FPGA 开源工作室将通过五篇文章来给大家讲解 xilinx FPGA 使用 mig IP 对 DDR3 的读写控制，旨在让大家更快的学习和应用 DDR3。
本实验和工程基于 Digilent 的 Arty Artix-35T FPGA 开发板完成。
软件使用 Vivado 2018.1

第二篇：mig IP 的创建

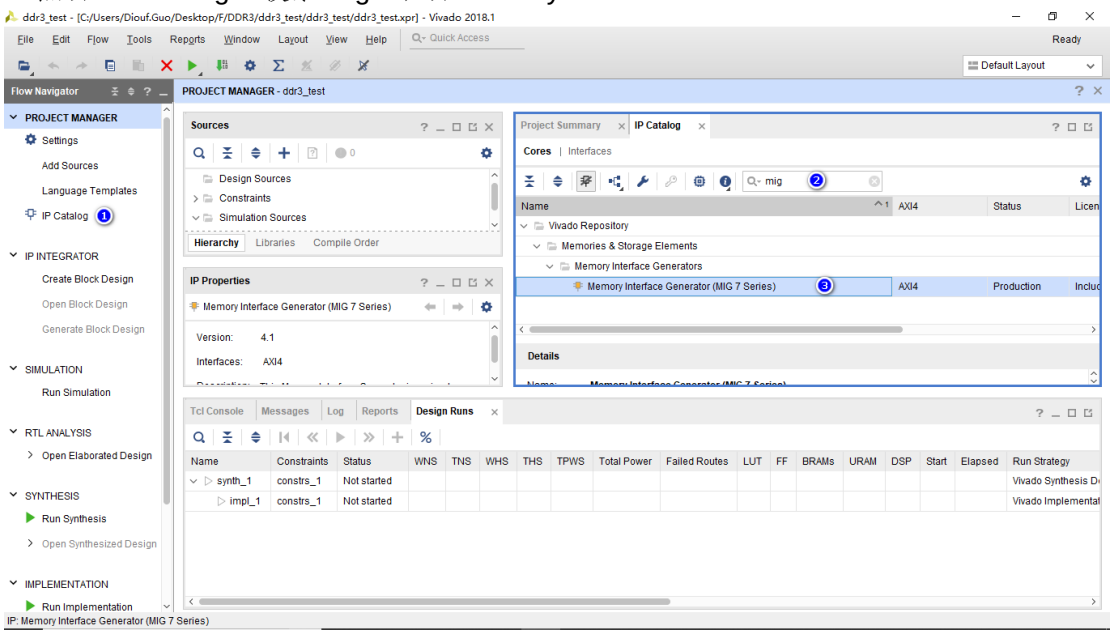
1 DDR3

Digilent 的 Arty Artix-35T FPGA 开发板板载 MT41K128M16JT-125 DDR3 基本信息如下表所示。

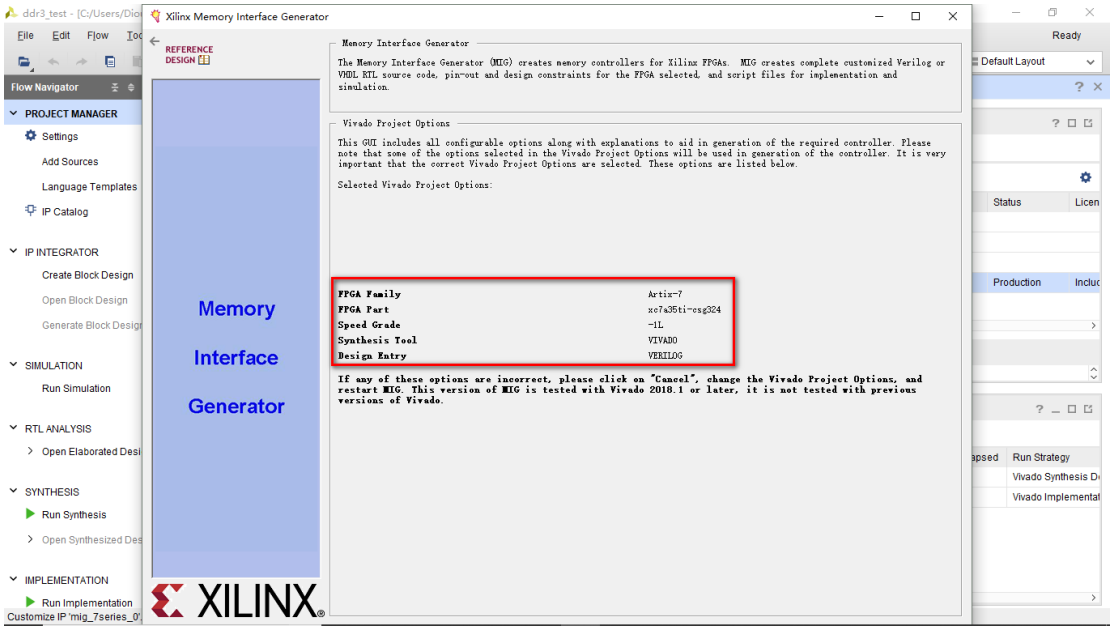
Setting	Value
Memory type	DDR3 SDRAM
Max clock period	3000ps (667Mbps data rate)
Memory part	MT41K128M16XX-15E
Memory voltage	1.35V
Data width	16
Data mask	Enabled
Recommended input clock period	6000ps (166.667 MHz)
Output driver impedance control	RZQ/6
Controller chip select pin	Enabled
Rtt (nominal) – on-die termination	RZQ/6
Interval Vref	Enabled
Internal termination impedance	50omhs

