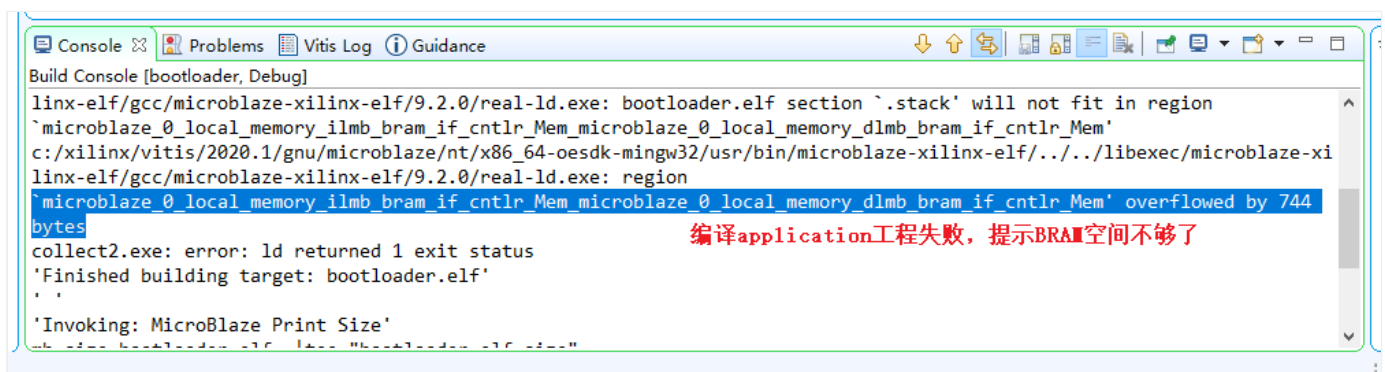


Xilinx Vitis 2020.1 method to modify the size of BRAM memory occupied by the project

tags: [FPGA](#) [Vitis](#) [BRAM](#) [Microblaze](#) [Xilinx](#)

After modifying the code and compiling the project, sometimes the compilation fails due to insufficient BRAM space, and the following error message appears:



'Invoking: MicroBlaze gcc linker'

mb-gcc -Wl,-T -Wl,.../src/lscript.ld -

LE:/fpga_projects/test4/microblaze_test/export/microblaze_test/sw/microblaze_test/sta
ndalone_domain/bsplib/lib -mlittle-endian -mcpu=v11.0 -mxl-soft-mul -Wl,--no-relax -
Wl,--gc-sections -o "bootloader.elf" ./src/bootloader.o ./src/platform.o ./src/srec.o -Wl,-
-start-group,-lxil,-lgcc,-lc,--end-group -Wl,--start-group,-lxilisf,-lxil,-lgcc,-lc,--end-group
c:/xilinx/vitis/2020.1/gnu/microblaze/nt/x86_64-oesdk-mingw32/usr/bin/microblaze-
xilinx-elf/././libexec/microblaze-xilinx-elf/gcc/microblaze-xilinx-elf/9.2.0/real-ld.exe:
bootloader.elf section `'.stack' will not fit in region

TOP

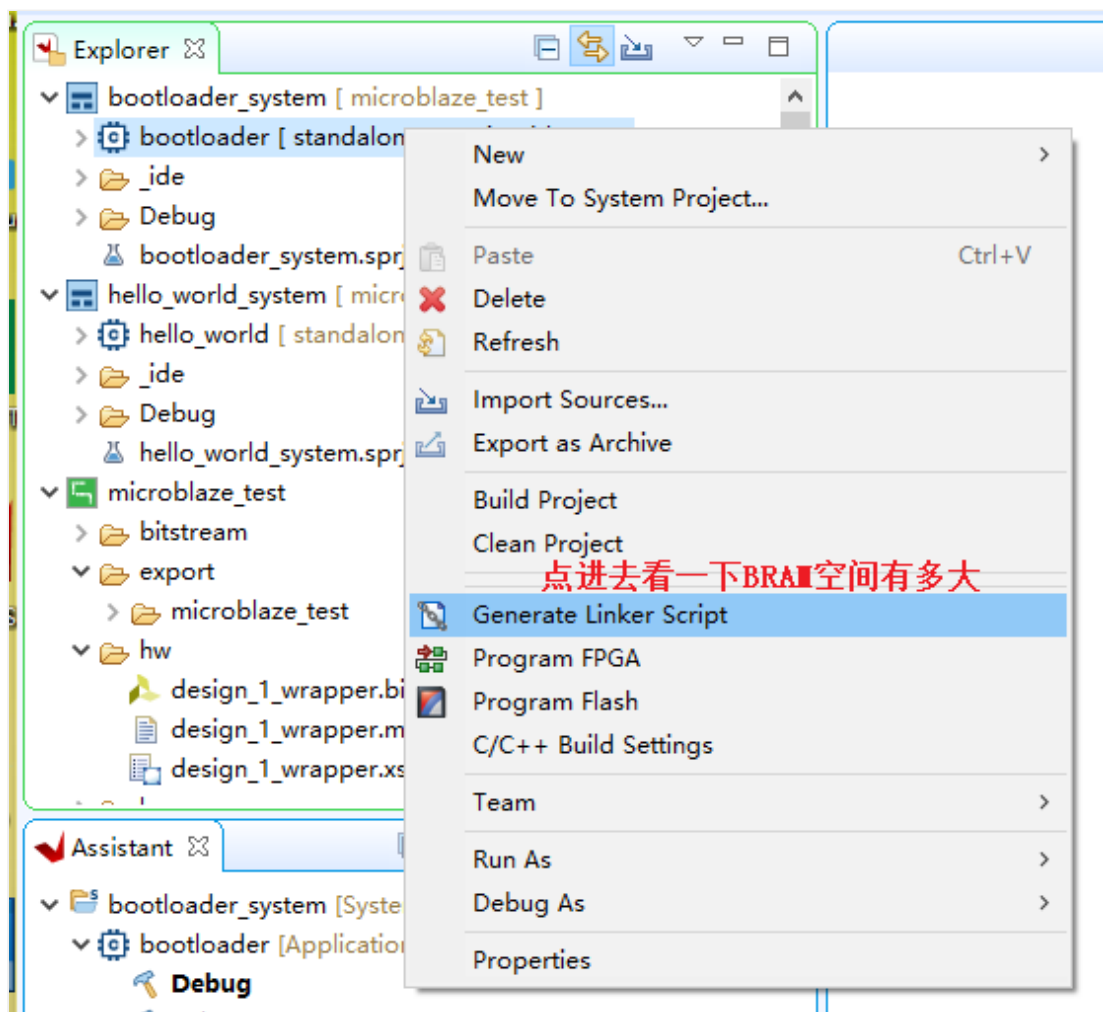
```
`microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microblaze_0_local_memory_  
dlmb_bram_if_cntlr_Mem'
```

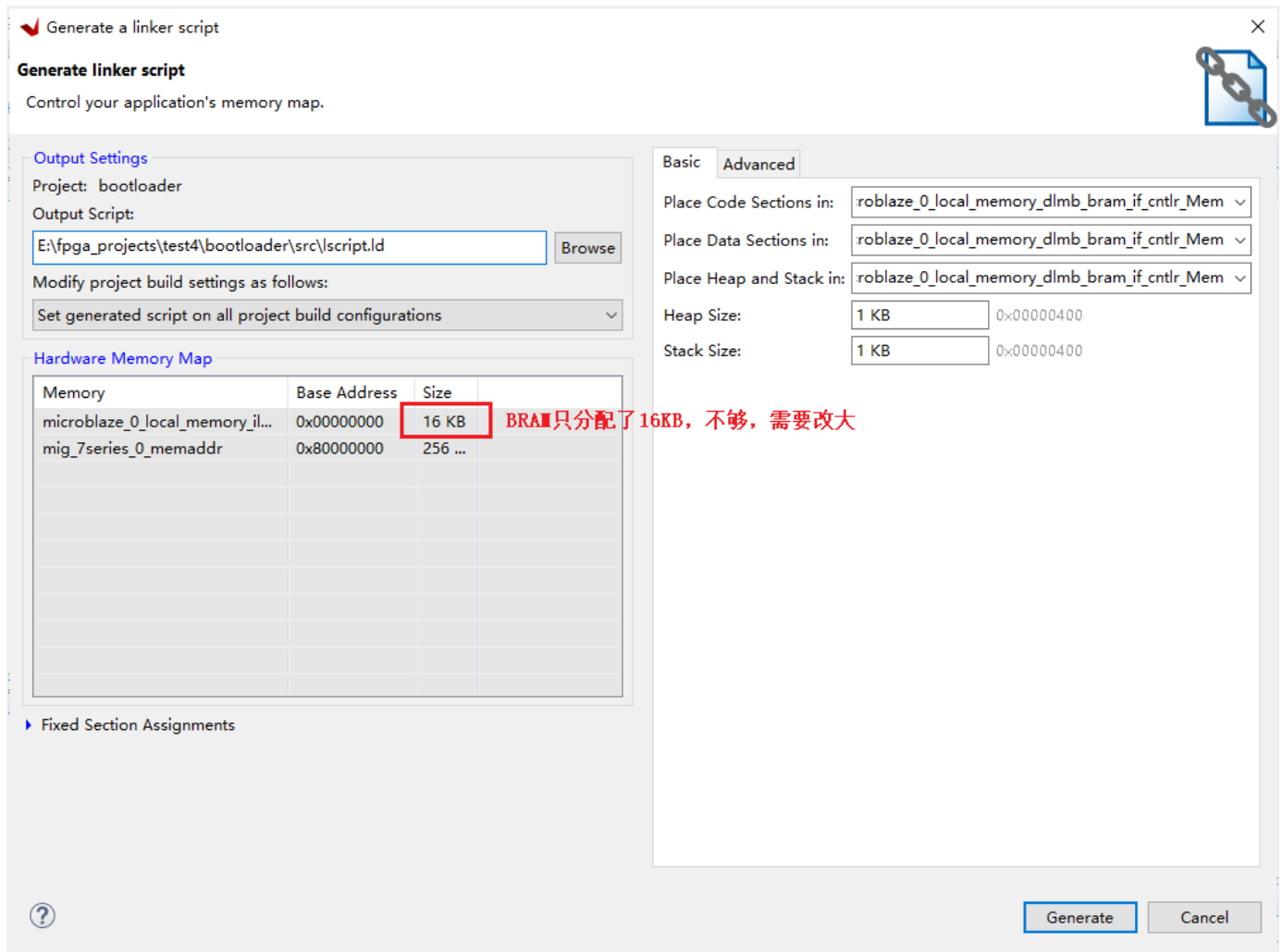
```
c:/xilinx/vitis/2020.1/gnu/microblaze/nt/x86_64-oesdk-mingw32/usr/bin/microblaze-  
xilinx-elf/../../libexec/microblaze-xilinx-elf/gcc/microblaze-xilinx-elf/9.2.0/real-ld.exe:  
region
```

```
`microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microblaze_0_local_memory_  
dlmb_bram_if_cntlr_Mem' overflowed by 744 bytes
```

```
collect2.exe: error: ld returned 1 exit status
```

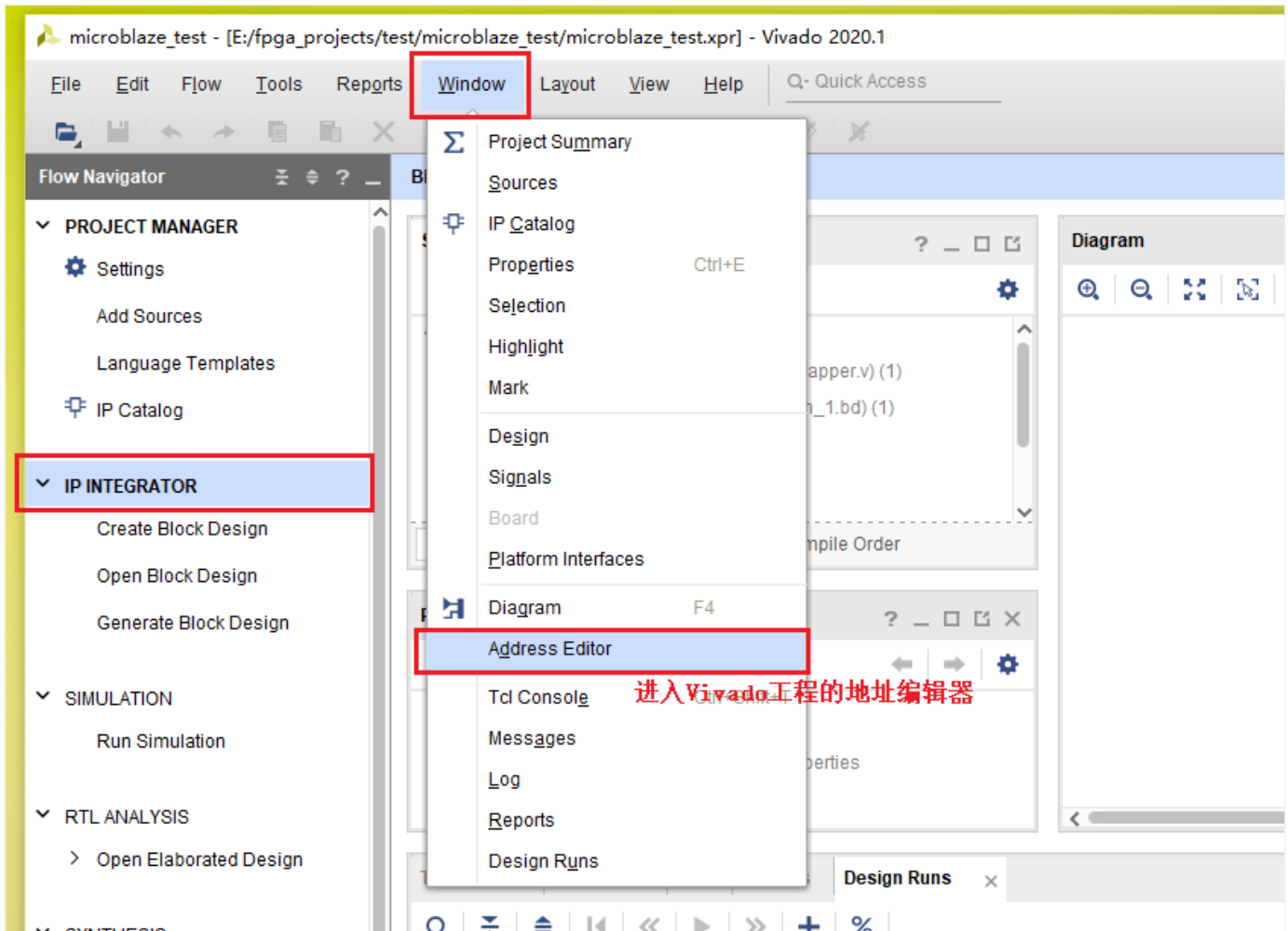
```
'Finished building target: bootloader.elf'
```



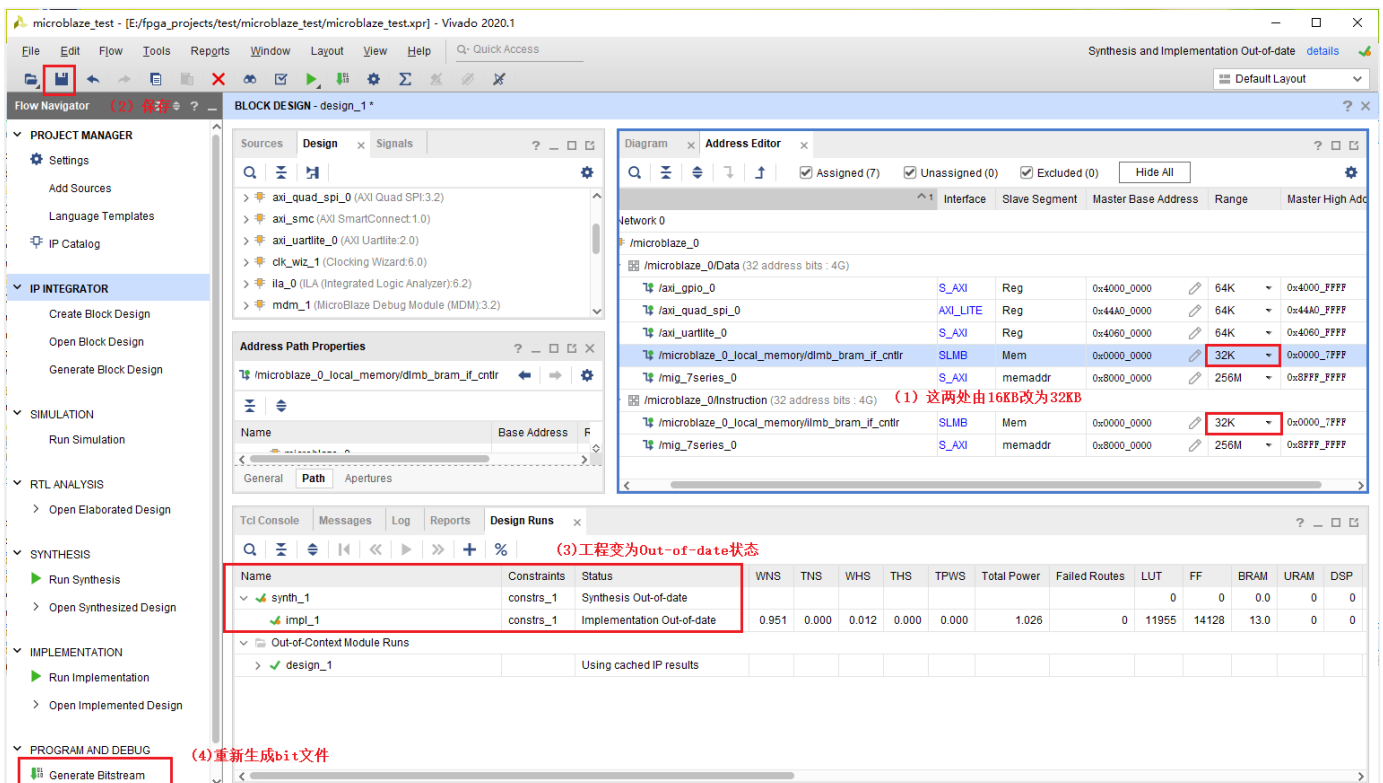


At this time, you need to go back to the Vivado project and increase the size of the allocated BRAM space in Block Design. The specific method is as follows.

Enter the Address Editor of IP INTEGRATOR:

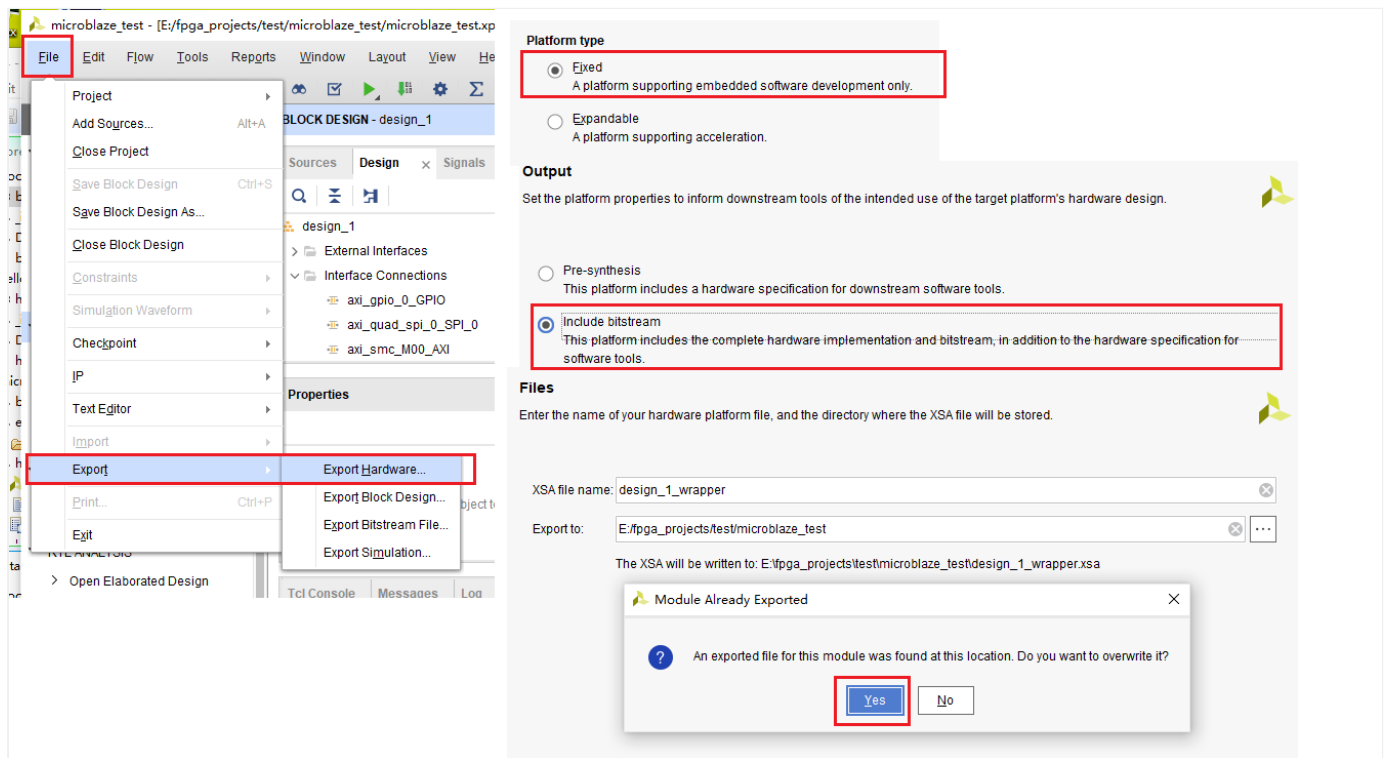


Modify the BRAM space size from 16KB to 32KB, save, and then regenerate the bit file.

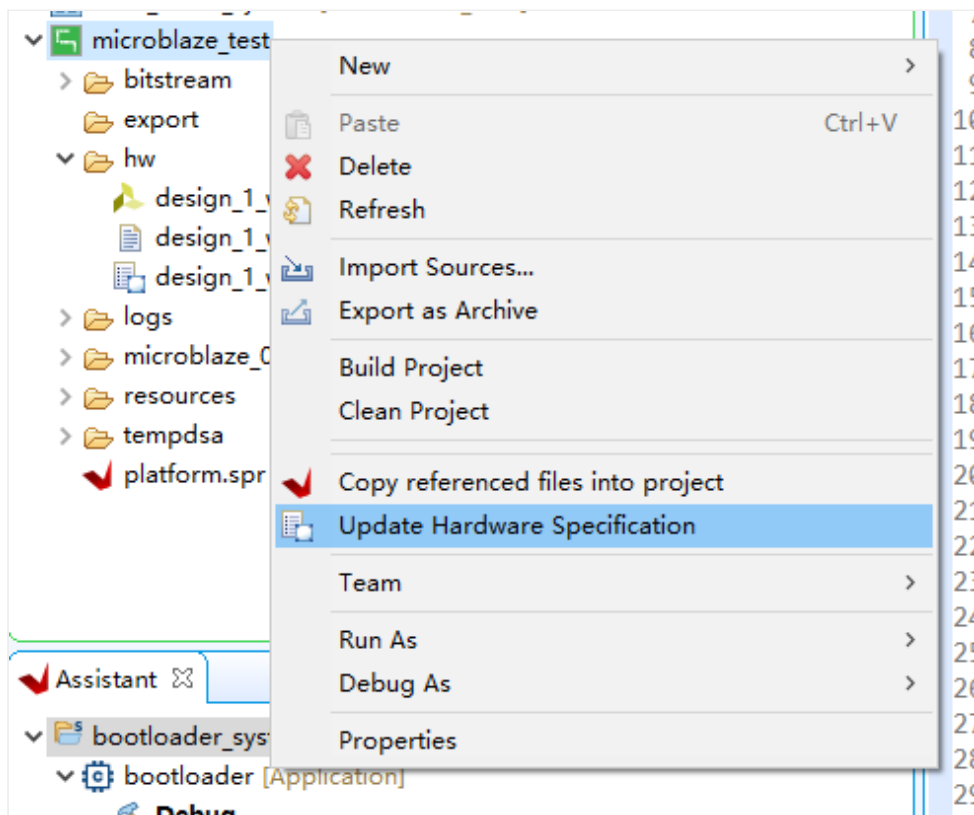


TOP

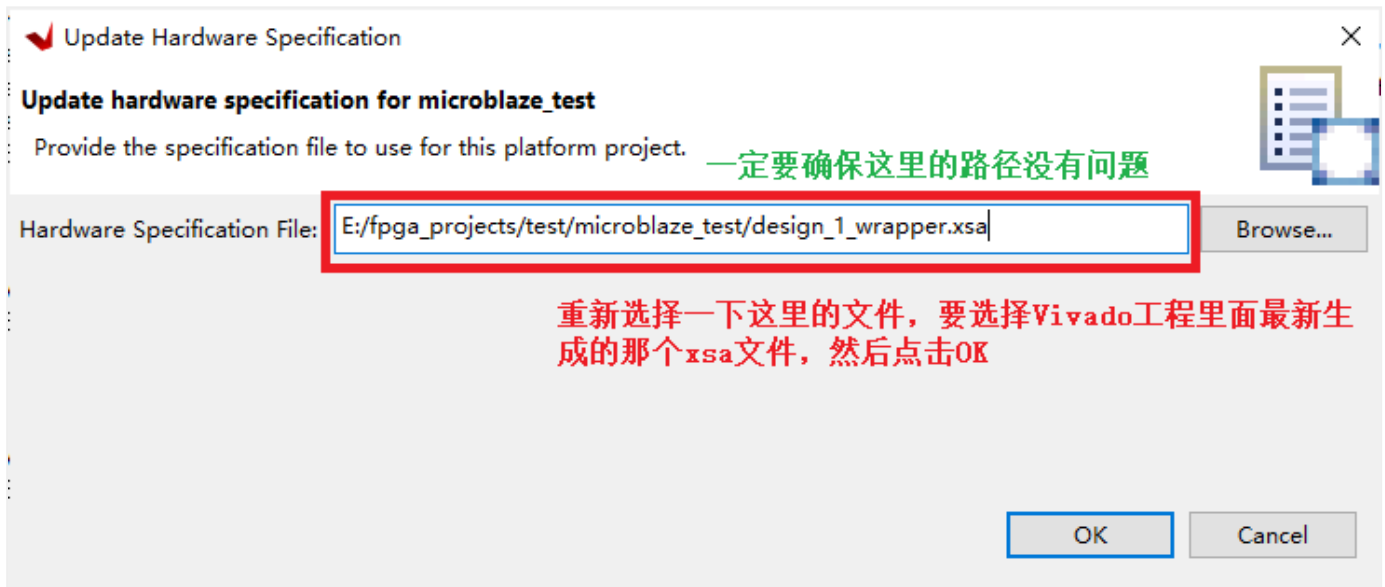
After generating the bit file, export the xsa file again:



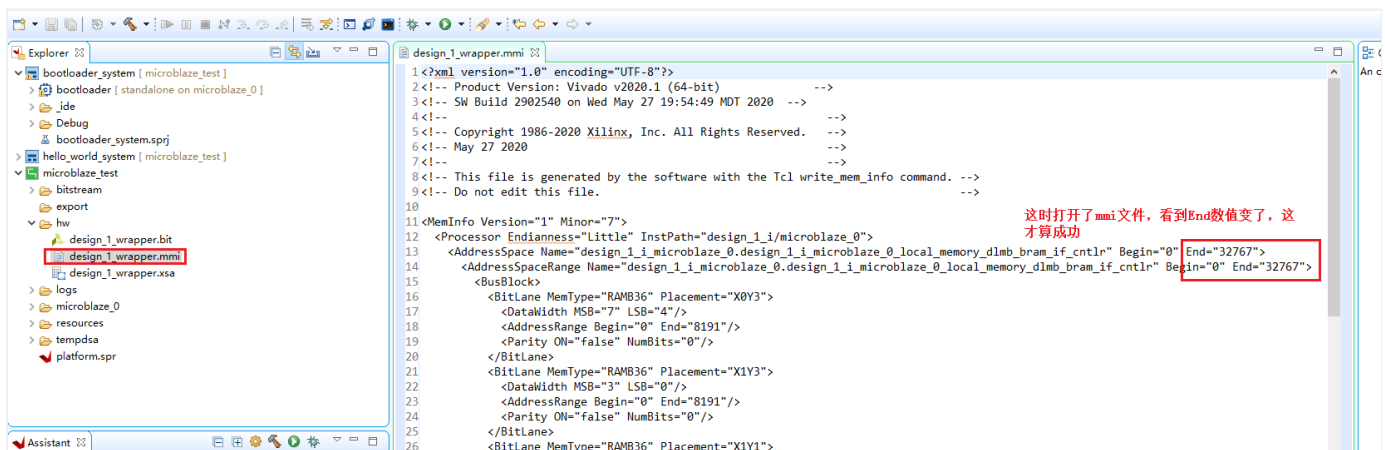
Go back to Vitis and re-import the xsa file:



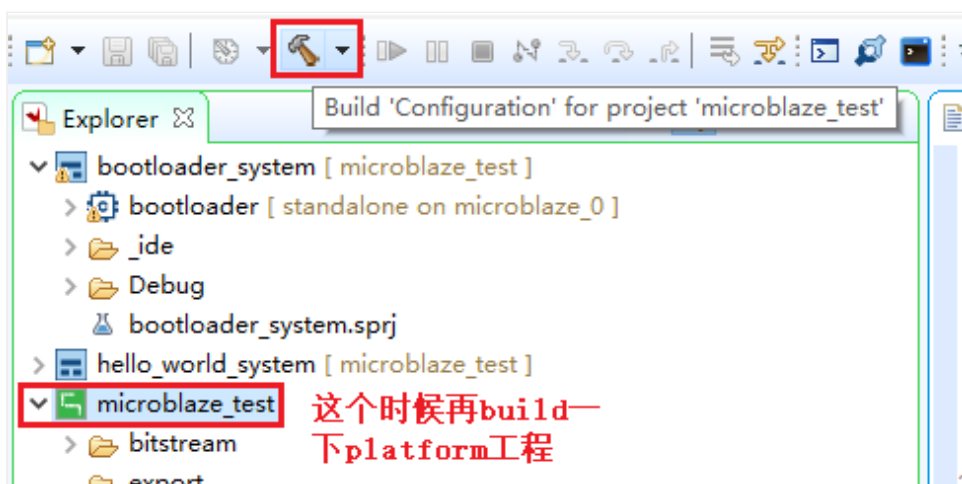
Take a closer look at the path of the xsa file here. Sometimes the path here is wrong and you need to select the correct file again.



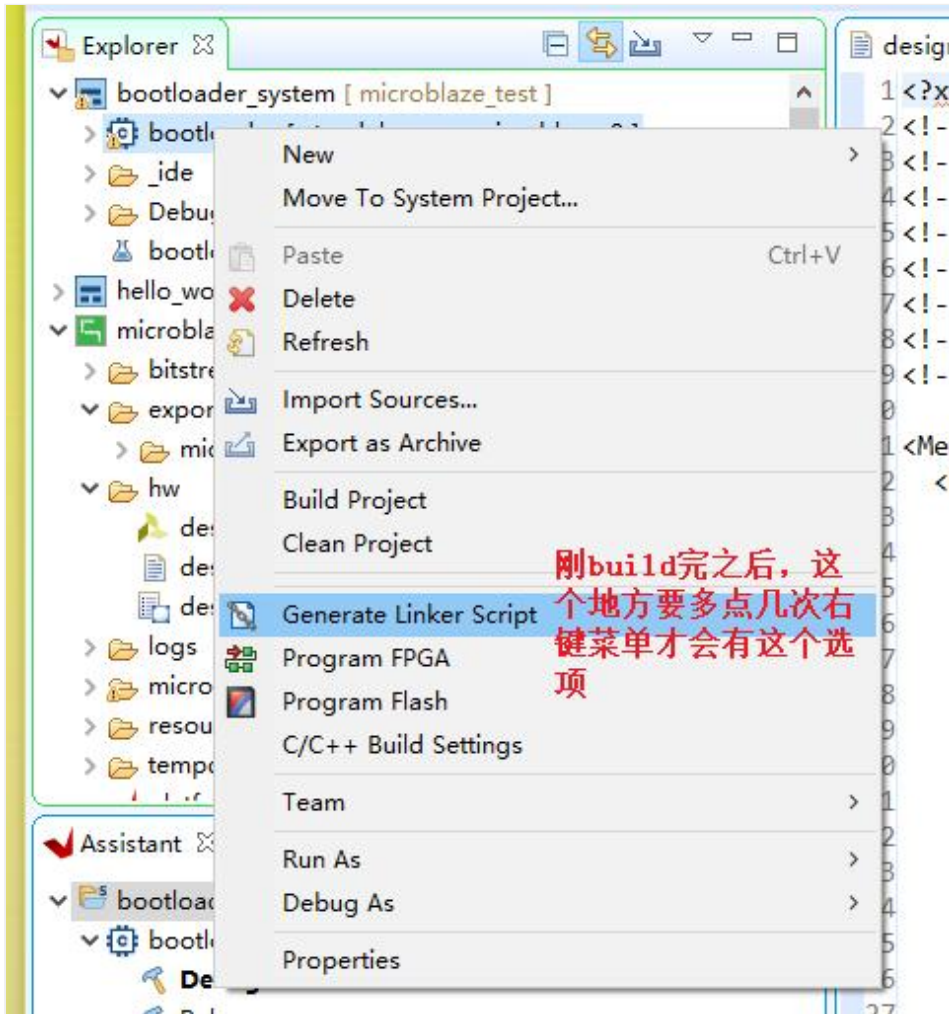
After clicking OK to close the dialog box, open the mmi file and take a look to see if the modified BRAM size really takes effect. If it does not take effect, the path of the xsa file you just selected is wrong.



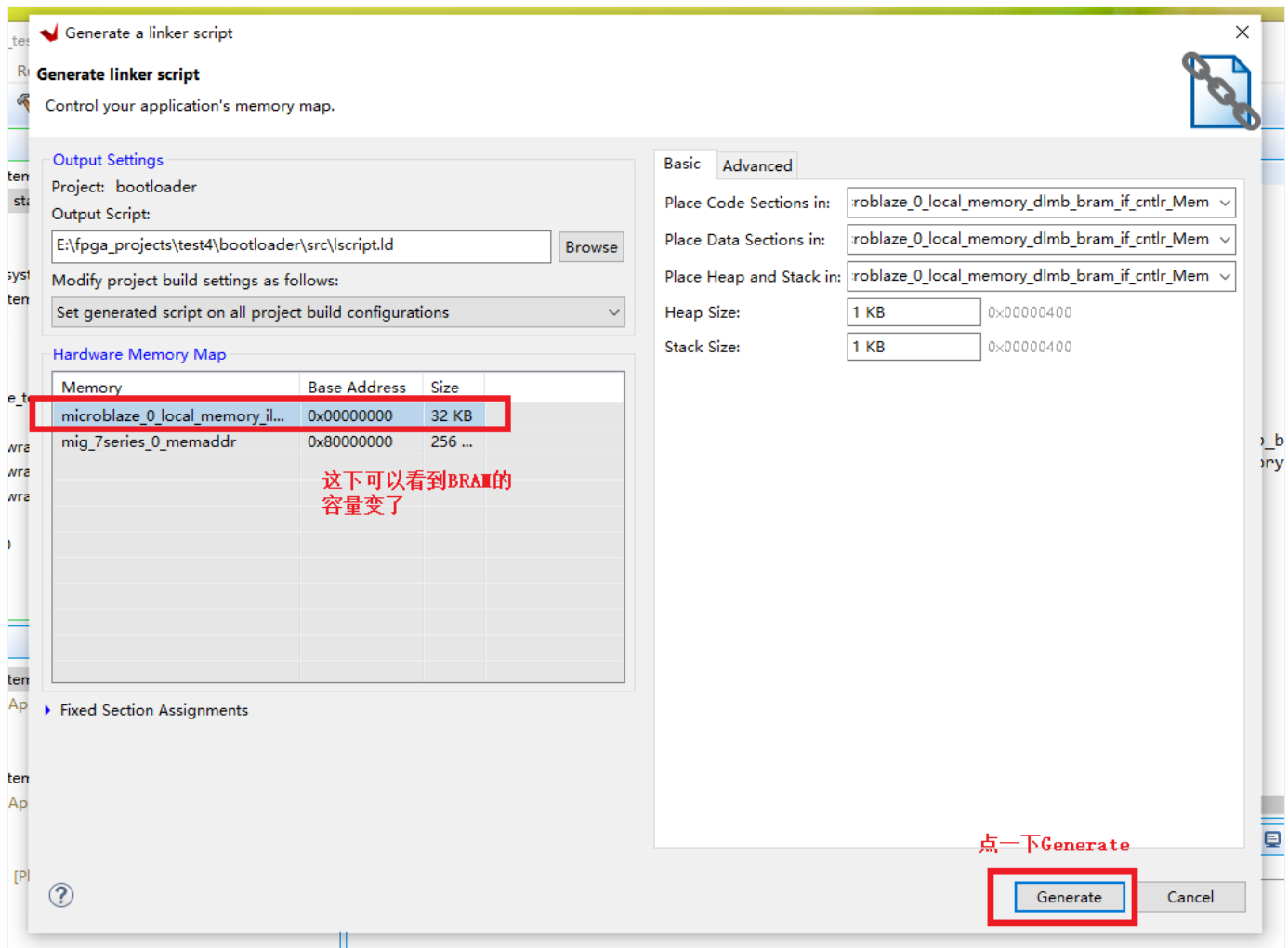
After confirming that the size is correct, you must build the platform project (the green project) at this time.



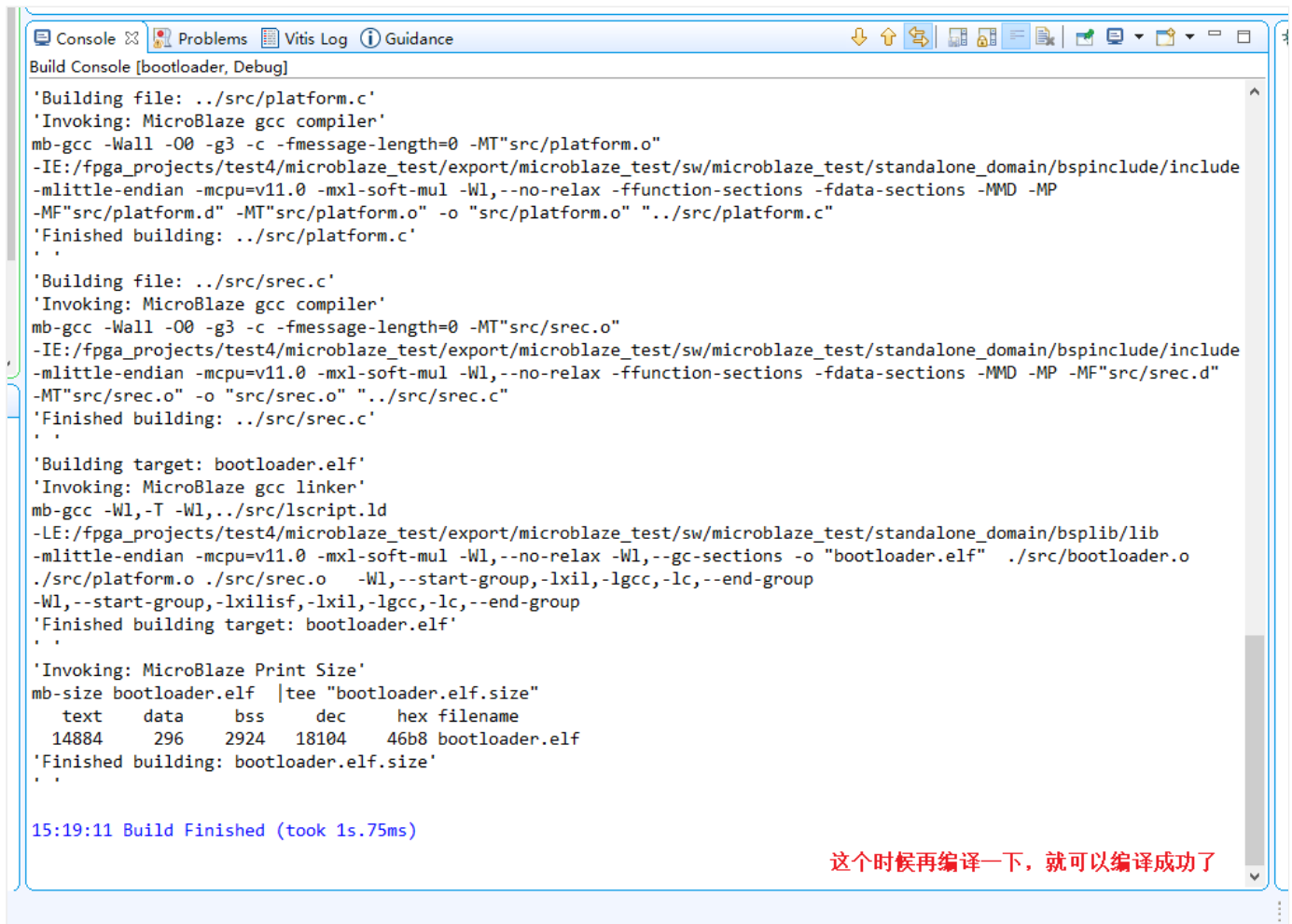
After the build is completed, check whether the BRAM size in the application project is effective. Here, the right-click menu usually pops up several times before the Generate Linker Script menu item comes out.



You can see that the size of BRAM has taken effect and has become 32KB. At this time, click the Generate button to update the lscript.ld file.



BRAM space is enough, the project can be compiled and passed:



```

Build Console [bootloader, Debug]

'Building file: ../src/platform.c'
'Invoking: MicroBlaze gcc compiler'
mb-gcc -Wall -O0 -g3 -c -fmessage-length=0 -MT"src/platform.o"
-IE:/fpga_projects/test4/microblaze_test/export/microblaze_test/sw/microblaze_test/standalone_domain/bspinclude/include
-mlittle-endian -mcpu=v11.0 -mxl-soft-mul -Wl,--no-relax -ffunction-sections -fdata-sections -MMD -MP
-MF"src/platform.d" -MT"src/platform.o" -o "src/platform.o" "../src/platform.c"
'Finished building: ../src/platform.c'
..

'Building file: ../src/srec.c'
'Invoking: MicroBlaze gcc compiler'
mb-gcc -Wall -O0 -g3 -c -fmessage-length=0 -MT"src/srec.o"
-IE:/fpga_projects/test4/microblaze_test/export/microblaze_test/sw/microblaze_test/standalone_domain/bspinclude/include
-mlittle-endian -mcpu=v11.0 -mxl-soft-mul -Wl,--no-relax -ffunction-sections -fdata-sections -MMD -MP -MF"src/srec.d"
-MT"src/srec.o" -o "src/srec.o" "../src/srec.c"
'Finished building: ../src/srec.c'
..

'Building target: bootloader.elf'
'Invoking: MicroBlaze gcc linker'
mb-gcc -Wl,-T -Wl,../src/lscript.ld
-LE:/fpga_projects/test4/microblaze_test/export/microblaze_test/sw/microblaze_test/standalone_domain/bsplib/lib
-mlittle-endian -mcpu=v11.0 -mxl-soft-mul -Wl,--no-relax -Wl,--gc-sections -o "bootloader.elf" ./src/bootloader.o
./src/platform.o ./src/srec.o -Wl,--start-group,-lxil,-lgcc,-lc,--end-group
-Wl,--start-group,-lxilisf,-lxil,-lgcc,-lc,--end-group
'Finished building target: bootloader.elf'
..

'Invoking: MicroBlaze Print Size'
mb-size bootloader.elf |tee "bootloader.elf.size"
  text    data    bss     dec     hex filename
 14884    296    2924    18104   46b8 bootloader.elf
'Finished building: bootloader.elf.size'
..

15:19:11 Build Finished (took 1s.75ms)

```

这个时候再编译一下，就可以编译成功了

The above are the complete steps to modify the project BRAM size. As long as the above steps are wrong, the modification will be unsuccessful. The whole process is quite complicated.

