

## 1. 新建一个顶层文件

此电脑 > 桌面 > V3_FPGA_2018 > enhance > 20180306 > s6_ddr3_ctrl > rtl				
名称	修改日期	类型	大小	
s6_ddr3_ctrl_top.v	2018/3/11 15:17	V 文件	0 KB	

## 2. 实例化 p11 ip 核

```
//-----p11 实例化-----  
clk_333M clk_333M_inst(  
    .clk_50Mhz (clk),          // IN 系统时钟用于锁相环的输入后就不能再驱动其他信号  
  
    .clk_333Mhz (clk_333Mhz), // OUT  
    .clk_100Mhz (clk_100Mhz)  // OUT  
);
```

## 3. 实例化 s6\_ddr3 ip 核 (直接使用例化模板)

<< 桌面 > V3_FPGA_2018 > enhance > 20180306 > s6_ddr3_ctrl > ise_prj > ipcore_dir > ddr3_ctrl_mig > user_design > rtl >				
名称	修改日期	类型	大小	
mcb_controller	2018/3/11 15:11	文件夹		
ddr3_ctrl_mig.v	2018/3/11 15:11	V 文件	39 KB	
infrastructure.v	2013/10/14 2:17	V 文件	11 KB	
memc_wrapper.v	2013/10/14 2:17	V 文件	65 KB	

```
ddr3_ctrl_mig #(  
    .C1_P0_MASK_SIZE(4),  
    .C1_P0_DATA_PORT_SIZE(32),  
    .C1_P1_MASK_SIZE(4),  
    .C1_P1_DATA_PORT_SIZE(32),  
    .DEBUG_EN(0),  
    .C1_MEMCLK_PERIOD(3000),  
    .C1_CALIB_SOFT_IP("TRUE"),  
    .C1_SIMULATION("FALSE"),  
    .C1_RST_ACT_LOW(0),  
    .C1_INPUT_CLK_TYPE("SINGLE_ENDED"),  
    .C1_MEM_ADDR_ORDER("ROW_BANK_COLUMN"),  
    .C1_NUM_DQ_PINS(16),  
    .C1_MEM_ADDR_WIDTH(14),  
    .C1_MEM_BANKADDR_WIDTH(3)  
)  
ddr3_ctrl_mig_inst (  
  
    .c1_sys_clk (c1_sys_clk),  
    .c1_sys_rst_i (c1_sys_rst_i),  
  
    .mcb1_dram_dq (mcb1_dram_dq),
```

```
.mcb1_dram_a      (mcb1_dram_a),
.mcb1_dram_ba     (mcb1_dram_ba),
.mcb1_dram_ras_n  (mcb1_dram_ras_n),
.mcb1_dram_cas_n  (mcb1_dram_cas_n),
.mcb1_dram_we_n   (mcb1_dram_we_n),
.mcb1_dram_odt    (mcb1_dram_odt),
.mcb1_dram_cke    (mcb1_dram_cke),
.mcb1_dram_ck     (mcb1_dram_ck),
.mcb1_dram_ck_n   (mcb1_dram_ck_n),
.mcb1_dram_dqs    (mcb1_dram_dqs),
.mcb1_dram_dqs_n  (mcb1_dram_dqs_n),
.mcb1_dram_udqs   (mcb1_dram_udqs),    // for X16 parts
.mcb1_dram_udqs_n (mcb1_dram_udqs_n),  // for X16 parts
.mcb1_dram_udm    (mcb1_dram_udm),    // for X16 parts
.mcb1_dram_dm     (mcb1_dram_dm),
.mcb1_dram_reset_n (mcb1_dram_reset_n),

.c1_clk0          (c1_clk0),
.c1_rst0          (c1_rst0),
.c1_calib_done    (c1_calib_done),
.mcb1_rzq         (rzq1),