2 mig IP 的创建

1>点击 IP Catalog ->搜索 mig->双击 Memory Interface Generator（MIG 7 Series）

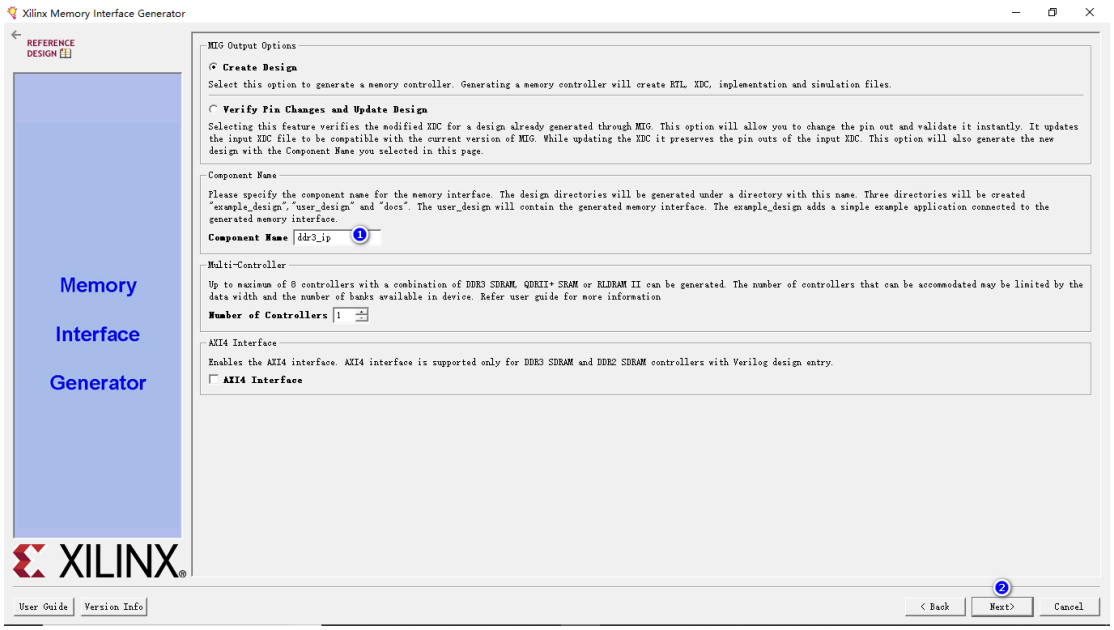


2>打开后可以看到一些基本信息

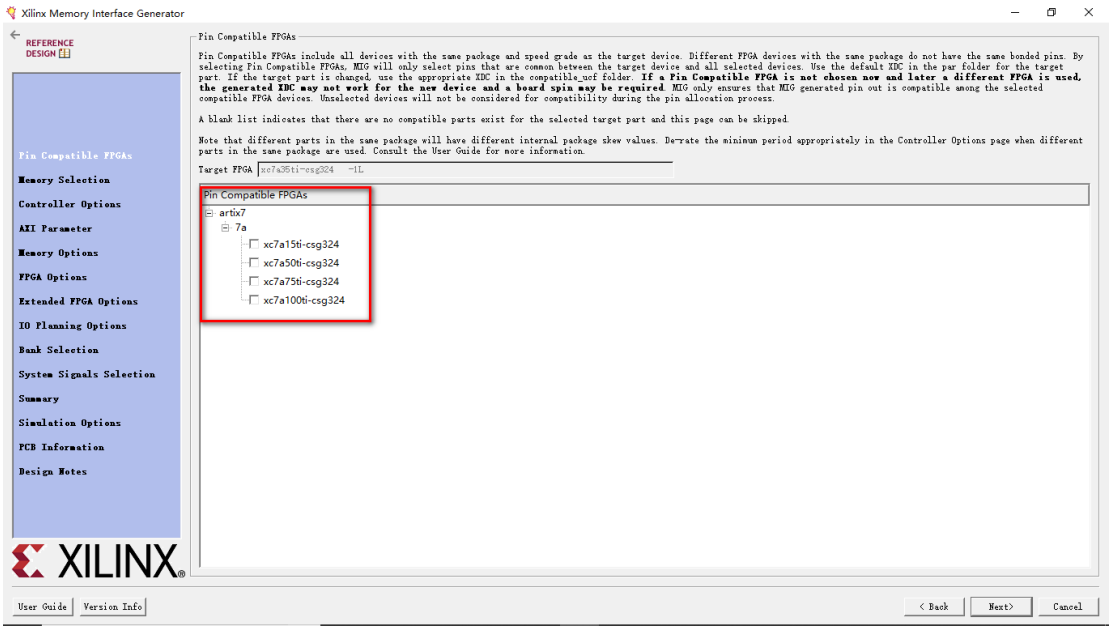


3> Enter a component name in the Component Name field ->Next

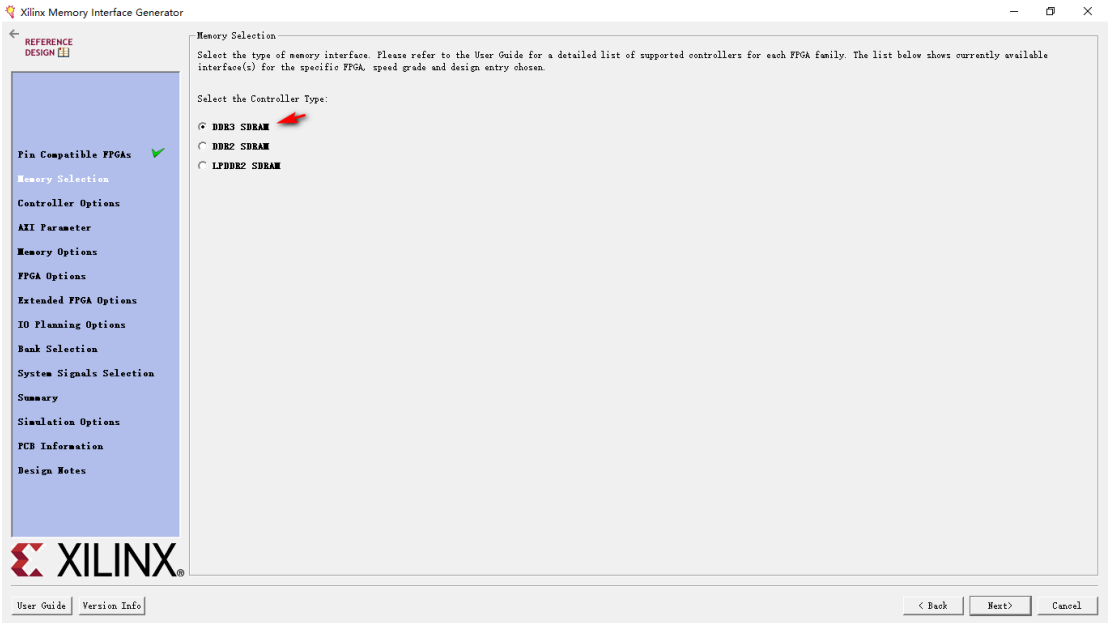
Component name(组件名称)：ddr3_ip



4>这里我们不做兼容性选择，直接下一步



5>控制类型选择 DDR3 SDRAM



6>①Clock Period:(此功能表示所有控制器的工作频率,频率模块受所选 FPGA 和器件速度等级等因素的限制。) 3000ps(333.33MHZ)。

②PHY to Controller Clock Ratio :(此功能确定物理层（存储器）时钟频率与控制器和用户界面时钟频率的比率。 由于 FPGA 逻辑时序限制，2：1 比率会降低最大存储器接口频率。2：1 比率的用户界面数据总线宽度是物理存储器接口宽度宽度的四倍，而 4：1 比率的总线宽度是物理存储器接口宽度的八倍。 2：1 比率具有较低的延迟。 4：1 的比率是最高数据速率所必需的)4:1。

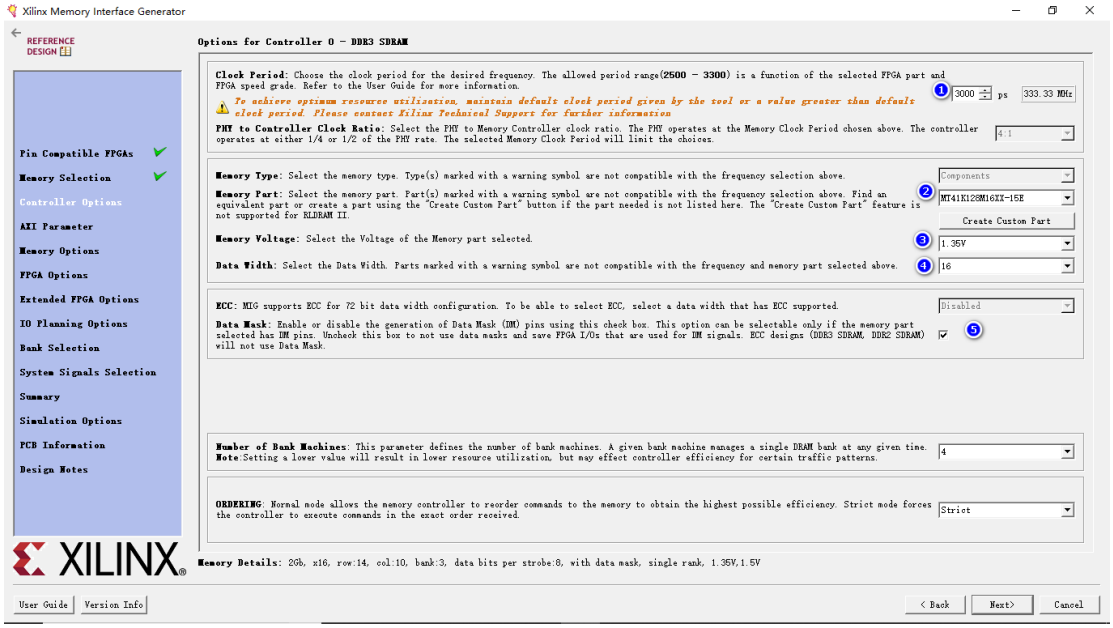
③Memory Type：此功能选择设计中使用的内存部件类型。

④Memory Part ：此选项为设计选择内存部件。 选择可以从列表中创建或者可以创建新部件。MT41K128M16XX-15E。

⑤Memory Voltage:根据设计原理图 1.35V。

⑥Data Width：（可以根据之前选择的存储器类型在此处选择数据宽度值。 该列表显示所选部件的所有支持的数据宽度。 可以选择其中一个数据宽度。 这些值通常是各个器件数据宽度的倍数。 在某些情况下，宽度可能不是精确倍数。 例如，16 位是 x16 组件的默认数据宽度，但 8 位也是有效值。）16。

⑦Data Mask：（选择时，此选项会分配数据屏蔽引脚。 应取消选择此选项以释放数据屏蔽引脚并提高引脚效率。 此外，对于不支持数据掩码的内存部分禁用此功能。）勾选。NXET。

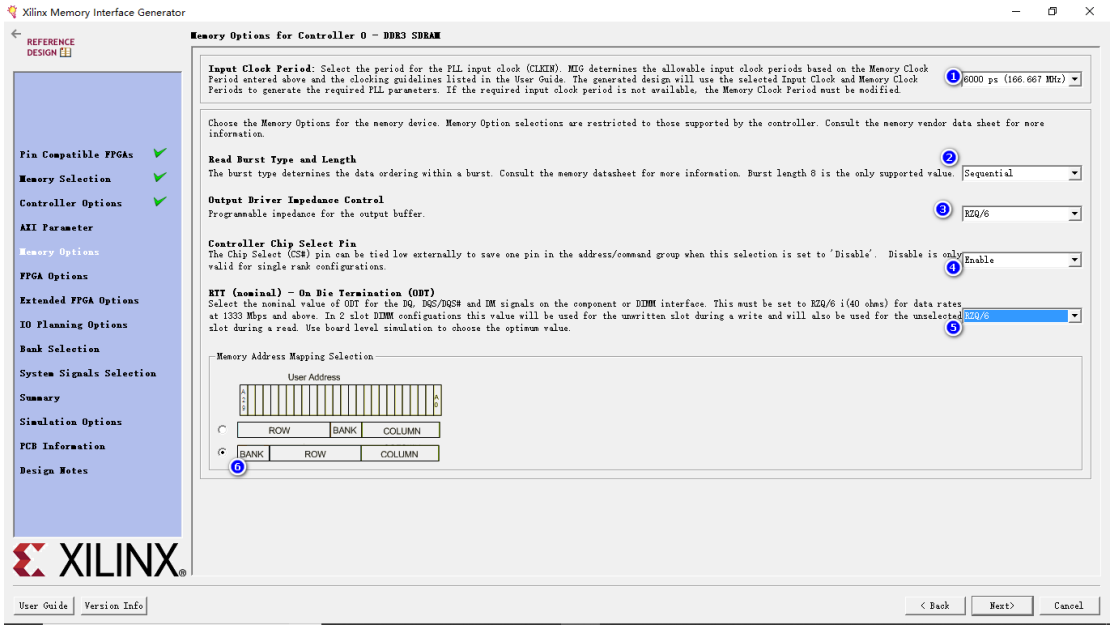


7>①Input clock Period:6000ps(166.667MHZ)。

②Read Burst Type and Length:Sequential。

③Output Driver Impedance Control:RZQ/6。

其他默认，NEXT。



8>①System Clock :(此选项为 sys_clk 信号对选择时钟类型（单端，差分或无缓冲）。 选择 No Buffer 选项时，IBUF 原语不会在 RTL 代码中实例化，并且不会为系统时钟分配引脚。)No Buffer。

②Reference Clock :（此选项为 clk_ref 信号对选择时钟类型（单端，差分，无缓冲或使用系统时钟）。 当输入频率介于 199 和 201 MHz 之间时（即输入时钟周期介于 5,025 ps（199 MHz）和 4,975 ps（201 MHz）之间），将显示 Use System Clock（使用系统时钟）选项。参考时钟频率基于数据速率 并注意添加 MMCM 以创建高于 1,333 Mb / s 的适当 ref_clk 频率。当选择 No Buffer 选项时，IBUF 原语不会在 RTL 代码中实例化，并且引脚不会分配给参考时钟。）No Buffer。

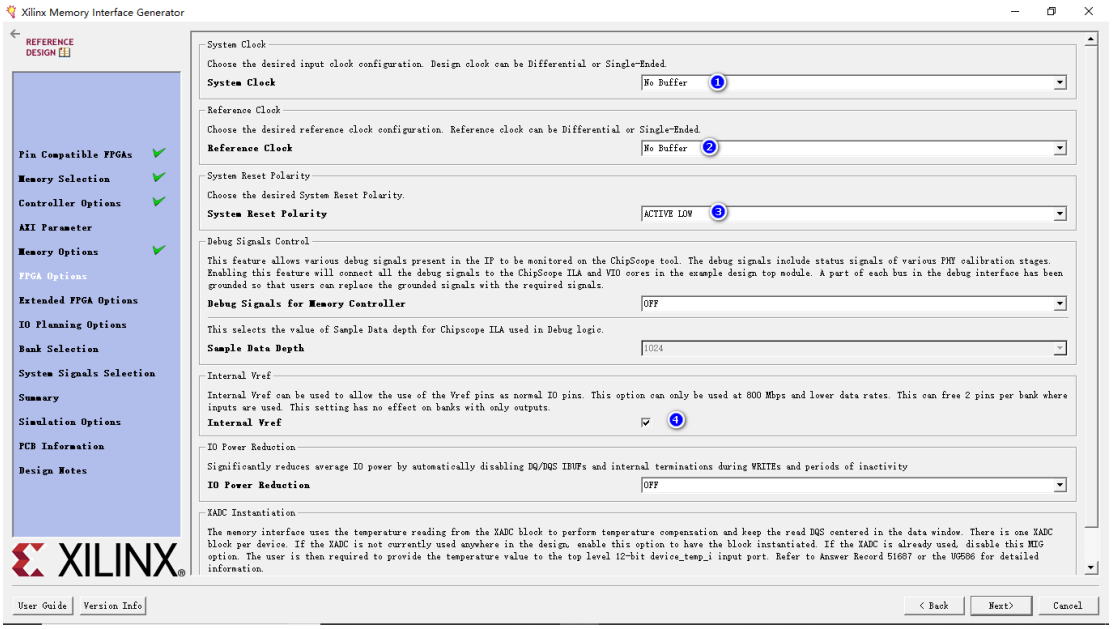
③System Reset Polarity:（可以选择系统复位（sys_rst）的极性。 如果选项选择为低电平有效，则参数 RST_ACT_LOW 设置为 1，如果设置为高电平 - 高，则参数 RST_ACT_LOW 设置为 0。）ACTIVE LOW。

④Debug Signals Control:选择此选项可以将校准状态和用户端口信号端口映射到 example_top 模块中的 ILA 和 VIO。这有助于使用 Vivado Design Suite 调试功能监控用户界面端口上的流量。 取消选择 Debug Signals Control 选项会使 example_top 模块中的调试信号保持未连接状态，并且 IP 目录不会生成 ILA / VIO 模块。 此外，始终禁用调试端口以进行功能仿真。OFF。

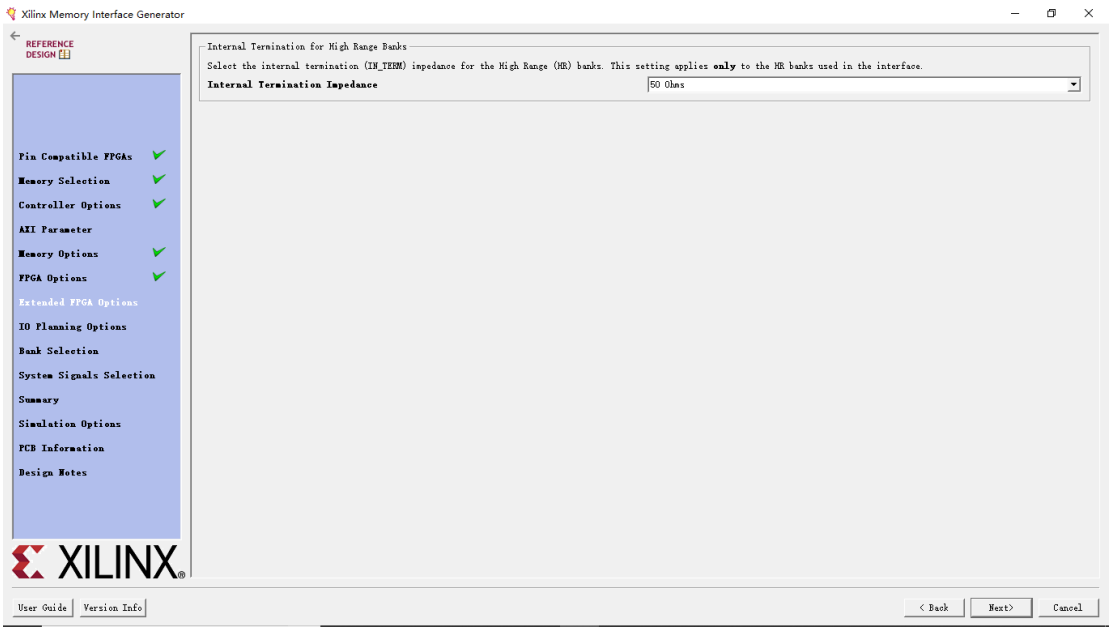
⑤Sample Data Depth：此选项选择 Vivado 调试逻辑中使用的 ILA 模块的样本数据深度。当“内存控制器的调试信号”选项为“开”时，可以选择此选项。

⑥Internal Verf:(内部 VREF 可用于数据组字节，以允许使用 VREF 引脚进行正常的 I / O 使用。 内部 VREF 仅应用于 800 Mb / s 或更低的数据速率。)勾选。