[Copyright Complaint](#) [Spam Report](#)

Intelligent Recommendation

Xilinx FPGA BRAM study notes

Chip model: XC7A100T Camera model: OV5640 Resolution: 320*240 BRAM type: single dual port bram The thing to do is to display the camera data on a VGA monitor. The solution is to first put the r...

C++--The size of the memory occupied by the structure

TOP

Structure occupied memory size In vs2010, use sizeof to display 80 bytes of memory occupied. The essence of the structure in memory size is: byte alignment The offset of each member of the structure r...



A JPG's memory size occupied in Android

2019 Unicorn Enterprise Heavy Recruitment Python Engineer
Standard >>> A JPG file, is displayed, it is to be converted into Bitmap, then how much memory space is this BitAmp takes up?
The fol...

Xilinx Vitis Petalinux installation and use

ubuntu16.04.6 Install Vitis, execute, According to the default settings, the desktop shortcut installation failed, it should be my Chinese version of ubuntu, the desktop shortcut installation error, j...



Xilinx Vitis Shortcut Binds and View

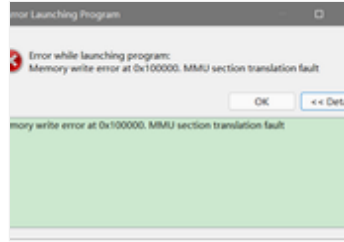
Shortcut key settings & views Learning to develop Zynq
Ultrascale +, VITIS is numerous, and there are many shortcuts.
You can modify the shortcut key settings in Window->
Preferences-> Keys ...

More Recommendation

Install Vivado Vitis Petalinux 2020.1 on Ubuntu18.04

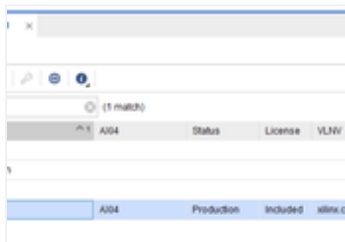
The latest version of the Xilinx tool has been updated to 2020.2, I installed 2020.1 mainly because Pynq latest version V2.6 support 2020.1, this year will do some Pynq related development. 1 New virt...

[TOP](#)



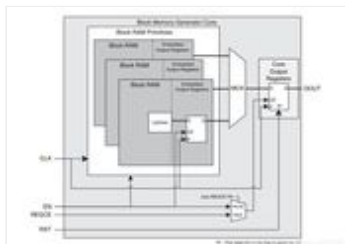
Xilinx Vitis Error Launchin g Program: Memory write e rror MMU section translati on fault

An error is reported at Run As -> Launch Hardware (Single Application Debug (GDB)):
Error while launching program: Memory write error at 0x100000. MMU section
translation fault The reason is that t...



Xilinx 2020.1 MIG Nuclear Read Write DDR3 Memory, Configuring MIG Nucleus When New Engineering

This paper uses this chip in XC7A35TFGG484-2 as an example, using the MILE FPGA development board, drive DDR3 memory with MIG core. FPGA external crystal size is 50 MHz, the drive frequency of DDR3 me...



Xilinx BlockRam (Bram) structure and read logic interpretation

As shown in the figure, bram consists of a lot of blocks and registers of ram blocks, through the Mux data selector, the data is sent to the output register, and finally output. As can be seen

from th...



Xilinx Bram IP core operating instructions

1. What is BRAM Search ram in vivado2018.3, configure IPcore according to the following figures, here first use simple

dual-port ram, use native interface The configuration data width is 8 bits...

Related Posts

- [Tutorial] Xilinx Vivado / Vitis 2020.1 Create a MicroBlaze project to run the Hello World C language program (using external DDR3 memory)
- [Tutorial] Xilinx Vivado / Vitis 2020.1 Create the simplest MicroBlaze project to run the Hello World C language program (do not use external DDR3 memory), and solidify to SPI Flash
- Xilinx Vitis 2020.1 cannot create SREC SPI Bootloader project, prompting the solution of lack of Xilisf library
- Xilinx Vitis
- Based on Xilinx, Vitis platforms read and write Bram infrared remote control experiments (2)
- Based on the infrared remote control experiments read and write BRAM based on Xilinx, Vitis platform (1)
- Xilinx Vitis 2020.1 Curing the MicroBlaze program with SREC SPI Bootloader, the solution that does not run after the power-on power-on
- Questions about the Sleep function card of the MicroBlaze soft core in Xilinx VITIS 2020.1
- I can't find MicroBlaze_0 when running a C program in Xilinx Vitis 2020.1
- Tomcat modify memory size method

Popular Posts

- Verilog design work
- ios UIView common animation effects
- Scala study notes (2)-basic grammar
- Java Mac computers need to pay attention to learning Java
- H5 uses input type = file to implement a direct call camera
- Numpy - Basic knowledge
- The difference between fair lock and unfair lock in AQS
- IIC control of VC707 development board
- Supplementary conversion, operation
- CentOS 7 Firewalls are frequently used settings

Recommended Posts

- Enter two sequences of integers. The first sequence indicates the push order of the stack. Please determine if the second sequence is the pop-up order of the stack. Assume that all numbers pushed onto the stack are not equal.
- Algorithm and Data Structure Analysis (1)
- Java internal class and exception class
- Overview of JavaBean
- [Kuangbin] four thematic shortest exercises Silver Cow Party POJ - 3268] [dijkstra
- Federal study notes (2)
- LeetCode - 50.Pow(x,n)
- [MyBatis] Architecture layering and main objects
- Page directory entry and page table entry
- Netease mailbox front-end Javascript coding specification: basic specification

Related Tags

FPGA

Xilinx

MicroBlaze

DDR3

MIG

Vitis

TOP

[Solidify](#)

[Vivado](#)

[SREC](#)

[FPGA development](#)

Copyright **DMCA** © 2018-2023 - All Rights Reserved -
www.programmersought.com **User Notice**