This tutorial shows how to create a simple project using a DCM (Digital Clock Manager).

The DCM generates 25MHz signals from the 100MHz xtal clock connected to the FPGA.

Create a simple module with the following ports:

```verilog
module DCM_example_Verilog(
    input fpga_clk,
    input reset,
    output lock_led,
    output counter_led
);

reg [3:0] count;
wire clk_25M;

// Count from 0 to 9 with a frequency of 25MHz
always @ (posedge clk_25M or posedge reset)
begin
    if (reset)
        count <= 4'b0;
    else
        if (count == 4'b1001)
            count <= 4'b0;
        else
            count <= count + 1'b1;
end

assign counter_led = (count == 9) ? 1'b1:1'b0;
endmodule
```

Add a new IP source:

Call it dcm_25 (for 25MHz DCM)
Next and Select the Clocking Wizard core:

Click Next and then Finish.
Change the output frequency of CLK_OUT1 to 25MHz.

Click Next to go through next few pages and click Generate.
You can now see the dcm_25 in the Design pane,

Select it and you will see some options in the Processes pane.

Select the View HDL Instantiation Template
Double-click the View HDL Instantiation Template process:

```verilog
65 // core to be instantiated. Change the instance name and port cc:
66 // (in parentheses) to your own signal names.
67
68 //--------- Begin Cut here for INSTANTIATION Template ---// INSTANTIATION Template
69
70 dcm_25  instance name
71 (// Clock in ports
72 .CLK_IN1(CLK_IN1), // IN
73 // Clock out ports
74 .CLK_OUT1(CLK_OUT1), // OUT
75 // Status and control signals
76 .RESET(RESET), // IN
77 .LOCKED(LOCKED)); // OUT
78 // INST_TAG_END ------- End INSTANTIATION Template ---------------
```
Copy and Paste the DCM instantiation templates to your original Verilog codes:

```verilog
module DCM_example_Verilog(
    input fpga_clk,
    input reset,
    output lock_led,
    output counter_led
);

reg [3:0] count;
wire clk_2SM;

dcm_25 instance_name
    (.
        CLK_INI(fpga_clk), // 100Mhz xtal clock to FPGA;
        CLK_OUT1(clk_2SM), // 25Mhz clock for the counter
        RESET(reset),
        LOCKED(lock_led)); // light is on when DCM is locked

// Count from 0 to 9 with a frequency of 25Mhz
always @ (posedge clk_2SM or posedge reset)
begin
    if (reset)
        count <= 4'b0;
    else
        if (count == 4'b1001)
            count <= 4'b0;
        else
            count <= count + 1'b1;
end

assign counter_led = (count == 9)? 1'b1:1'b0;
endmodule
```
Create a UCF File

![Image of UCF file]

Synthesize, Implement, and Generate a Programming File.

Here is a schematic of what we have created: