



基于 VIVADO2022.2 版本 应用教程

摘要

远赴人间惊鸿宴，一睹人间盛世颜。最是人间留不住，朱颜辞镜花辞树

SUN

[电子邮件地址]

文档修改记录

[illegible]

目录

1	概述	3
2	软件设置	4
	Vivado 文本编辑器设置	4
	CPU 多线程 (WIN)	4
3	名词解释 (FPGA/软件)	5
	时钟	5
	仿真	6
4	名词解释 (接口相关)	8
	DDR	8
	HDMI	8
	VCU	8
	ETH	8
	PCIE	9
5	应用技巧	10
	软件及对应第三方软件版本	10
	DCP 生成步骤	10
	DCP 文件仿真	10
	EDIF 生成步骤	10
	编辑和修改官方 IP 核	10
	BIT 文件校验与回读	11
	增量式编译	11
	Vivado Store	13
	代码加密	13
	将代码文件直接添加到 BD	13
	Tcl 脚本配置 IP 方式	14
	Tcl 脚本下载程序	14
6	官方 IP 使用说明	15
	RAM 初始化	15
	MIG 配置	15
7	个人 IP 使用说明	21
8	硬件接口设计	21

1 概述

本文使用软件版本：Vivado 2022.2，包含 Tcl 示例工程。ISE 或者其它低版本 Vivado，不推荐参考某些工程示例或工程！

本文内容基于 Xilinx 官方文档总结或参考。

2 软件设置

Vivado 文本编辑器设置

常用 notepad++ 和 VS code

vivado->settings ->Text Editor -> Custom Editor

path/notepad++.exe [file name] -n[line number]

e.g.

D:\Program Files\Notepad++\notepad++.exe [file name] -n[line number]

D:\Programs\Microsoft VS Code\Code.exe [file name] -n[line number]

CPU 多线程 (WIN)

建立文件并命名 Vivado_init.tcl

内容: set_param general.maxThreads 8

将 tcl 文件放入...\Vivado\2022.2\scripts 文件夹

3 名词解释（FPGA/软件）

时钟

1, 时钟概述

CMT: 时钟管理块, 包含一个 MMCM 和一个 PLL

MMCM: 混合模式时钟管理器

PLL: 锁相环

BUFG: 全局时钟线, 7 系每个片子有 32 条

BUFH/BUFHCE: 水平时钟缓冲器, 允许通过水平时钟行访问单个时钟区域中的全局时钟线。它还可以用作时钟使能电路 (BUFHCE), 以独立启用或禁用跨越单个时钟区域的时钟

使用每个时钟区域中的 12 条水平时钟线, 每个时钟区域最多可支持 12 个时钟。

BUFMR: 多时钟区域缓冲器, 允许区域和 I/O 时钟跨越多达三个垂直相邻的时钟区域。

CMT 概述:

每个 7 系列 FPGA 最多有 24 个 CMT, 每个 CMT 由一个 MMCM 和一个 PLL 组成。MMCM 和 PLL 用作各种频率的频率合成器, 用作外部或内部时钟的抖动滤波器, 以及去歪斜时钟。该锁相环包含 MMCM 功能的子集。7 系列 FPGA 时钟输入连接允许多种资源为 MMCM 和 PLL 提供参考时钟。

MMCM 在任一方向具有无限精细相移能力, 可用于动态相移模式。MMCM 在反馈路径或一个输出路径中也有一个小数计数器, 从而实现频率合成能力的进一步粒度

MRCC: MRCC 用于本时钟区域和相邻时钟区域

SRCC: 本区域时钟区域

两者都可以连接到全局时钟

BUFIO: IO buffer

2, 时钟管理单元

CMT

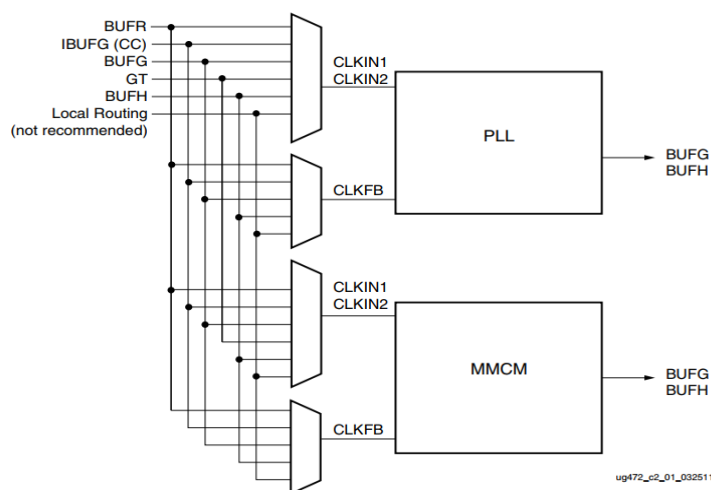


Figure 3-1: Block Diagram of the 7 Series FPGAs CMT

MMCM

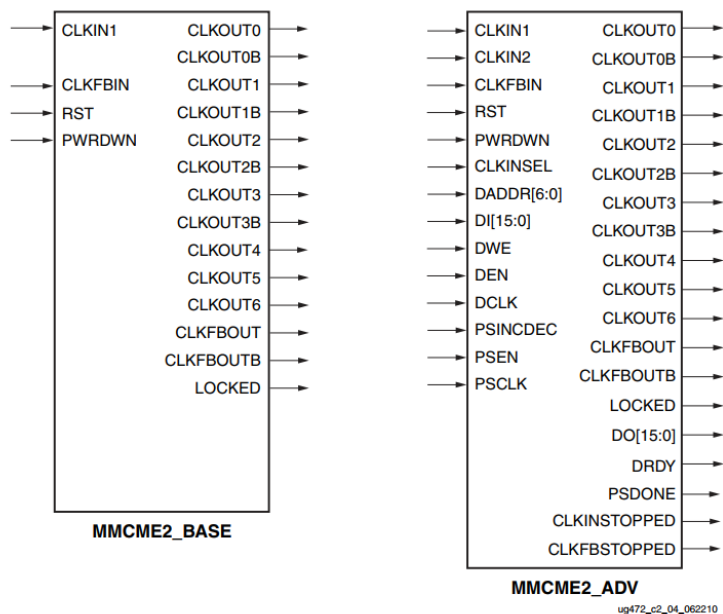


Figure 3-4: MMCM Primitives

PLL

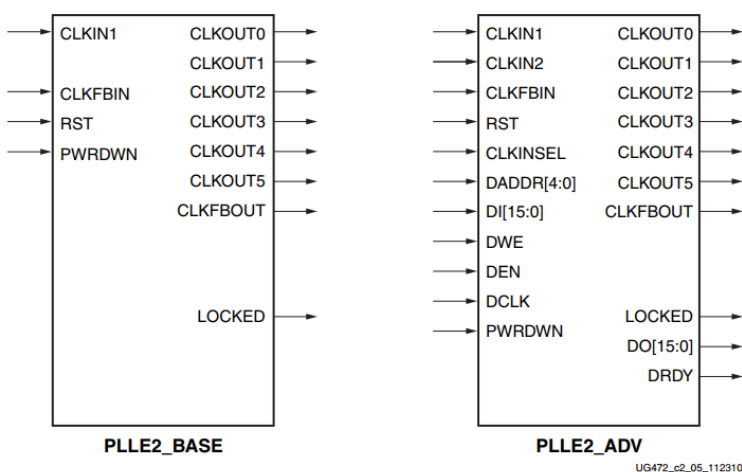


Figure 3-5: PLL Primitives

3, 时钟支持动态重配, 支持 AXI lite 和 DRP 接口
但并不常用, 参考示例 XAPP888

仿真

仿真类型

1, 仿真含三种

RTL 级

一般都是逻辑代码

Post-Synthesis Simulation

可以对网表进行仿真

Post-Implementation Simulation

实现后执行功能或时序仿真。时序仿真是最接近实际将设计下载到设备的仿真

2, 编译库时, GCC executable path 为 GCC 路径, 用于 SystemC IP。可以忽略!

