV_OUT,reset);//ins
tantiate fifo

dcm_fifo dcm_fifo1(CLKIN_IN, RST_IN, CLKDV_OUT, CLKFX_OUT,
CLKIN_IBUFG_OUT,CLK0_OUT,
LOCKED_OUT); //instantiate DCM

```

FIFO code `a_fifo5.v` is also instantiated in the top level module. Binary counters are instantiated in the `a_fifo5.v` program. Thus total hierarchical structure of the design is as shown in the Figure (17). Top level module has instantiations to the modules `dcm_fifo`, `a_fifo5`. `dcm_fifo` module generates required clocks for the design while `a_fifo5` module implements FIFO memory controller. `B_counter` module is instantiated twice in `a_fifo5` module to obtain read and write address generators.

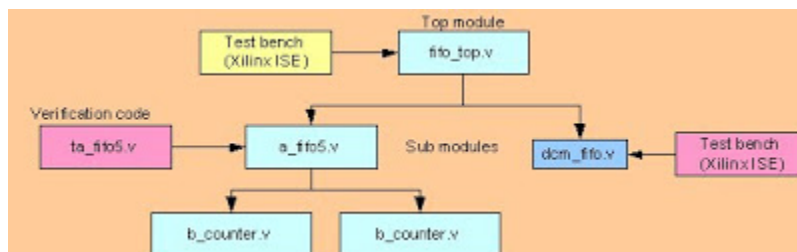


Figure (17) Hierarchical structure of the FIFO design

FIFO top level module `fifo_top.v` is verified by the help of test bench program generated by Xilinx ISE and behavioral simulation results are shown in the Figure (19). RTL schematic generated for the top module is shown in the Figure (18). The schematic includes both DCM module and FIFO module connected to each other as per the design requirement. Outputs `x`, `y` and `z` are from DCM module and are not used. The simulation results justify the FIFO working

without any error. Input clock CLKIN_IN itself becomes read clock. Since w_clk is generated internally it is not shown in the simulation waveform.

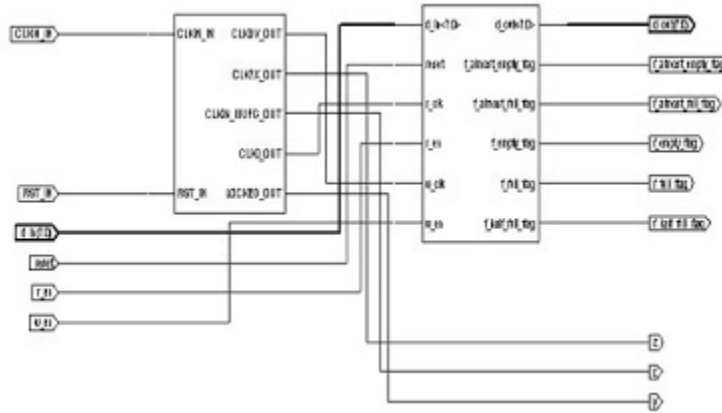


Figure (18) RTL schematic of DCM and FIFO

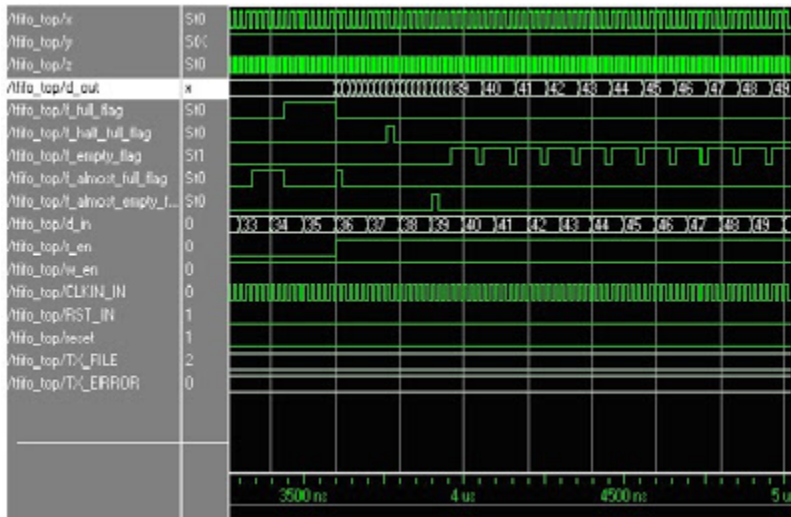


Figure (19) FIFO top module simulation waveform

The summary of the FPGA resources used for the complete design is listed below. The below mentioned device utilization summary is part of the synthesis report generated by Xilinx-synthesize-XST.

Device utilization summary:

Selected Device : 3s200ft256-5

Number of Slices: 28 out of 1920 1%

Number of Slice Flip Flops: 23 out of 3840 0%

Number of 4 input LUTs: 54 out of 3840 1%

Number used as logic: 38

Number used as RAMs: 16

Number of IOs: 29

Number of bonded IOBs: 29 out of 173 16%

Number of GCLKs: 3 out of 8 37%

Number of DCMs: 1 out of 4 25%

Timing Summary:

Speed Grade: -5

Minimum period: 5.091ns (Maximum Frequency: 196.444MHz)

Minimum input arrival time before clock: 6.185ns

Maximum output required time after clock: 11.059ns

Maximum combinational path delay: 6.662ns

From the timing summary it can be observed that maximum operating frequency is increased from around 110 MHz to 196.444 MHz. this can be attributed to the DCM which provides stable clock signal to the hardware resources. 16 numbers of LUTs are utilized as distributed dual port RAM. Synthesis of a_fifo5 showed around 35 numbers of slices. But with the complete implementation it has reduced to 28. Total numbers of flip flops used also reduced.

Merits and demerits of the design

One of the important advantages of the proposed design is that the design shows efficient performance for the synchronous as well as asynchronous clocks (frequency within the maximum operating frequency). All status flags are asserted and deasserted with zero clock cycle delay. The new design uses simple 4 bit binary counters for addressing the FIFO memory. Synchronization between the clock domains is achieved with the pointer difference concept which is very easy to understand and implement. But the design can be slower compared to the design of FIFO with gray pointer approach. But this has to be tested.

Source : <http://asic-soc.blogspot.in/2007/12/asynchronous-fifo-clock-generation.html>