Embedded Linux

Home (Https://numato.com/help) / Knowledge Base (Https://numato.com/kb/) / Saturn Spartan 6 FPGA Module (Https://numato.com/kb-category/saturn-spartan-6-fpga-module/)

Have a question? Enter a search term SEARCH

Popular Search: USB GPIO (), USB Relay (), FPGA ()

Saturn, Microblaze and Linux – How to Run Linux on Saturn Spartan 6 FPGA Module – Part III

1699 views March 9, 2016 admin 4

(https://numato.com/help/wp-content/uploads/2016/03/microblaze_Linux.png)

- **Part I (http://docs.numato.com/knowledge/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-i/)**
- **Part II (https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-ii/)**
- **Part III**
- **Part IV (https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-iv/)**

In Part I and II of this article series, we saw how to create a Microblaze based embedded platform for Saturn Spartan 6 Module (https://numato.com/saturn-spartan-6-fpga-development-board-with-ddr-sdram) and build Linux kernel for it. In this tutorial we will take a look at how to download the kernel image to Saturn and boot using Xilinx Platform cable USB and XMD (Xilinx Microprocessor Debugger). Xilinx Platform cable USB slightly expensive, but if you don't have one don't loose hope. There is still a way to download the kernel and boot. This is discussed in Part IV (https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-iv/) of this article series.

Before we continue, we need to make sure that the channel B of the onboard FT2232H is configured as USB serial port. Configuring FT2232H is covered in the article Configuring FT2232H USB Serial converter on Saturn Spartan 6 Module (https://numato.com/kb/configuring-ft2232h-usb-serial-converter-saturn-spartan-6-module/). Please follow the steps in the article and configure FT2232H appropriately. Once successfully configured, a new serial port should appear in Windows Device Manager that corresponds to FT2232H channel B. We will use this serial port as a serial terminal through which we will interact with Linux.

Assuming you have hooked up Xilinx Platform cable USB properly, we will open the SDK project that we created in Part II of this article series (https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-ii/) (in Xilinx SDK of course !). Make sure the project builds without errors. Go to "Xilinx Tools" menu in SDK and select "Program FPGA". When FPGA programming dialog pops up, leave all settings as is and click on "Program" button and let FPGA programming to finish.

Once FPGA programming is finished, we can now communicate with the Microblaze processor through XMD. It is possible to invoke XMD within SDK but I found it very easy and convenient to invoke XMD in a standalone commandline (DOS) window.