3, 仿真库部分解释

UNISIM: 原语功能仿真

UNIMACRO: 宏功能仿真

UNIFAST: 快速仿真库

SIMPRIM: 原语时序仿真

SECUREIP: 用于功能和时序仿真库, Xilinx 器件特性仿真, 例如 PCIE, 收发器等

XPM: 原语的功能仿真

4 名词解释（接口相关）

DDR

HDMI

VCU

ETH

分层概念

[OSI 模型](#)

[Internet protocol suite](#)

[VLAN](#)

[Precision Time Protocol \(IEEE 1588\)](#)

[POE](#)

帧概念相关

[以太网帧](#)

[巨型帧](#)

IPv4/UDP/ICMP/ARP/FCS

[UDP lite](#)

总结

- 1, 严格来说，整个帧（前导码到 FCS）都属于数据链路层范围！
- 2, 数据帧指的是 IP 数据，不包含首部和尾部之这 18 个字节。
常规帧长度 64-1518，所以数据帧的长度在 46-1500 字节之间
- 3, IP 头为 20byte，所以 IP 数据区最长 1480。
- 4, 如果 TCP 或 UDP 包数据长度超过 1500，，系统进行分包操作。
- 5, UDP 校验和计算对于 IPv4 是可选的。如果不使用校验和，则应将其设置为零值
- 6, TCP 整个包的最大长度是由最大传输大小决定。UDP 整个包的最大长度为 65535

TCP

整个包的最大长度是由最大传输大小（MSS, Maximum Segment Size）决定，MSS 就是 TCP 数据包每次能够传输的最大数据分段

UDP

从理论上说，包含报头在内的数据报的最大长度为 65535 字节 (64K)

- 7, ICMP 属于网络层协议

帧组成示例

以 TCP 和 UDP (IPV4) 为例，，帧最少 64 字节，最高 1518 字节

我们知道数据帧长度 46 - 1500。即

$46 = 64 - 18$ （首部+FCS）

$1500 = 1518 - 18$ （首部 +FCS）

以帧最低长度 64 为例

TCP

首部（14byte）+ IP 头（20byte）+ TCP 头（20byte）+ payload（6byte）+ FCS（4byte）

UDP

首部 (14byte) + IP 头 (20byte) + UDP 头 (8byte) + payload (18byte) + FCS (4byte)

这就出现一个问题，假如 TCP 或 UDP 传输少于 payload 最低字节数，就需要填充 0 补充到最低字节

关于 wireshark

PC 接收数据：

物理层的网卡会去掉前导码和帧开始符，然后对帧进行 CRC 校验，如果帧校验错误，丢弃此帧！如果校验和正确，就判断帧的目的硬件地址是否符合自己接收条件（目的地址是自己的物理硬件地址、广播地址、可接收的多播硬件地址等），如果符合，就将帧交“设备驱动程序”做进一步处理。这时我们的抓包软件才能抓到数据，因此，抓包软件抓到的是去掉前导同步码、帧开始分界符、FCS 之外的数据

PC 发送数据：

Wireshark 是可以抓到 FCS 的。

为什么会出现接收和发送，wireshark 有无 FCS 的区别呢？

Wireshark，IP 头校验和错误

这是由于网卡属性->高级->硬件校验和的关系。默认 RXTX 开启，由网卡计算。

PCIE

SG (scatter/gather) 模式

scatter/gather 指的在多个缓冲区上实现一个简单的 I/O 操作，比如从通道中读取数据到多个缓冲区，或从多个缓冲区中写入数据到通道；

scatter（分散）：指的是从通道中读取数据分散到多个缓冲区 Buffer 的过程，该过程会将每个缓存区填满，直至通道中无数据或缓冲区没有空间；

gather（聚集）：指的是将多个缓冲区 Buffer 聚集起来写入到通道的过程，该过程类似于将多个缓冲区的内容连接起来写入通道；

5 应用技巧

软件及对应第三方软件版本

Vivado 每个版本对应的第三方工具版本是不一致的，如果不对应可能出错！
使用不同版本 Vivado，请参考官方文档为 ug973

DCP 生成步骤

disable XDC 文件

设置生成网表程序 top 层

输入以下命令

```
synth_design -mode out_of_context -flatten_hierarchy rebuilt -top <top_module_name>
```

```
-part <part>
```

```
write_checkpoint <file_name>.dcp
```

命令解释

Command Option	Description
-mode out_of_context	Prevents I/O insertion for synthesis and downstream tools. The mode is saved in checkpoints if write_checkpoint is issued.
-flatten_hierarchy rebuilt	There are several values allowed for -flatten_hierarchy, but rebuilt is the recommended setting for HD flows.
-top	This is the module/entity name of the module being synthesized. This switch can be omitted if set_property top <top_module_name> [current_fileset] is issued prior to synth_design.
-part	This is the Xilinx part being targeted (e.g., xc7k325tffg900-3)

Figure 1

详细信息参考 ug905 page10

DCP 文件仿真

ViVado 软件直接打开 DCP 文件，或者 open_checkpoint XX.dcp

write_verilog -mode funcsim XX.v （具体参考 write_verilog 命令）

使用生成的.v 文件即可进行仿真！

EDIF 生成步骤

设置生成网表程序 top 层

在综合选项中去掉 IOBuffer，具体操作为在综合设置窗口的 Options 下面最后一项 More Options 一栏写入 -no_iobuf;

综合完成后，打开 Synthesized Design，并在 Tcl Console 中输入：write_edif path/xx.edif 例化时，要保留一个跟 eidf 同名的 hdl 文件，且文件中只保留 module 的接口和 parameter 参数

Tcl 命令：

不含 Xilinx IP

```
write_edif F:/FPGA/abc.edf
```

包含 Xilinx IP

```
write_edif -security_mode all F:/FPGA/abc.edf
```

编辑和修改官方 IP 核

在某些情况下，需要修改官方 IP 核。由于 vivado 版本不同，方法步骤也不尽相同，，具体参考软件使用版本对应文档。

本章节不具体介绍方法步骤！

具体参考文档为 UG896

Editing or Overriding IP Sources 章节

BIT 文件校验与回读

回读

使用 Vivado tcl 命令可以回读程序，可以生成 ASCII 和二进制两种格式！

ASCII (rbd)

```
readback_hw_device -readback_file a.rbd
```

二进制 (bin)

```
readback_hw_device -bin_file a.bin
```

校验

校验的前提是生成 mask file

另外：软件可以永久禁止回读，通过配置寄存器禁止 JTAG 访问

增量式编译

增量包含两部分，增量综合和增量布局布线！

过程如下：

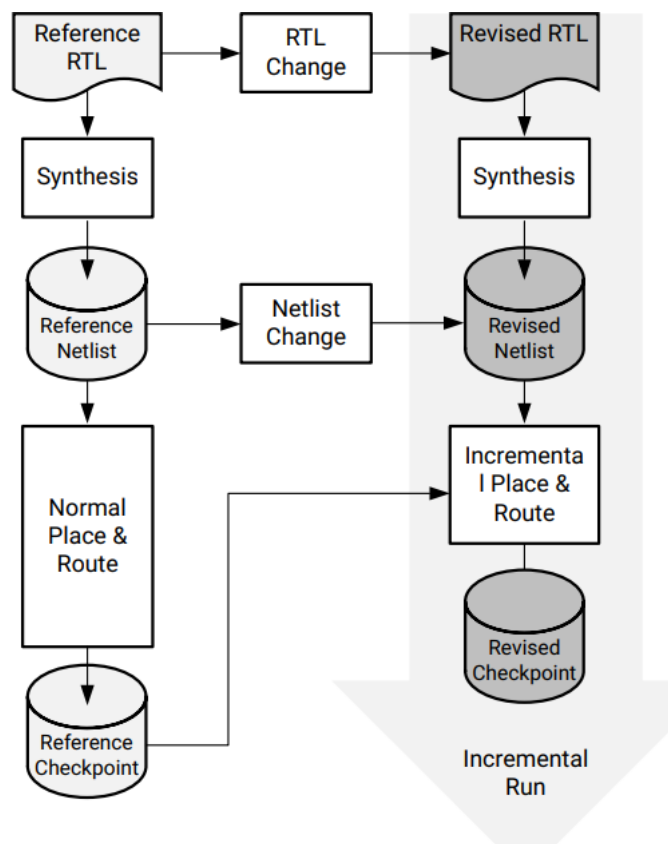


Figure 2

增量综合

在此过程中，该工具会将增量综合信息放入生成的 DCP 文件中，以便在以后的运行中引用。它将检测设计何时发生更改，然后仅对已更改的设计部分重新运行综合。此流程的主要优势在于，对于更改很小的设计，运行时间将显著减少。此外，在 RTL 中插入小的更改时，设计的 QoR 波动会更小。

Settings -> Synthesis -> Incremental Synthesis 进行设置

Incremental Synthesis: 可以选择使用已知 dcp，最后创建 dcp（默认），关闭增量综合

incremental_mode: 描述跨分区优化，可以选择 Quick, default, aggressive, off

增量布局布线:

发生在增量编译设计流程的实现阶段

通过重复使用参考设计中先前的布局和布线，保持 QoR 的可预测性。

加快布局和布线运行时间或尝试最后一英里时序收敛。

增量设计包括 RTL 更改、网表更改或两者。

允许更改约束，但一般收紧约束将显著影响布局和布线，通常最好在增量流程之外添加

Incremental Implementation 和 Incremental Synthesis 是有区别的，前者默认状态是没设置的，需要在软件上进行设置

Incremental Directives: 包含以下三种方式

RuntimeOptimized

指令尝试尽可能多地重用来自参考运行的布局和布线信息。计时目标将与参考运行相同。如果参考运行的 WNS 为 -0.050 ，那么增量运行将不会尝试关闭此设计的时序，而是以 -0.050 为目标。这仅影响设置时间。这是未指定指令时的默认行为

TimingClosure

指令将重用参考中的布局和布线，但它会破坏不符合时序的路径并尝试关闭它们。运行一些运行时间密集型算法以获得尽可能多的时序改进，但由于很大程度上放弃了布局，因此前端增益有限。这种技术对参考 WNS > -0.250 ns 的设计非常有效。

Quick

是一种特殊的模式，在布局布线过程中不调用定时器，而是以相关逻辑的布局为指导。这是最快的模式，但不适用于大多数设计

设计需要 WNS > 1.000 ns 才能有效。这些通常是 ASIC 仿真或原型设计

增量实现包含三种模式：自动，高复用，低重用！

自动

需满足以下条件

94% cell 匹配

90% net 匹配

WNS > -0.250

高复用

Cell 重复率高于 75%，运行

低复用

用户可以针对单元类型、分层单元、时钟区域和 SLR 进行重用

目标 WNS 始终为 0

增量指令被忽略，而使用默认布局布线算法中的指令

低重用模式对于在特定区域对布局和布线提出挑战的设计最为有效！

备注：增量式编译，用户一般只进行实现部分，综合部分软件默认即可！这是一个较为复杂的过程，此文仅为简单介绍！详细参考相关官方文档！另外部分内容不同软件版本有稍许不同！

Vivado Store

Vivado Store 将 tcl 应用程序，pcb 和可配置的示例设计整合到一个位置。所有的项目都显示在 GUI 中，并且可以安装，删除或升级单个项目。

TCL 应用程序：主要为与 Vivado 设计套件一起使用而设计的 Tcl 代码的开源存储库。

Boards：Xilinx 和第三方托管板文件的 GitHub 存储库。将电路板文件与 Vivado 结合使用可以通过将电路板级资源集成到设计环境中来简化设计创建。

示例设计：由 Xilinx 和第三方可配置示例设计组成的 GitHub 存储库
可以添加和扩展 tcl 命令，也可以进行修改。可以自定义部分 tcl 命令！
还有一些示例工程链接，有一定参考价值！

代码加密

Xilinx 软件 Vivado 可以对 verilog 或 VHDL 代码进行加密
加密方式采用 RSA 加密方式！

加密注意事项

- 1, 每个版本加密文件只能用于此版本，不支持其他版本混用
- 2, 加密文件为.vp 文件
- 3, Xilinx 每个软件版本均提供公版 RSA 密钥，不能混用。
- 4, Verilog 和 VHDL 加密命令不同
- 5, 可以对整个代码加密，也可以加密代码一部分。
整个代码加密无端口号，key 文件要进行部分修改。

以 Verilog 文件加密为例

a), 代码和密钥文件独立

```
encrypt -lang verilog -ext .vp -key keyfile.txt user.v
```

b), 将密钥写入代码（全加密/部分加密）

```
encrypt -lang verilog -ext .vp user.v
```

将代码文件直接添加到 BD

选中代码文件

- 1, set_property source_mgmt_mode All [current_project]
 - 2, create_bd_cell -type module -reference <module_name> <module_name_inst>
- 或者 右键 add module to block design

如若顶层内含有一般 IP，转成 xci 格式！

```
set_property generate_synth_checkpoint 0 [get_files <ip_name>.xci]
```

此方式类似一般打包 IP，但不能包含 dcp 文件。

Tcl 脚本配置 IP 方式

我们在分析了解某些官方示例工程中，，有时会出现一些不同情况。

比如，某些 IP 有些端口我在 GUI 配置时无法隐藏，或者不用，但却无法进行配置。

Tcl 脚本下载程序

Download.bat

内容如下

```
////////////////////////////////////
call C:\Xilinx\Vivado\2022.2\bin\vivado.bat -mode batch -source download.tcl

if exist *isWriteableTest*.tmp del /F *isWriteableTest*.tmp
if exist vivado_*.backup.jou del /F vivado_*.backup.jou
if exist vivado_*.backup.log del /F vivado_*.backup.log
if exist vivado_*.str del /F vivado_*.str
if exist *isWriteableTest*.tmp del /F *isWriteableTest*.tmp
pause
////////////////////////////////////

Bit_download.tcl
内容如下
////////////////////////////////////
open_hw
connect_hw_server -url localhost:3121
current_hw_target [get_hw_targets */xilinx_tcf/Digilent/*]
set_property PARAM.FREQUENCY 15000000 [get_hw_targets */xilinx_tcf/Digilent/*]
open_hw_target
set_property PROGRAM.FILE {example_top.bit} [lindex [get_hw_devices] 0]
current_hw_device [lindex [get_hw_devices] 0]
refresh_hw_device -quiet [lindex [get_hw_devices] 0]
program_hw_devices [lindex [get_hw_devices] 0]
refresh_hw_device -quiet [lindex [get_hw_devices] 0]
close_hw
```

6 官方 IP 使用说明

RAM 初始化

```
IP
memory_initialization_radix = 2;
memory_initialization_vector =
000000000000010110000000100000000
000000000000010110000001000000001
000000000000000010000110100001100
000000000000000010000111000001101;
```

FIR
Radix = 16;
CoefData= 346, EDA, 0D6, F91, F91, 0D6, EDA, 346;
注意对应 IP 参数配置

逻辑代码方式

```
reg [DATA_WIDTH-1:0] ram [DEPTH-1:0];
integer i;
initial for (i=0; i<DEPTH; i=i+1) ram[i] = 0;
end
```

```
ASCII 文本 reg [31:0] ram [0:63];
initial begin
$readmemb("rams_20c.data", ram, 0, 63);
end
```

MIG 配置

使用和配置 MIG 前,, 需要了解 DDR 芯片或者内存条的容量, 组成方式, 数据宽度等基本信息!
MIG 不同芯片选择的 IP 核些许不同! 但方式原理大体相似!

下面简要介绍 IP 配置关键参数及含义!

逻辑模式: MIG 可以选择逻辑接口或者 AXI

BD 模式: 只能 AXI 模式

Options for Controller 0 - DDR3 SDRAM

Clock Period: Choose the clock period for the desired frequency. The allowed period range(2500 - 3300) is a function of the selected FPGA part and FPGA speed grade. Refer to the User Guide for more information. ps

PHY to Controller Clock Ratio: Select the PHY to Memory Controller clock ratio. The PHY operates at the Memory Clock Period chosen above. The controller operates at either 1/4 or 1/2 of the PHY rate. The selected Memory Clock Period will limit the choices.

Vccaux_io: Vccaux_io must be set to 2.0V in the High Performance banks for the highest data rates. Vccaux_io is not available in the High Range banks. Note that Vccaux_io is common to groups of banks. Consult the 7 Series Datasheets and FPGA SelectIO Resources User Guide for more information.

Memory Type: Select the memory type. Type(s) marked with a warning symbol are not compatible with the frequency selection above.

Memory Part: Select the memory part. Part(s) marked with a warning symbol are not compatible with the frequency selection above. Find an equivalent part or create a part using the "Create Custom Part" button if the part needed is not listed here. The "Create Custom Part" feature is not supported for RLD RAM II.

Memory Voltage: Select the Voltage of the Memory part selected.

Data Width: Select the Data Width. Parts marked with a warning symbol are not compatible with the frequency and memory part selected above.

ECC: MIG supports ECC for 72 bit data width configuration. To be able to select ECC, select a data width that has ECC supported.