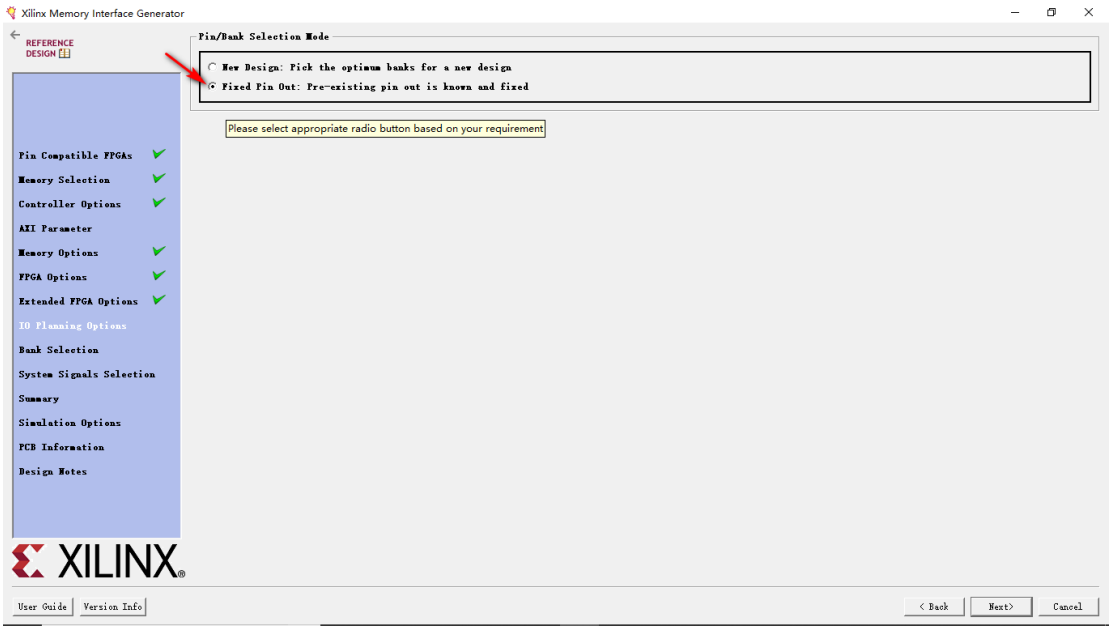
其他默认，NEXT。



9>默认，NEXT。



10>选择 Fixed Pin Out。我们的原理图管脚已经确定无需从新设计。



10>点击 Read XDC/UCF，这里 DDR3 管脚支持两种约束文件。

UCF:

```
1.  NET "ddr3_dq[0]"        LOC = "K5"      |   IOSTANDARD = SSTL135        ;
2.  NET "ddr3_dq[1]"        LOC = "L3"      |   IOSTANDARD = SSTL135        ;
3.  NET "ddr3_dq[2]"        LOC = "K3"      |   IOSTANDARD = SSTL135        ;
4.  NET "ddr3_dq[3]"        LOC = "L6"      |   IOSTANDARD = SSTL135        ;
5.  NET "ddr3_dq[4]"        LOC = "M3"      |   IOSTANDARD = SSTL135        ;
6.  NET "ddr3_dq[5]"        LOC = "M1"      |   IOSTANDARD = SSTL135        ;
7.  NET "ddr3_dq[6]"        LOC = "L4"      |   IOSTANDARD = SSTL135        ;
8.  NET "ddr3_dq[7]"        LOC = "M2"      |   IOSTANDARD = SSTL135        ;
9.  NET "ddr3_dq[8]"        LOC = "V4"      |   IOSTANDARD = SSTL135        ;
10. NET "ddr3_dq[9]"        LOC = "T5"      |   IOSTANDARD = SSTL135        ;
11. NET "ddr3_dq[10]"       LOC = "U4"      |   IOSTANDARD = SSTL135        ;
12. NET "ddr3_dq[11]"       LOC = "V5"      |   IOSTANDARD = SSTL135        ;
13. NET "ddr3_dq[12]"       LOC = "V1"      |   IOSTANDARD = SSTL135        ;
```

```
14. NET "ddr3_dq[13]"      LOC = "T3"      |   IOSTANDARD = SSTL135          ;
15. NET "ddr3_dq[14]"      LOC = "U3"      |   IOSTANDARD = SSTL135          ;
16. NET "ddr3_dq[15]"      LOC = "R3"      |   IOSTANDARD = SSTL135          ;
17. NET "ddr3_dm[0]"       LOC = "L1"      |   IOSTANDARD = SSTL135          ;
18. NET "ddr3_dm[1]"       LOC = "U1"      |   IOSTANDARD = SSTL135          ;
19. NET "ddr3_dqs_p[0]"    LOC = "N2"      |   IOSTANDARD = DIFF_SSTL135     ;
20. NET "ddr3_dqs_n[0]"    LOC = "N1"      |   IOSTANDARD = DIFF_SSTL135     ;
21. NET "ddr3_dqs_p[1]"    LOC = "U2"      |   IOSTANDARD = DIFF_SSTL135     ;
22. NET "ddr3_dqs_n[1]"    LOC = "V2"      |   IOSTANDARD = DIFF_SSTL135     ;
23. NET "ddr3_addr[13]"    LOC = "T8"      |   IOSTANDARD = SSTL135          ;
24. NET "ddr3_addr[12]"    LOC = "T6"      |   IOSTANDARD = SSTL135          ;
25. NET "ddr3_addr[11]"    LOC = "U6"      |   IOSTANDARD = SSTL135          ;
26. NET "ddr3_addr[10]"    LOC = "R6"      |   IOSTANDARD = SSTL135          ;
27. NET "ddr3_addr[9]"     LOC = "V7"      |   IOSTANDARD = SSTL135          ;
28. NET "ddr3_addr[8]"     LOC = "R8"      |   IOSTANDARD = SSTL135          ;
29. NET "ddr3_addr[7]"     LOC = "U7"      |   IOSTANDARD = SSTL135          ;
30. NET "ddr3_addr[6]"     LOC = "V6"      |   IOSTANDARD = SSTL135          ;
31. NET "ddr3_addr[5]"     LOC = "R7"      |   IOSTANDARD = SSTL135          ;
32. NET "ddr3_addr[4]"     LOC = "N6"      |   IOSTANDARD = SSTL135          ;
33. NET "ddr3_addr[3]"     LOC = "T1"      |   IOSTANDARD = SSTL135          ;
34. NET "ddr3_addr[2]"     LOC = "N4"      |   IOSTANDARD = SSTL135          ;
35. NET "ddr3_addr[1]"     LOC = "M6"      |   IOSTANDARD = SSTL135          ;
```

```
36. NET "ddr3_addr[0]"     LOC = "R2"      |   IOSTANDARD = SSTL135          ;
37. NET "ddr3_ba[2]"       LOC = "P2"      |   IOSTANDARD = SSTL135          ;
38. NET "ddr3_ba[1]"       LOC = "P4"      |   IOSTANDARD = SSTL135          ;
39. NET "ddr3_ba[0]"       LOC = "R1"      |   IOSTANDARD = SSTL135          ;
40. NET "ddr3_ck_p[0]"     LOC = "U9"      |   IOSTANDARD = DIFF_SSTL135     ;
41. NET "ddr3_ck_n[0]"     LOC = "V9"      |   IOSTANDARD = DIFF_SSTL135     ;
42. NET "ddr3_ras_n"       LOC = "P3"      |   IOSTANDARD = SSTL135          ;
43. NET "ddr3_cas_n"       LOC = "M4"      |   IOSTANDARD = SSTL135          ;
44. NET "ddr3_we_n"        LOC = "P5"      |   IOSTANDARD = SSTL135          ;
45. NET "ddr3_reset_n"     LOC = "K6"      |   IOSTANDARD = SSTL135          ;
46. NET "ddr3_cke[0]"      LOC = "N5"      |   IOSTANDARD = SSTL135          ;
47. NET "ddr3_odt[0]"      LOC = "R5"      |   IOSTANDARD = SSTL135          ;
48. NET "ddr3_cs_n[0]"     LOC = "U8"      |   IOSTANDARD = SSTL135          ;
```

XDC:

```
1. ##############################################################################
   ######################
2. ##
3. ##  Xilinx, Inc. 2010            www.xilinx.com
4. ##  周五 一月  25 13:58:27 2019
5. ##  Generated by MIG Version 2.4
6. ##
7. ##############################################################################
   ######################
8. ##  File name :       ddr3_ip.xdc
9. ##  Details :     Constraints file
10. ##                    FPGA Family:       ARTIX7
11. ##                    FPGA Part:         XC7A35TI-CSG324
12. ##                    Speedgrade:        -1L
13. ##                    Design Entry:      VERILOG
14. ##                    Frequency:         333.333 MHz
```

```
##                    Time Period:       3000 ps
##############################################################################
######################

##############################################################################
######################
## Controller 0
## Memory Device: DDR3_SDRAM->Components->MT41K128M16XX-15E
## Data Width: 16
## Time Period: 3000
## Data Mask: 1
##############################################################################
######################

#create_clock -period 6 [get_ports sys_clk_i]

#create_clock -period 5 [get_ports clk_ref_i]

############## NET - IOSTANDARD #################


# PadFunction: IO_L5P_T0_34
set_property SLEW FAST [get_ports {ddr3_dq[0]}]
set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[0]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[0]}]
set_property PACKAGE_PIN K5 [get_ports {ddr3_dq[0]}]

# PadFunction: IO_L2N_T0_34
set_property SLEW FAST [get_ports {ddr3_dq[1]}]
set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[1]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[1]}]
set_property PACKAGE_PIN L3 [get_ports {ddr3_dq[1]}]

# PadFunction: IO_L2P_T0_34
set_property SLEW FAST [get_ports {ddr3_dq[2]}]
set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[2]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[2]}]
set_property PACKAGE_PIN K3 [get_ports {ddr3_dq[2]}]

# PadFunction: IO_L6P_T0_34
set_property SLEW FAST [get_ports {ddr3_dq[3]}]
set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[3]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[3]}]
set_property PACKAGE_PIN L6 [get_ports {ddr3_dq[3]}]
```

```
56.
57. # PadFunction: IO_L4P_T0_34
58. set_property SLEW FAST [get_ports {ddr3_dq[4]}]
59. set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[4]}]
60. set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[4]}]
61. set_property PACKAGE_PIN M3 [get_ports {ddr3_dq[4]}]
62.
63. # PadFunction: IO_L1N_T0_34
64. set_property SLEW FAST [get_ports {ddr3_dq[5]}]
65. set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[5]}]
66. set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[5]}]
67. set_property PACKAGE_PIN M1 [get_ports {ddr3_dq[5]}]
68.
69. # PadFunction: IO_L5N_T0_34
70. set_property SLEW FAST [get_ports {ddr3_dq[6]}]
71. set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[6]}]
72. set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[6]}]
73. set_property PACKAGE_PIN L4 [get_ports {ddr3_dq[6]}]
74.
75. # PadFunction: IO_L4N_T0_34
76. set_property SLEW FAST [get_ports {ddr3_dq[7]}]
77. set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[7]}]
78. set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[7]}]
79. set_property PACKAGE_PIN M2 [get_ports {ddr3_dq[7]}]
80.
81. # PadFunction: IO_L10N_T1_34
82. set_property SLEW FAST [get_ports {ddr3_dq[8]}]
83. set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[8]}]
84. set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[8]}]
85. set_property PACKAGE_PIN V4 [get_ports {ddr3_dq[8]}]
86.
87. # PadFunction: IO_L12P_T1_MRCC_34
88. set_property SLEW FAST [get_ports {ddr3_dq[9]}]
89. set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[9]}]
90. set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[9]}]
91. set_property PACKAGE_PIN T5 [get_ports {ddr3_dq[9]}]
92.
93. # PadFunction: IO_L8P_T1_34
94. set_property SLEW FAST [get_ports {ddr3_dq[10]}]
95. set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[10]}]
96. set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[10]}]
97. set_property PACKAGE_PIN U4 [get_ports {ddr3_dq[10]}]
98.
99. # PadFunction: IO_L10P_T1_34
```

```
set_property SLEW FAST [get_ports {ddr3_dq[11]}]
set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[11]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[11]}]
set_property PACKAGE_PIN V5 [get_ports {ddr3_dq[11]}]

# PadFunction: IO_L7N_T1_34
set_property SLEW FAST [get_ports {ddr3_dq[12]}]
set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[12]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[12]}]
set_property PACKAGE_PIN V1 [get_ports {ddr3_dq[12]}]

# PadFunction: IO_L11N_T1_SRCC_34
set_property SLEW FAST [get_ports {ddr3_dq[13]}]
set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[13]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[13]}]
set_property PACKAGE_PIN T3 [get_ports {ddr3_dq[13]}]

# PadFunction: IO_L8N_T1_34
set_property SLEW FAST [get_ports {ddr3_dq[14]}]
set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[14]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[14]}]
set_property PACKAGE_PIN U3 [get_ports {ddr3_dq[14]}]

# PadFunction: IO_L11P_T1_SRCC_34
set_property SLEW FAST [get_ports {ddr3_dq[15]}]
set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dq[15]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_dq[15]}]
set_property PACKAGE_PIN R3 [get_ports {ddr3_dq[15]}]

# PadFunction: IO_L24N_T3_34
set_property SLEW FAST [get_ports {ddr3_addr[13]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[13]}]
set_property PACKAGE_PIN T8 [get_ports {ddr3_addr[13]}]

# PadFunction: IO_L23N_T3_34
set_property SLEW FAST [get_ports {ddr3_addr[12]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[12]}]
set_property PACKAGE_PIN T6 [get_ports {ddr3_addr[12]}]

# PadFunction: IO_L22N_T3_34
set_property SLEW FAST [get_ports {ddr3_addr[11]}]
set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[11]}]
set_property PACKAGE_PIN U6 [get_ports {ddr3_addr[11]}]

```

```
144. # PadFunction: IO_L19P_T3_34
145. set_property SLEW FAST [get_ports {ddr3_addr[10]}]
146. set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[10]}]
147. set_property PACKAGE_PIN R6 [get_ports {ddr3_addr[10]}]
148.
149. # PadFunction: IO_L20P_T3_34
150. set_property SLEW FAST [get_ports {ddr3_addr[9]}]
151. set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[9]}]
152. set_property PACKAGE_PIN V7 [get_ports {ddr3_addr[9]}]
153.
154. # PadFunction: IO_L24P_T3_34
155. set_property SLEW FAST [get_ports {ddr3_addr[8]}]
156. set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[8]}]
157. set_property PACKAGE_PIN R8 [get_ports {ddr3_addr[8]}]
158.
159. # PadFunction: IO_L22P_T3_34
160. set_property SLEW FAST [get_ports {ddr3_addr[7]}]
161. set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[7]}]
162. set_property PACKAGE_PIN U7 [get_ports {ddr3_addr[7]}]
163.
164. # PadFunction: IO_L20N_T3_34
165. set_property SLEW FAST [get_ports {ddr3_addr[6]}]
166. set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[6]}]
167. set_property PACKAGE_PIN V6 [get_ports {ddr3_addr[6]}]
168.
169. # PadFunction: IO_L23P_T3_34
170. set_property SLEW FAST [get_ports {ddr3_addr[5]}]
171. set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[5]}]
172. set_property PACKAGE_PIN R7 [get_ports {ddr3_addr[5]}]
173.
174. # PadFunction: IO_L18N_T2_34
175. set_property SLEW FAST [get_ports {ddr3_addr[4]}]
176. set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[4]}]
177. set_property PACKAGE_PIN N6 [get_ports {ddr3_addr[4]}]
178.
179. # PadFunction: IO_L17N_T2_34
180. set_property SLEW FAST [get_ports {ddr3_addr[3]}]
181. set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[3]}]
182. set_property PACKAGE_PIN T1 [get_ports {ddr3_addr[3]}]
183.
184. # PadFunction: IO_L16N_T2_34
185. set_property SLEW FAST [get_ports {ddr3_addr[2]}]
186. set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[2]}]
187. set_property PACKAGE_PIN N4 [get_ports {ddr3_addr[2]}]
```

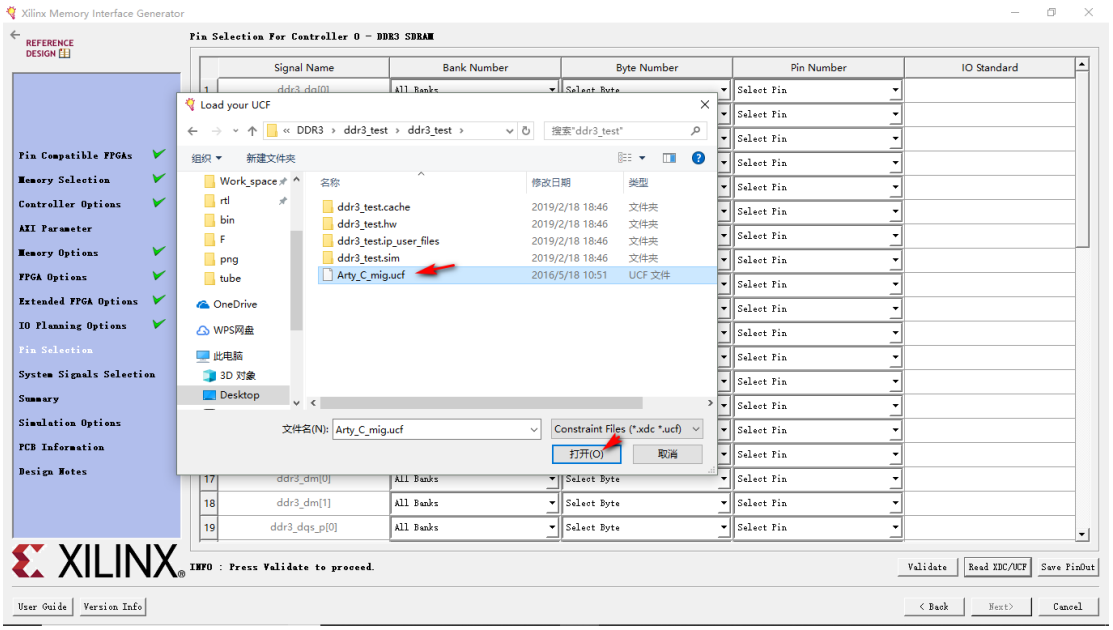
```
188.
189. # PadFunction: IO_L18P_T2_34
190. set_property SLEW FAST [get_ports {ddr3_addr[1]}]
191. set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[1]}]
192. set_property PACKAGE_PIN M6 [get_ports {ddr3_addr[1]}]
193.
194. # PadFunction: IO_L15N_T2_DQS_34
195. set_property SLEW FAST [get_ports {ddr3_addr[0]}]
196. set_property IOSTANDARD SSTL135 [get_ports {ddr3_addr[0]}]
197. set_property PACKAGE_PIN R2 [get_ports {ddr3_addr[0]}]
198.
199. # PadFunction: IO_L15P_T2_DQS_34
200. set_property SLEW FAST [get_ports {ddr3_ba[2]}]
201. set_property IOSTANDARD SSTL135 [get_ports {ddr3_ba[2]}]
202. set_property PACKAGE_PIN P2 [get_ports {ddr3_ba[2]}]
203.
204. # PadFunction: IO_L14P_T2_SRCC_34
205. set_property SLEW FAST [get_ports {ddr3_ba[1]}]
206. set_property IOSTANDARD SSTL135 [get_ports {ddr3_ba[1]}]
207. set_property PACKAGE_PIN P4 [get_ports {ddr3_ba[1]}]
208.
209. # PadFunction: IO_L17P_T2_34
210. set_property SLEW FAST [get_ports {ddr3_ba[0]}]
211. set_property IOSTANDARD SSTL135 [get_ports {ddr3_ba[0]}]
212. set_property PACKAGE_PIN R1 [get_ports {ddr3_ba[0]}]
213.
214. # PadFunction: IO_L14N_T2_SRCC_34
215. set_property SLEW FAST [get_ports {ddr3_ras_n}]
216. set_property IOSTANDARD SSTL135 [get_ports {ddr3_ras_n}]
217. set_property PACKAGE_PIN P3 [get_ports {ddr3_ras_n}]
218.
219. # PadFunction: IO_L16P_T2_34
220. set_property SLEW FAST [get_ports {ddr3_cas_n}]
221. set_property IOSTANDARD SSTL135 [get_ports {ddr3_cas_n}]
222. set_property PACKAGE_PIN M4 [get_ports {ddr3_cas_n}]
223.
224. # PadFunction: IO_L13N_T2_MRCC_34
225. set_property SLEW FAST [get_ports {ddr3_we_n}]
226. set_property IOSTANDARD SSTL135 [get_ports {ddr3_we_n}]
227. set_property PACKAGE_PIN P5 [get_ports {ddr3_we_n}]
228.
229. # PadFunction: IO_0_34
230. set_property SLEW FAST [get_ports {ddr3_reset_n}]
231. set_property IOSTANDARD SSTL135 [get_ports {ddr3_reset_n}]
```

```
232. set_property PACKAGE_PIN K6 [get_ports {ddr3_reset_n}]
233.
234. # PadFunction: IO_L13P_T2_MRCC_34
235. set_property SLEW FAST [get_ports {ddr3_cke[0]}]
236. set_property IOSTANDARD SSTL135 [get_ports {ddr3_cke[0]}]
237. set_property PACKAGE_PIN N5 [get_ports {ddr3_cke[0]}]
238.
239. # PadFunction: IO_L19N_T3_VREF_34
240. set_property SLEW FAST [get_ports {ddr3_odt[0]}]
241. set_property IOSTANDARD SSTL135 [get_ports {ddr3_odt[0]}]
242. set_property PACKAGE_PIN R5 [get_ports {ddr3_odt[0]}]
243.
244. # PadFunction: IO_25_34
245. set_property SLEW FAST [get_ports {ddr3_cs_n[0]}]
246. set_property IOSTANDARD SSTL135 [get_ports {ddr3_cs_n[0]}]
247. set_property PACKAGE_PIN U8 [get_ports {ddr3_cs_n[0]}]
248.
249. # PadFunction: IO_L1P_T0_34
250. set_property SLEW FAST [get_ports {ddr3_dm[0]}]
251. set_property IOSTANDARD SSTL135 [get_ports {ddr3_dm[0]}]
252. set_property PACKAGE_PIN L1 [get_ports {ddr3_dm[0]}]
253.
254. # PadFunction: IO_L7P_T1_34
255. set_property SLEW FAST [get_ports {ddr3_dm[1]}]
256. set_property IOSTANDARD SSTL135 [get_ports {ddr3_dm[1]}]
257. set_property PACKAGE_PIN U1 [get_ports {ddr3_dm[1]}]
258.
259. # PadFunction: IO_L3P_T0_DQS_34
260. set_property SLEW FAST [get_ports {ddr3_dqs_p[0]}]
261. set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dqs_p[0]}]
262. set_property IOSTANDARD DIFF_SSTL135 [get_ports {ddr3_dqs_p[0]}]
263. set_property PACKAGE_PIN N2 [get_ports {ddr3_dqs_p[0]}]
264.
265. # PadFunction: IO_L3N_T0_DQS_34
266. set_property SLEW FAST [get_ports {ddr3_dqs_n[0]}]
267. set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dqs_n[0]}]
268. set_property IOSTANDARD DIFF_SSTL135 [get_ports {ddr3_dqs_n[0]}]
269. set_property PACKAGE_PIN N1 [get_ports {ddr3_dqs_n[0]}]
270.
271. # PadFunction: IO_L9P_T1_DQS_34
272. set_property SLEW FAST [get_ports {ddr3_dqs_p[1]}]
273. set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dqs_p[1]}]
274. set_property IOSTANDARD DIFF_SSTL135 [get_ports {ddr3_dqs_p[1]}]
275. set_property PACKAGE_PIN U2 [get_ports {ddr3_dqs_p[1]}]
```