.mcb1_zio         (zio1),

.c1_p2_cmd_clk    (c1_p2_cmd_clk),
.c1_p2_cmd_en     (c1_p2_cmd_en),
.c1_p2_cmd_instr  (c1_p2_cmd_instr),
.c1_p2_cmd_b1     (c1_p2_cmd_b1),
.c1_p2_cmd_byte_addr (c1_p2_cmd_byte_addr),
.c1_p2_cmd_empty  (c1_p2_cmd_empty),
.c1_p2_cmd_full   (c1_p2_cmd_full),

.c1_p2_wr_clk     (c1_p2_wr_clk),
.c1_p2_wr_en      (c1_p2_wr_en),
.c1_p2_wr_mask    (c1_p2_wr_mask),
.c1_p2_wr_data    (c1_p2_wr_data),
.c1_p2_wr_full    (c1_p2_wr_full),
.c1_p2_wr_empty   (c1_p2_wr_empty),
.c1_p2_wr_count   (c1_p2_wr_count),
.c1_p2_wr_underrun (c1_p2_wr_underrun),
.c1_p2_wr_error   (c1_p2_wr_error),

.c1_p3_cmd_clk    (c1_p3_cmd_clk),
.c1_p3_cmd_en     (c1_p3_cmd_en),
.c1_p3_cmd_instr  (c1_p3_cmd_instr),
```

```
.c1_p3_cmd_bl      (c1_p3_cmd_bl),
.c1_p3_cmd_byte_addr(c1_p3_cmd_byte_addr),
.c1_p3_cmd_empty    (c1_p3_cmd_empty),
.c1_p3_cmd_full     (c1_p3_cmd_full),

.c1_p3_rd_clk      (c1_p3_rd_clk),
.c1_p3_rd_en       (c1_p3_rd_en),
.c1_p3_rd_data     (c1_p3_rd_data),
.c1_p3_rd_full     (c1_p3_rd_full),
.c1_p3_rd_empty    (c1_p3_rd_empty),
.c1_p3_rd_count    (c1_p3_rd_count),
.c1_p3_rd_overflow  (c1_p3_rd_overflow),
.c1_p3_rd_error    (c1_p3_rd_error)
);

复制修改上面的文件得到
//-----ddr3_ctrl_mig 实例化-----
ddr3_ctrl_mig #(
    .C1_P0_MASK_SIZE(4),
    .C1_P0_DATA_PORT_SIZE(32),
    .C1_P1_MASK_SIZE(4),
    .C1_P1_DATA_PORT_SIZE(32),
    .DEBUG_EN(0),
    .C1_MEMCLK_PERIOD(3000),
    .C1_CALIB_SOFT_IP("TRUE"),
    .C1_SIMULATION("FALSE"),
    .C1_RST_ACT_LOW(1), //该参数控制复位信号的有效电平，为 1 时低复位，为 0 时高复
位，所以设置为 1
    .C1_INPUT_CLK_TYPE("SINGLE_ENDED"),
    .C1_MEM_ADDR_ORDER("ROW_BANK_COLUMN"),
    .C1_NUM_DQ_PINS(16),
    .C1_MEM_ADDR_WIDTH(14),
    .C1_MEM_BANKADDR_WIDTH(3)
)
ddr3_ctrl_mig_inst(
    .c1_sys_clk      (clk_333MHz),
    .c1_sys_rst_i    (rst_n),

    //控制外部 ddr3 芯片的输出信号
    .mcb1_dram_dq     (mcb1_dram_dq),
    .mcb1_dram_a      (mcb1_dram_a),
    .mcb1_dram_ba     (mcb1_dram_ba),
    .mcb1_dram_ras_n  (mcb1_dram_ras_n),
    .mcb1_dram_cas_n  (mcb1_dram_cas_n),
    .mcb1_dram_we_n   (mcb1_dram_we_n),
```

```
.mcb1_dram_odt      (mcb1_dram_odt),