Data Mask: Enable or disable the generation of Data Mask (DM) pins using this check box. This option can be selectable only if the memory part selected has DM pins. Uncheck this box to not use data masks and save FPGA I/Os that are used for DM signals. ECC designs (DDR3 SDRAM, DDR2 SDRAM) will not use Data Mask. ☒

Number of Bank Machines: This parameter defines the number of bank machines. A given bank machine manages a single DRAM bank at any given time.
Note: Setting a lower value will result in lower resource utilization, but may effect controller efficiency for certain traffic patterns.

ORDERING: Normal mode allows the memory controller to reorder commands to the memory to obtain the highest possible efficiency. Strict mode forces the controller to execute commands in the exact order received.

Clock Period:

控制器的工作频率，该频率和 FPGA 芯片以及 DDR 芯片有关（DDR 工作频率）

PHY to Controller Clock Ratio:（一般可选 2:1 或 4:1）

以 AXI 为例，即 DDR3 频率与外部逻辑频率之比。

假设 DDR 频率为 400MHZ，选择 2:1，外部逻辑频率为 200MHZ，该数据影响后续配置的位置

Memory Type

Components: 一般 DDR 芯片选择此项

RDIMM: 带寄存器的双线内存模块（用于工作站服务器内存条）

UDIMM: 无缓冲双信道内存模块（一般台式机内存条）

SODIMM: 小型双列直插式内存模块（笔记本内存条）

Memory Part :

若使用的 DDR 选项里没有，可以选择参数类似的，，或者可以先点击 Creat Custom Part 进行创新器件

（配置时需熟悉 DDR 芯片参数概念，诸如，row/col/bank/等）

Memory Voltage:

DDR 电压,, 参考使用 DDR

Data Width:

DDR 数据宽度,, 如果单片 16 位,, 两片并联则为 32 位

Number of Bank Machines:

和设备大小有关和资源利用有关, 默认 4 (无休修改)

Axi Parameter Options C0 - DDR3 SDRAM

Data Width

AXI DATA WIDTH: Data width of AXI read & write channels. The data width is less than or equal to user interface data width with the possible values 32, 64, 128, 256 & 512.

32

Arbitration Scheme

Select the arbitration scheme between the read and write address channels

RD_PRI_REG

Narrow Burst Support

Enables logic to support narrow bursts on the AXI4 slave interface. Can be set to zero if no masters in the system issue narrow bursts and all the data widths are equal. (1-Enable, 0-Disable)

0

Address Width

AXI4 address width of read and write address channels.

27

ID Width

AXI4 ID width for read and write channels. AXI4 ID is used as the identification tag for write or read address group of signals

4

(参考 page164)

Date Width: AXI 数据宽度 (AXI 时钟可以由 ui_clk)

Table 1-62: AXI Byte Address Masking

PHY to MC Clock Ratio	UI Data Width	Memory Interface Data Width	AXI Byte Address [7:0] (LSBs) Masking
4:1	64	8	A[7:0]
	128	16	A[7:1], 1'b0
	256	32	A[7:2], 2'b00
	512	64	A[7:3], 3'b000

Table 1-62: AXI Byte Address Masking (Cont'd)

PHY to MC Clock Ratio	UI Data Width	Memory Interface Data Width	AXI Byte Address [7:0] (LSBs) Masking
2:1	32	8	A[7:0]
	64	16	A[7:1], 1'b0
	128	32	A[7:2], 2'b00
	256	64	A[7:3], 3'b000

Arbitration Scheme: (仲裁方案)

在读写地址之间选择仲裁方案

Narrow Burst Suppot: (窄突发支持)

1-Enable , 0-Disable

Memory Options C0 - DDR3 SDRAM

Input Clock Period:

Select the period for the PLL input clock (CLKIN). MIG determines the allowable input clock periods based on the Memory Clock Period entered above and the clocking guidelines listed in the User Guide. The generated design will use the selected Input Clock and Memory Clock Periods to generate the required PLL parameters. If the required input clock period is not available, the Memory Clock Period must be modified.

5000 ps (200 MHz)

☐ Select Additional Clocks (if required)

MIG can generate up to 5 additional clocks to be used in Fabric logic. This will be generated from the same MMCM which is used for generation of UI_CLK. The first clock(Clock 0) has a wider range of choices. All the values in the additional clocks drop downs are calculated considering the MMCM VCO frequency as 1250 ps (800 MHz) Mhz. For complete details on clocking of MIG, refer to MIG User Guide.