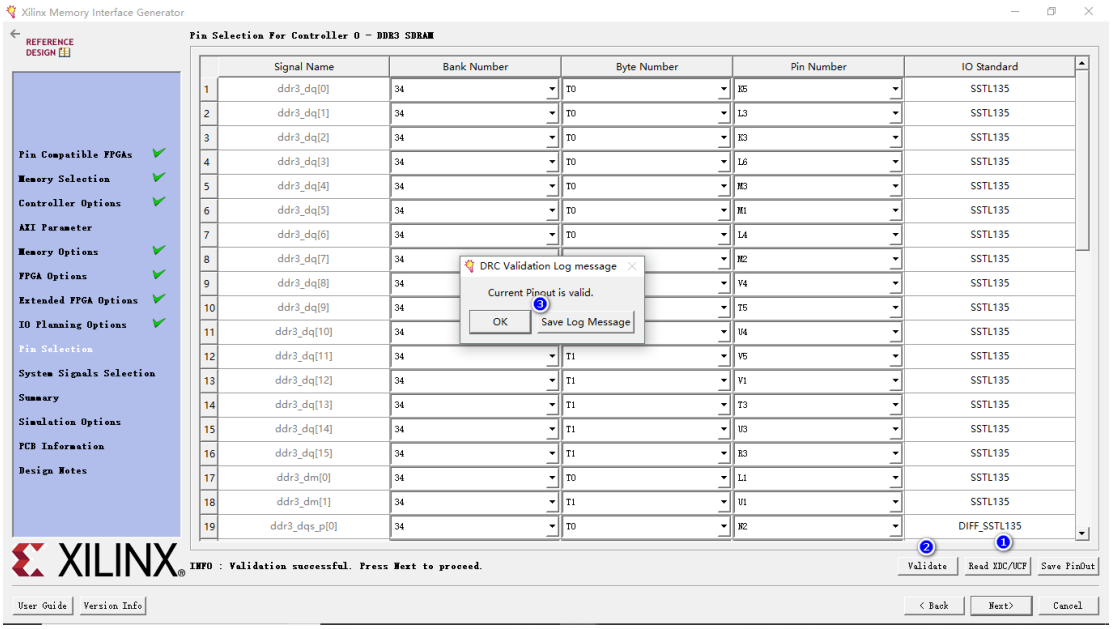
```
276.
277. # PadFunction: IO_L9N_T1_DQS_34
278. set_property SLEW FAST [get_ports {ddr3_dqs_n[1]}]
279. set_property IN_TERM UNTUNED_SPLIT_50 [get_ports {ddr3_dqs_n[1]}]
280. set_property IOSTANDARD DIFF_SSTL135 [get_ports {ddr3_dqs_n[1]}]
281. set_property PACKAGE_PIN V2 [get_ports {ddr3_dqs_n[1]}]
282.
283. # PadFunction: IO_L21P_T3_DQS_34
284. set_property SLEW FAST [get_ports {ddr3_ck_p[0]}]
285. set_property IOSTANDARD DIFF_SSTL135 [get_ports {ddr3_ck_p[0]}]
286. set_property PACKAGE_PIN U9 [get_ports {ddr3_ck_p[0]}]
287.
288. # PadFunction: IO_L21N_T3_DQS_34
289. set_property SLEW FAST [get_ports {ddr3_ck_n[0]}]
290. set_property IOSTANDARD DIFF_SSTL135 [get_ports {ddr3_ck_n[0]}]
291. set_property PACKAGE_PIN V9 [get_ports {ddr3_ck_n[0]}]
292.
293.
294. set_property INTERNAL_VREF  0.675 [get_iobanks 34]
295.
296. set_property LOC PHASER_OUT_PHY_X1Y1 [get_cells  -hier -filter {NAME =~ */d
    dr_phy_4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_B.ddr_byte_lane_B/phaser_out}]

297. set_property LOC PHASER_OUT_PHY_X1Y0 [get_cells  -hier -filter {NAME =~ */d
    dr_phy_4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_A.ddr_byte_lane_A/phaser_out}]

298. set_property LOC PHASER_OUT_PHY_X1Y3 [get_cells  -hier -filter {NAME =~ */d
    dr_phy_4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_D.ddr_byte_lane_D/phaser_out}]

299. set_property LOC PHASER_OUT_PHY_X1Y2 [get_cells  -hier -filter {NAME =~ */d
    dr_phy_4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_C.ddr_byte_lane_C/phaser_out}]

300.
301. ## set_property LOC PHASER_IN_PHY_X1Y1 [get_cells  -hier -filter {NAME =~ *
    /ddr_phy_4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_B.ddr_byte_lane_B/phaser_in_
    gen.phaser_in}]
302. ## set_property LOC PHASER_IN_PHY_X1Y0 [get_cells  -hier -filter {NAME =~ *
    /ddr_phy_4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_A.ddr_byte_lane_A/phaser_in_
    gen.phaser_in}]
303. set_property LOC PHASER_IN_PHY_X1Y3 [get_cells  -hier -filter {NAME =~ */dd
    r_phy_4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_D.ddr_byte_lane_D/phaser_in_gen
    .phaser_in}]
```

```
304. set_property LOC PHASER_IN_PHY_X1Y2 [get_cells  -hier -filter {NAME =~ */dd
     r_phy_4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_C.ddr_byte_lane_C/phaser_in_gen
     .phaser_in}]
305.
306.
307.
308. set_property LOC OUT_FIFO_X1Y1 [get_cells  -hier -filter {NAME =~ */ddr_phy
     _4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_B.ddr_byte_lane_B/out_fifo}]
309. set_property LOC OUT_FIFO_X1Y0 [get_cells  -hier -filter {NAME =~ */ddr_phy
     _4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_A.ddr_byte_lane_A/out_fifo}]
310. set_property LOC OUT_FIFO_X1Y3 [get_cells  -hier -filter {NAME =~ */ddr_phy
     _4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_D.ddr_byte_lane_D/out_fifo}]
311. set_property LOC OUT_FIFO_X1Y2 [get_cells  -hier -filter {NAME =~ */ddr_phy
     _4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_C.ddr_byte_lane_C/out_fifo}]
312.
313. set_property LOC IN_FIFO_X1Y3 [get_cells  -hier -filter {NAME =~ */ddr_phy_
     4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_D.ddr_byte_lane_D/in_fifo_gen.in_fifo
     }]
314. set_property LOC IN_FIFO_X1Y2 [get_cells  -hier -filter {NAME =~ */ddr_phy_
     4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_C.ddr_byte_lane_C/in_fifo_gen.in_fifo
     }]
315.
316. set_property LOC PHY_CONTROL_X1Y0 [get_cells  -hier -filter {NAME =~ */ddr_
     phy_4lanes_0.u_ddr_phy_4lanes/phy_control_i}]
317.
318. set_property LOC PHASER_REF_X1Y0 [get_cells  -hier -filter {NAME =~ */ddr_p
     hy_4lanes_0.u_ddr_phy_4lanes/phaser_ref_i}]
319.
320. set_property LOC OLOGIC_X1Y43 [get_cells  -hier -filter {NAME =~ */ddr_phy_
     4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_D.ddr_byte_lane_D/ddr_byte_group_io/*
     slave_ts}]
321. set_property LOC OLOGIC_X1Y31 [get_cells  -hier -filter {NAME =~ */ddr_phy_
     4lanes_0.u_ddr_phy_4lanes/ddr_byte_lane_C.ddr_byte_lane_C/ddr_byte_group_io/*
     slave_ts}]
322.
323. set_property LOC PLLE2_ADV_X1Y0 [get_cells -hier -filter {NAME =~ */u_ddr3_
     infrastructure/plle2_i}]
324. set_property LOC MMCME2_ADV_X1Y0 [get_cells -hier -filter {NAME =~ */u_ddr3
     _infrastructure/gen_mmcm.mmcm_i}]
325.
326.
327. set_multicycle_path -from [get_cells -hier -filter {NAME =~ */mc0/mc_read_i
     dle_r_reg}] \
```