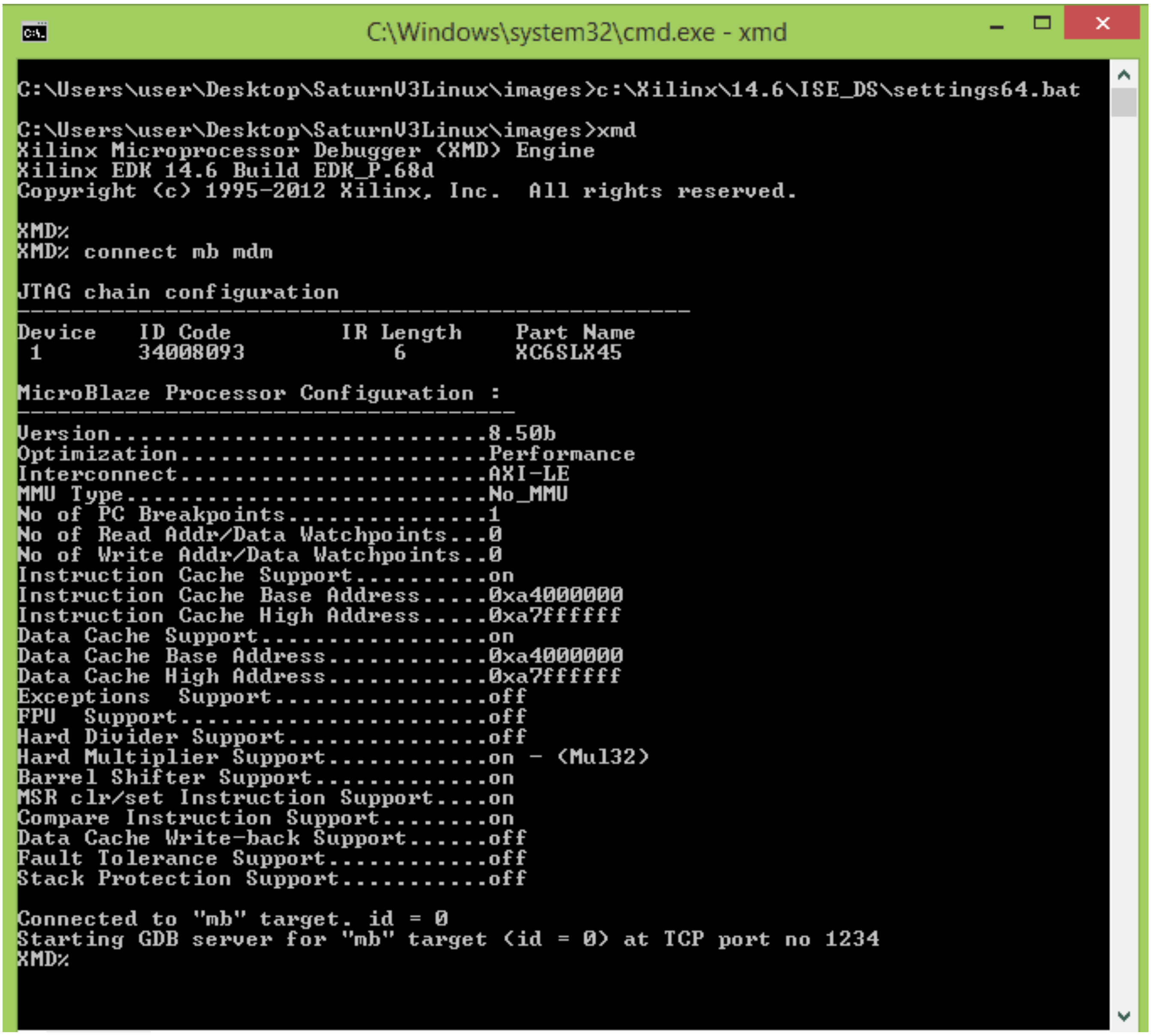
Before starting XMD, copy “simpleImage.saturn_v3” in the folder /output/images under buildroot directory on Linux build machine to the Windows machine. Open a new commandline window (in Windows) where the kernel image is copied and run the following commands.

```
cmd>c:\Xilinx\14.6\ISE_DS\settings64.bat
cmd>xmd
```

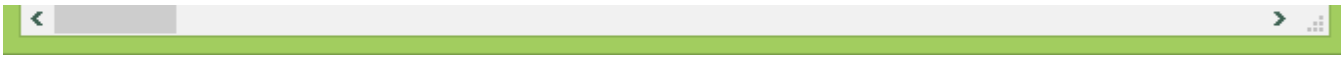
make sure to replace xx.xx with your ISE version and to choose settings32.bat if your system is 32 bit. These commands will start XMD within the commandline window. Now type the following command at the XMD command line to connect to the Microblaze processor running on the FPGA.

```
cmd>connect mb mdm
```

If everything went well, you should see XMD spit out some information as shown below.



```
C:\Users\user\Desktop\SaturnV3Linux\images>c:\Xilinx\14.6\ISE_DS\settings64.bat

C:\Users\user\Desktop\SaturnV3Linux\images>xmd
Xilinx Microprocessor Debugger (XMD) Engine
Xilinx EDK 14.6 Build EDK_P.68d
Copyright (c) 1995-2012 Xilinx, Inc.  All rights reserved.

XMD%
XMD% connect mb mdm

JTAG chain configuration
--------------------------------------------------
Device   ID Code        IR Length    Part Name
 1       34008093           6        XC6SLX45

MicroBlaze Processor Configuration :
-------------------------------------
Version............................8.50b
Optimization.......................Performance
Interconnect.......................AXI-LE
MMU Type...........................No_MMU
No of PC Breakpoints...............1
No of Read Addr/Data Watchpoints...0
No of Write Addr/Data Watchpoints..0
Instruction Cache Support..........on
Instruction Cache Base Address.....0xa4000000
Instruction Cache High Address.....0xa7ffffff
Data Cache Support.................on
Data Cache Base Address............0xa4000000
Data Cache High Address............0xa7ffffff
Exceptions  Support................off
FPU  Support.......................off
Hard Divider Support...............off
Hard Multiplier Support............on - (Mul32)
Barrel Shifter Support.............on
MSR clr/set Instruction Support....on
Compare Instruction Support........on
Data Cache Write-back Support......off
Fault Tolerance Support............off
Stack Protection Support...........off

Connected to "mb" target. id = 0
Starting GDB server for "mb" target (id = 0) at TCP port no 1234
XMD%
```