.mcb1_dram_cke      (mcb1_dram_cke),
.mcb1_dram_ck       (mcb1_dram_ck),
.mcb1_dram_ck_n     (mcb1_dram_ck_n),
.mcb1_dram_dqs      (mcb1_dram_dqs),
.mcb1_dram_dqs_n    (mcb1_dram_dqs_n),
.mcb1_dram_udqs     (mcb1_dram_udqs),    // for X16 parts
.mcb1_dram_udqs_n   (mcb1_dram_udqs_n),  // for X16 parts
.mcb1_dram_udm      (mcb1_dram_udm),    // for X16 parts
.mcb1_dram_dm       (mcb1_dram_dm),
.mcb1_dram_reset_n  (mcb1_dram_reset_n),

.clk_clk0           (clk_clk0), //由 ip 核内部产生, 相当 a7 中的 ui_clk
.clk_rst0           (clk_rst0), //由 ip 核内部产生, 相当 a7 中的
ui_clk_sync_rst
.clk_calib_done      (clk_calib_done), //ddr3 ip 核是否初始化成功, 为 1
时表示成功
.mcb1_rzq            (mcb1_rzq), //设置终端电阻
.mcb1_zio            (mcb1_zio), //设置终端电阻

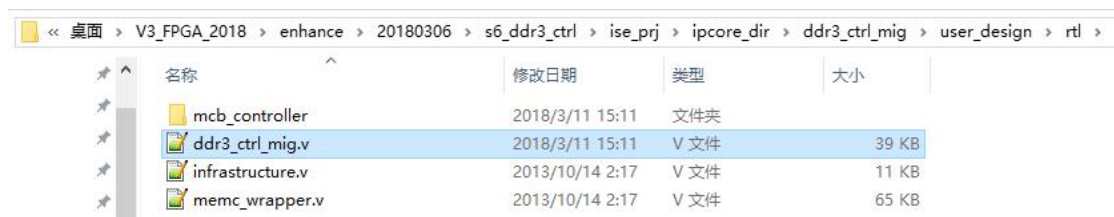
//用户控制 ddr3 ip 核的输入信号
.clk_p2_cmd_clk      (clk_100Mhz),
.clk_p2_cmd_en       (clk_p2_cmd_en),
.clk_p2_cmd_instr    (3'b000), //配置时该指令只能为写指令, 指令参考数据手册
27 页
.clk_p2_cmd_b1       (clk_p2_cmd_b1), //突发长度 Burst length
.clk_p2_cmd_byte_addr (clk_p2_cmd_byte_addr), //数据存储的起始地址
.clk_p2_cmd_empty     (clk_p2_cmd_empty),
.clk_p2_cmd_full      (clk_p2_cmd_full),

.clk_p2_wr_clk       (clk_100Mhz),
.clk_p2_wr_en        (clk_p2_wr_en),
.clk_p2_wr_mask      (clk_p2_wr_mask),
.clk_p2_wr_data      (clk_p2_wr_data),
.clk_p2_wr_full      (clk_p2_wr_full),
.clk_p2_wr_empty     (clk_p2_wr_empty),
.clk_p2_wr_count     (clk_p2_wr_count),
.clk_p2_wr_underrun   (clk_p2_wr_underrun), //
.clk_p2_wr_error      (clk_p2_wr_error),

.clk_p3_cmd_clk      (clk_100Mhz),
.clk_p3_cmd_en       (clk_p3_cmd_en),
.clk_p3_cmd_instr    (3'b001), //配置时该指令只能为读指令
.clk_p3_cmd_b1       (clk_p3_cmd_b1),
```

```
.c1_p3_cmd_byte_addr(c1_p3_cmd_byte_addr),  
.c1_p3_cmd_empty    (c1_p3_cmd_empty),  
.c1_p3_cmd_full     (c1_p3_cmd_full),  
  
.c1_p3_rd_clk       (clk_100Mhz),  
.c1_p3_rd_en        (c1_p3_rd_en),  
.c1_p3_rd_data      (c1_p3_rd_data),  
.c1_p3_rd_full      (c1_p3_rd_full),  
.c1_p3_rd_empty     (c1_p3_rd_empty),  
.c1_p3_rd_count     (c1_p3_rd_count),  
.c1_p3_rd_overflow  (c1_p3_rd_overflow),  
.c1_p3_rd_error     (c1_p3_rd_error)  
);
```