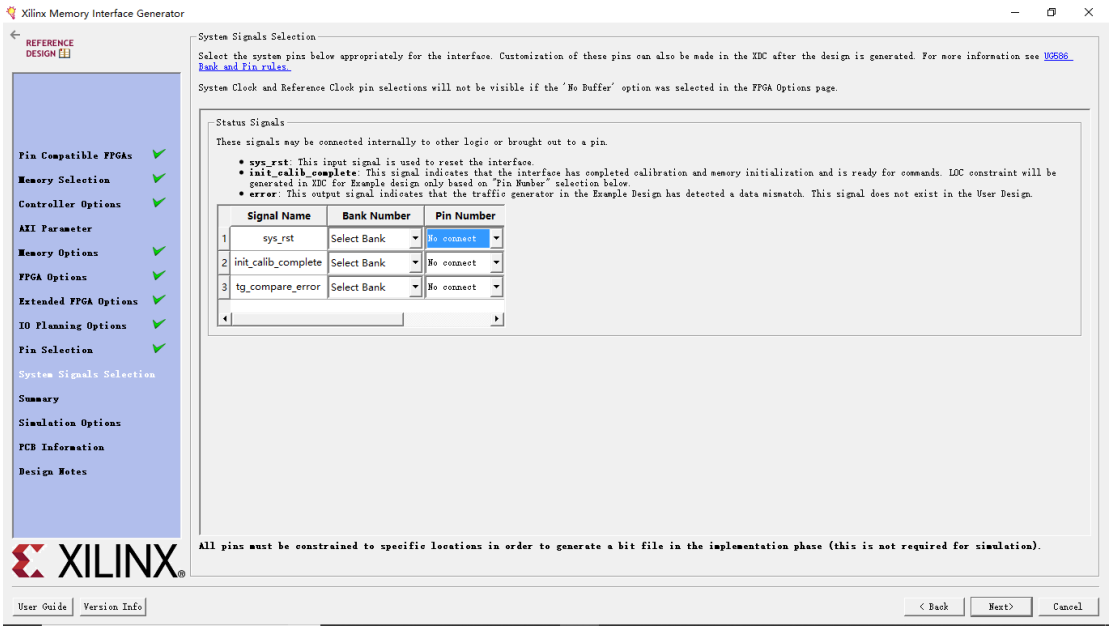
```
                    -to   [get_cells -hier -filter {NAME =~ */input_[?].iserdes_dq_.iserdesdq}] \
                    -setup 6

set_multicycle_path -from [get_cells -hier -filter {NAME =~ */mc0/mc_read_idle_r_reg}] \
                    -to   [get_cells -hier -filter {NAME =~ */input_[?].iserdes_dq_.iserdesdq}] \
                    -hold 5

set_false_path -through [get_pins -filter {NAME =~ */DQSFOUND} -of [get_cells -hier -filter {REF_NAME == PHASER_IN_PHY}]]

set_multicycle_path -through [get_pins -filter {NAME =~ */OSERDESRST} -of [get_cells -hier -filter {REF_NAME == PHASER_OUT_PHY}]] -setup 2 -start
set_multicycle_path -through [get_pins -filter {NAME =~ */OSERDESRST} -of [get_cells -hier -filter {REF_NAME == PHASER_OUT_PHY}]] -hold 1 -start

set_max_delay -datapath_only -from [get_cells -hier -filter {NAME =~ *temp_mon_enabled.u_tempmon/* && IS_SEQUENTIAL}] -to [get_cells -hier -filter {NAME =~ *temp_mon_enabled.u_tempmon/device_temp_sync_r1*}] 20
set_max_delay -from [get_cells -hier *rstdiv0_sync_r1_reg*] -to [get_pins -filter {NAME =~ */RESET} -of [get_cells -hier -filter {REF_NAME == PHY_CONTROL}]] -datapath_only 5
set_false_path -through [get_pins -hier -filter {NAME =~ */u_iodelay_ctrl/sys_rst}]

set_max_delay -datapath_only -from [get_cells -hier -filter {NAME =~ *ddr3_infrastructure/rstdiv0_sync_r1_reg*}] -to [get_cells -hier -filter {NAME =~ *temp_mon_enabled.u_tempmon/xadc_supplied_temperature.rst_r1*}] 20