Clock 0

NONE

D = 1

Clock 1

NONE

D = 1

Clock 2

NONE

D = 1

Clock 3

NONE

D = 1

Clock 4

NONE

D = 1

Choose the Memory Options for the memory device. Memory Option selections are restricted to those supported by the controller. Consult the memory vendor data sheet for more information.

Read Burst Type and Length

The burst type determines the data ordering within a burst. Consult the memory datasheet for more information. Burst length 8 is the only supported value.

Sequential

Output Driver Impedance Control

Programmable impedance for the output buffer.

RZQ/7

RTT (nominal) - On Die Termination (ODT)

Select the nominal value of ODT for the DQ, DQS/DQS# and DM signals on the component or DIMM interface. This must be set to RZQ/6 ((40 ohms) for data rates at 1333 Mbps and above. In 2 slot DIMM configurations this value will be used for the unwritten slot during a write and will also be used for the unselected slot during a read. Use board level simulation to choose the optimum value.

RZQ/4

Controller Chip Select Pin

The Chip Select (CS#) pin can be tied low externally to save one pin in the address/command group when this selection is set to 'Disable'. Disable is only valid for single rank configurations.

Enable

Memory Address Mapping Selection

User Address

A

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

BANK

ROW

COLUMN

☐

BANK

ROW

COLUMN

☒

BANK

ROW

COLUMN

input clock period:

与连接晶振输出时钟保持一致

Read Burst Type and Length:

默认

Output Driver Impedance Control: (输出阻抗, RZQ 是 240 欧, RZQ/6 就是 40 欧)

查看 DDR

RTT

System Clock
Choose the desired input clock configuration. Design clock can be Differential or Single-Ended.

System Clock

Differential

Reference Clock
Choose the desired reference clock configuration. Reference clock can be Differential or Single-Ended.

Reference Clock

Differential

System Reset Polarity
Choose the desired System Reset Polarity.

System Reset Polarity

ACTIVE LOW

Debug Signals Control
This feature allows various debug signals present in the IP to be monitored on the ChipScope tool. The debug signals include status signals of various PHY calibration stages. Enabling this feature will connect all the debug signals to the ChipScope ILA and VIO cores in the example design top module. A part of each bus in the debug interface has been grounded so that users can replace the grounded signals with the required signals.

Debug Signals for Memory Controller

OFF

Sample Data Depth
This selects the value of Sample Data depth for Chipscope ILA used in Debug logic.

Sample Data Depth

1024

Internal Vref
Internal Vref can be used to allow the use of the Vref pins as normal IO pins. This option can only be used at 800 Mbps and lower data rates. This can free 2 pins per bank where inputs are used. This setting has no effect on banks with only outputs.

Internal Vref

☐

IO Power Reduction
Significantly reduces average IO power by automatically disabling DQ/DQS IBUFs and internal terminations during WRITES and periods of inactivity

IO Power Reduction

ON

XADC Instantiation
The memory interface uses the temperature reading from the XADC block to perform temperature compensation and keep the read DQS centered in the data window. There is one XADC block per device. If the XADC is not currently used anywhere in the design, enable this option to have the block instantiated. If the XADC is already used, disable this MIG option. The user is then required to provide the temperature value to the top level 12-bit device_temp_i input port. Refer to Answer Record 51687 or the UG586 for detailed information.

XADC Instantiation

Enabled

System Clock

Differential / Single-Ended 一般连接输入时钟

No Buffer 一般使用 PLL 输出时钟

Reference Clock

当输入频率在 199 和 201 MHz 之间 (即输入时钟周期在 5,025ps (199 MHz) 和 4,975 ps (201 MHz) 之间) 时, 会出现 Use System Clock 选项。参考时钟频率基于数据速率请注意, 添加 MMCM 以创建适当的 REF_CLK 频率以上 1,333 MB / s。当没有选择“无缓冲器”选项时, 在 RTL 代码中不实例化

IBUF 原语，不为参考时钟分配引脚

如果从 MIG 为 No Buffer 选项生成的设计在未执行更改的情况下实施，则设计可能会由于未针对 ref_clk_i 信号实例化的 IBUF 而导致实施失败。因此对于 No Buffer 场景，需要将 ref_clk_i 信号连接到内部时钟

System Reset Polarity

选择高低电平复位（默认低电平复位）

7 个人 IP 使用说明

8 硬件接口设计