#### 4. 顶层的输出信号



名称	修改日期	类型	大小
mcb_controller	2018/3/11 15:11	文件夹	
ddr3_ctrl_mig.v	2018/3/11 15:11	V 文件	39 KB
infrastructure.v	2013/10/14 2:17	V 文件	11 KB
memc_wrapper.v	2013/10/14 2:17	V 文件	65 KB

```
inout  [C1_NUM_DQ_PINS-1:0]    mcb1_dram_dq,  
output [C1_MEM_ADDR_WIDTH-1:0] mcb1_dram_a,  
output [C1_MEM_BANKADDR_WIDTH-1:0] mcb1_dram_ba,  
output                                mcb1_dram_ras_n,  
output                                mcb1_dram_cas_n,  
output                                mcb1_dram_we_n,  
output                                mcb1_dram_odt,  
output                                mcb1_dram_reset_n,  
output                                mcb1_dram_cke,  
output                                mcb1_dram_dm,  
inout                                mcb1_dram_udqs,  
inout                                mcb1_dram_udqs_n,  
inout                                mcb1_rzq,  
inout                                mcb1_zio,  
output                                mcb1_dram_udm,  
input                                c1_sys_clk,  
input                                c1_sys_rst_i,  
output                                c1_calib_done,  
output                                c1_clk0,  
output                                c1_rst0,  
inout                                mcb1_dram_dqs,  
inout                                mcb1_dram_dqs_n,  
output                                mcb1_dram_ck,
```

```
output                                mcb1_dram_ck_n
```

复制修改上面的文件得到

//控制外部 ddr3 芯片的输出信号

```
inout wire    [15:0] mcb1_dram_dq,
output wire    [13:0] mcb1_dram_a,
output wire    [2:0]  mcb1_dram_ba,
output wire                    mcb1_dram_ras_n,
output wire                    mcb1_dram_cas_n,
output wire                    mcb1_dram_we_n,
output wire                    mcb1_dram_odt,
output wire                    mcb1_dram_reset_n,
output wire                    mcb1_dram_cke,
output wire                    mcb1_dram_dm,
inout wire                    mcb1_dram_udqs,
inout wire                    mcb1_dram_udqs_n,
inout wire                    mcb1_rzq,
inout wire                    mcb1_zio,
output wire                    mcb1_dram_udm,
inout wire                    mcb1_dram_dqs,
inout wire                    mcb1_dram_dqs_n,
output wire                    mcb1_dram_ck,
output wire                    mcb1_dram_ck_n
```

## 5. 在顶层文件中定义变量

```
wire    c1_clk0;
wire    c1_rst0;
wire    clk_333Mhz;
wire    clk_100Mhz;
wire    c1_calib_done;
```

## 6. 新建一个 tb 文件



## 7. 在 tb 中实例化顶层文件并连接端口线

```
s6_ddr3_ctrl_top s6_ddr3_ctrl_top_inst(
    .clk                (clk),
    .rst_n              (rst_n),

    .mcb1_dram_dq        (mcb1_dram_dq),
    .mcb1_dram_a         (mcb1_dram_a),
    .mcb1_dram_ba        (mcb1_dram_ba),
```

```
.mcb1_dram_ras_n (mcb1_dram_ras_n),
.mcb1_dram_cas_n (mcb1_dram_cas_n),
.mcb1_dram_we_n (mcb1_dram_we_n),
.mcb1_dram_odt (mcb1_dram_odt),
.mcb1_dram_reset_n(mcb1_dram_reset_n),
.mcb1_dram_cke (mcb1_dram_cke),
.mcb1_dram_dm (mcb1_dram_dm),
.mcb1_dram_udqs (mcb1_dram_udqs),
.mcb1_dram_udqs_n (mcb1_dram_udqs_n),
.mcb1_rzq (mcb1_rzq),
.mcb1_zio (mcb1_zio),
.mcb1_dram_udm (mcb1_dram_udm),
.mcb1_dram_dqs (mcb1_dram_dqs),
.mcb1_dram_dqs_n (mcb1_dram_dqs_n),
.mcb1_dram_ck (mcb1_dram_ck),
.mcb1_dram_ck_n (mcb1_dram_ck_n)
);
```

## 8. 在 tb 文件中实例化 ddr3 仿真模型文件并连接端口线



```
ddr3_model_c1(
    rst_n,
    ck,
    ck_n,
    cke,
    cs_n,
    ras_n,
    cas_n,
    we_n,
    dm_tdqs,
    ba,
    addr,
    dq,
    dqs,
    dqs_n,
    tdqs_n,
    odt
);
```