```



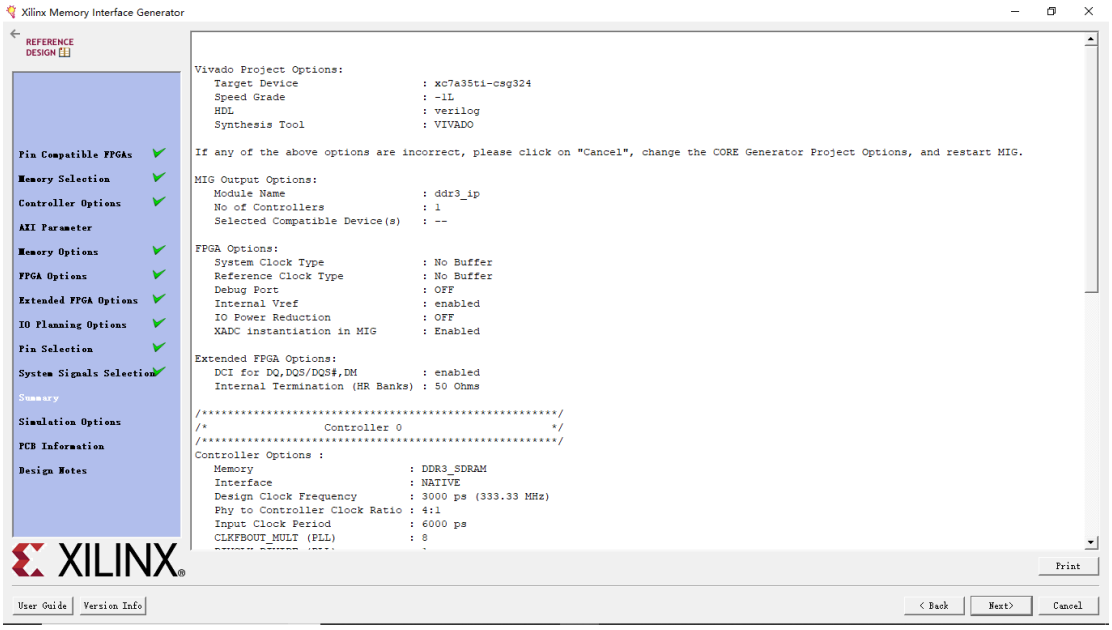
12>点击 Validate 验证管脚约束是否有错误。验证通过 NEXT。



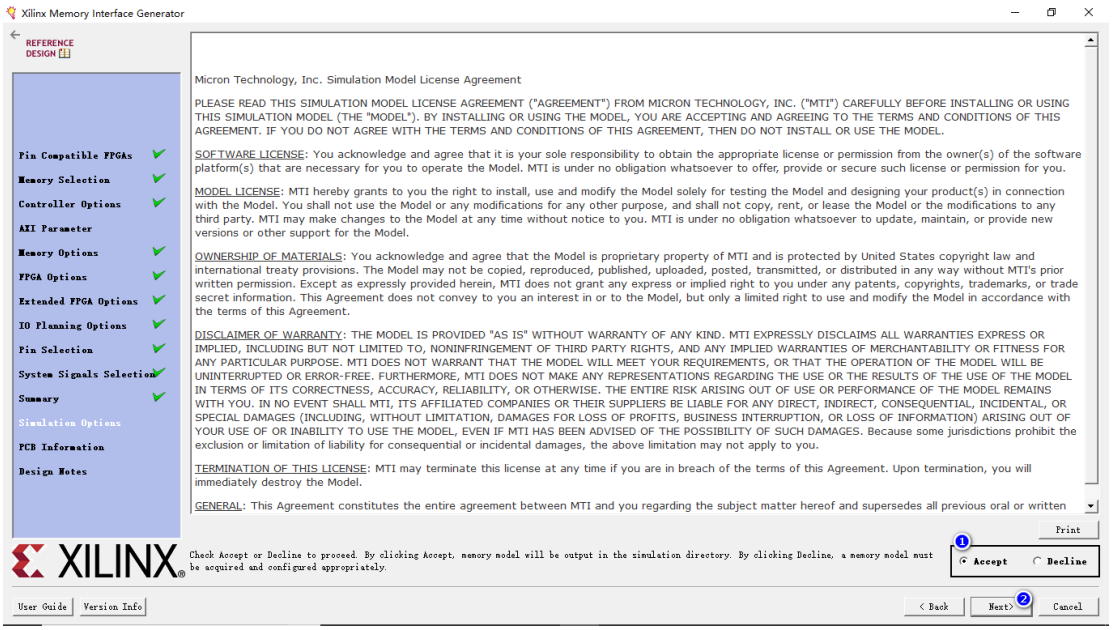
13>默认，NEXT。



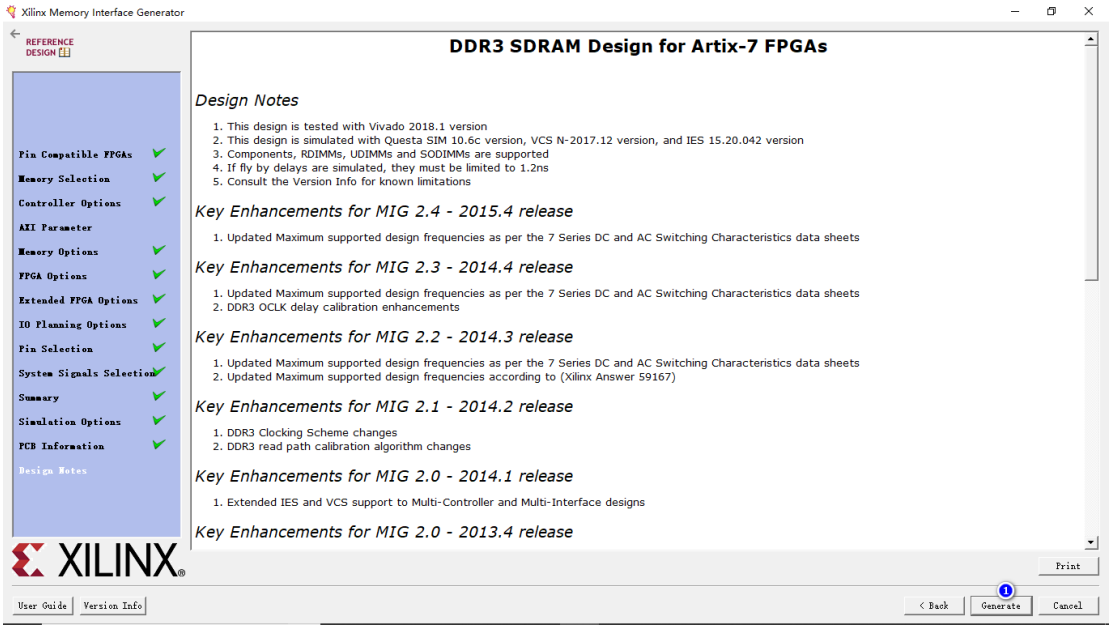
14>Next。



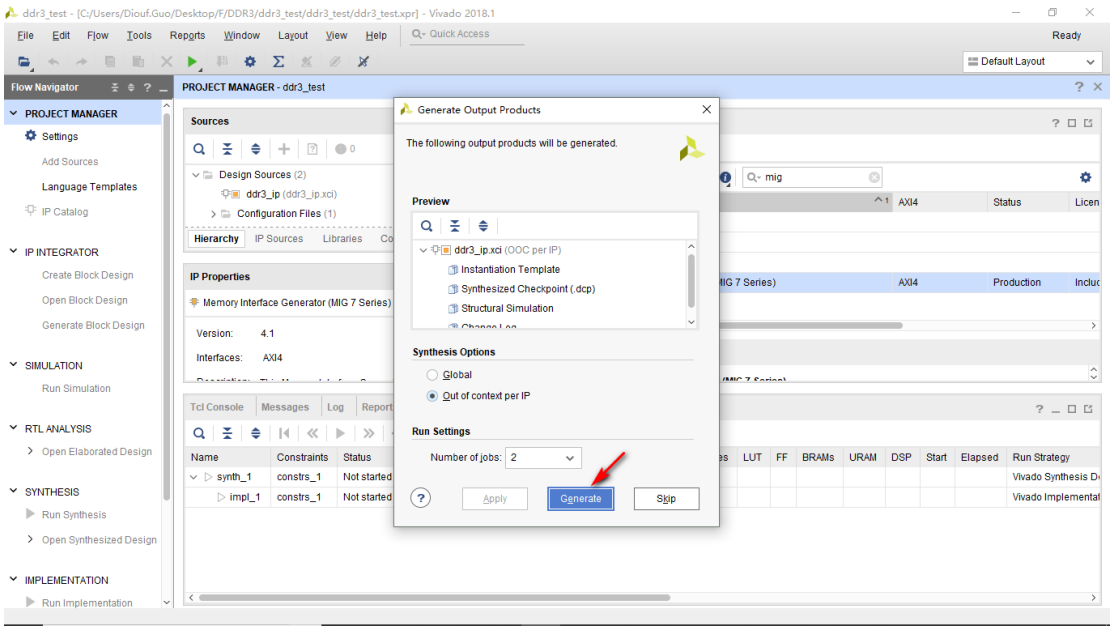
15>Accept,Next。



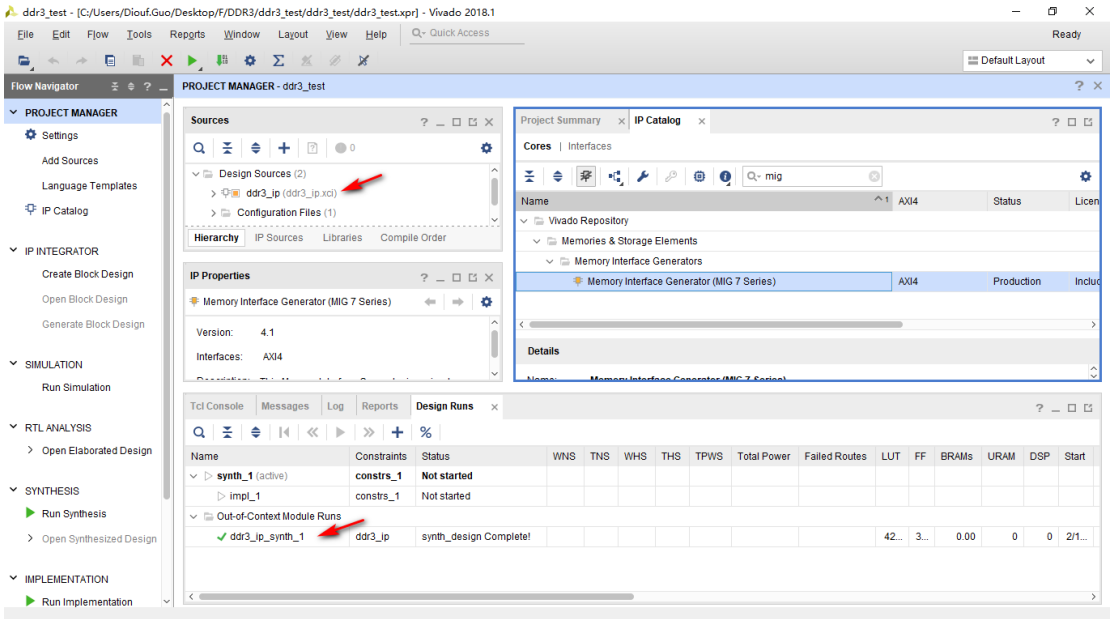
16>Generate



17>Generate



18>至此我们的 mig IP 创建完成。



FPGA 开源工作室为了大家更好更快的学习 FPGA 目前开通了知识星球(FPGA 自习学院)。FPGA 自习学院将不断更新和总结 FPGA 相关的学习资料，欢迎大家加入，一起学习一起成长。


知识星球
FPGA自习学院
星主: OpenS_Lee
长按扫码预览社群内容
和星主关系更近一步