复制修改上面的文件得到

```
//-----ddr3 仿真模型 实例化-----  
ddr3_model_c1 ddr3_model_c1 (  
    .rst_n (mcb1_dram_reset_n),  
    .ck     (mcb1_dram_ck),  
    .ck_n   (mcb1_dram_ck_n),  
    .cke     (mcb1_dram_cke),  
    .cs_n    (1'b0),  
    .ras_n   (mcb1_dram_ras_n),  
    .cas_n   (mcb1_dram_cas_n),  
    .we_n    (mcb1_dram_we_n),  
    .dm_tdqs ({mcb1_dram_udm,mcb1_dram_dm}),  
    .ba      (mcb1_dram_ba),  
    .addr    (mcb1_dram_a),  
    .dq      (mcb1_dram_dq),  
    .dqs     ({mcb1_dram_udqs,mcb1_dram_dqs}),  
    .dqs_n   ({mcb1_dram_udqs_n,mcb1_dram_dqs_n}),  
    .tdqs_n (),  
    .odt     (mcb1_dram_odt)  
);
```

注意：在信号连线时找不到的信号参考例程设计



## 9. 在 tb 文件中定义变量

```
wire [15:0] mcb1_dram_dq;  
wire [13:0] mcb1_dram_a;  
wire [2:0]  mcb1_dram_ba;  
wire       mcb1_dram_ras_n;  
wire       mcb1_dram_cas_n;  
wire       mcb1_dram_we_n;  
wire       mcb1_dram_odt;  
wire       mcb1_dram_reset_n;  
wire       mcb1_dram_cke;  
wire       mcb1_dram_dm;  
wire       mcb1_dram_udqs;  
wire       mcb1_dram_udqs_n;  
wire       mcb1_rzq;  
wire       mcb1_zio;
```



获取更多资料参与技术交流欢迎加入 QQ 群:450843130

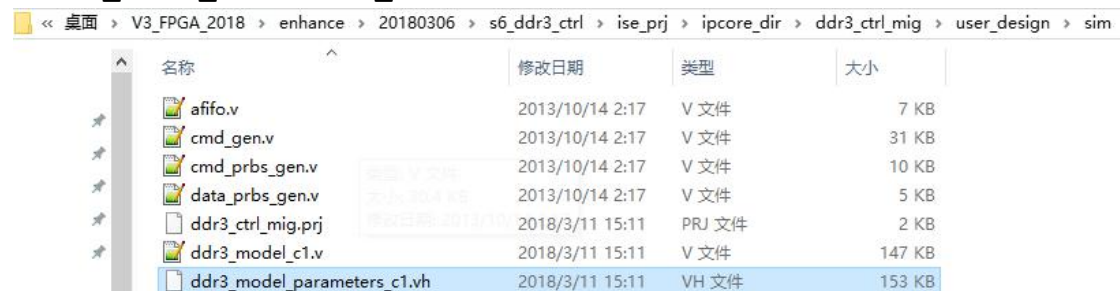
```
wire            mcb1_dram_udm;
wire            mcb1_dram_dqs;
wire            mcb1_dram_dqs_n;
wire            mcb1_dram_ck;
wire            mcb1_dram_ck_n;
```

#### 10. 初始化时钟和复位，并生成时钟

```
initial    begin
    clk = 1'b1;
    rst_n <= 1'b0;
    #200;
    rst_n <= 1'b1;
end

always #10 clk = ~clk;
```

#### 11. 在 ISE 工程中添加 ddr3 仿真模型和 ddr3 仿真模型的初始化参数文件 (ddr3\_model\_parameters\_c1.vh 要放在工程目录下，不然找不到)



名称	修改日期	类型	大小
afifo.v	2013/10/14 2:17	V 文件	7 KB
cmd_gen.v	2013/10/14 2:17	V 文件	31 KB
cmd_prbs_gen.v	2013/10/14 2:17	V 文件	10 KB
data_prbs_gen.v	2013/10/14 2:17	V 文件	5 KB
ddr3_ctrl_mig.prj	2018/3/11 15:11	PRJ 文件	2 KB
ddr3_model_c1.v	2018/3/11 15:11	V 文件	147 KB
ddr3_model_parameters_c1.vh	2018/3/11 15:11	VH 文件	153 KB

#### 12. 启动仿真观察 c1\_calib\_done 信号是否拉高(0.48ms 左右)，如果拉高说明初始化成功