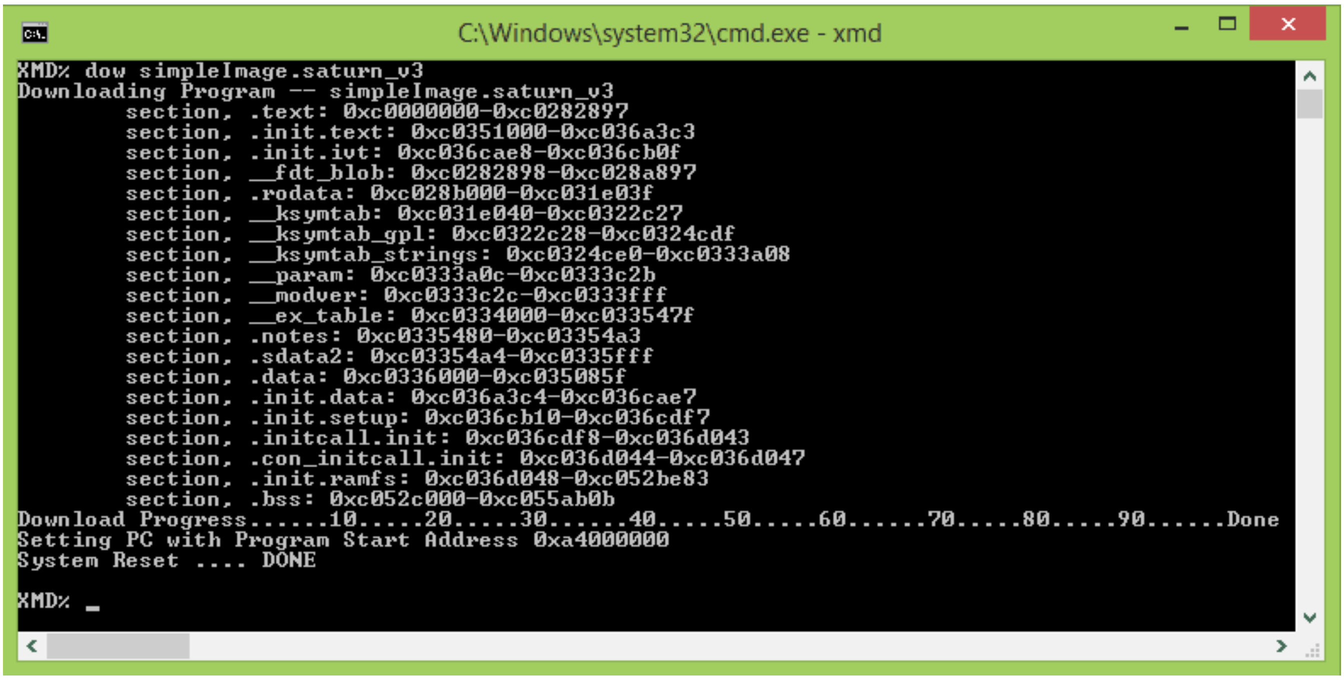


(https://numato.com/help/wp-content/uploads/2016/03/XMD_connect1.png)

Now we are connected to the Microblaze processor and we can download Linux kernel image. To download the image, run the below command at XMD command line.

```
cmd>dow simpleImage.saturn_v3
```

This command should start downloading the image to the DDR memory. Since simpleImage.saturn_v3 is in ELF format, it already contains the DDR address where it should be copied to. If download fails, chances are that you have not updated the Linux kernel base address with the starting address of the LPDDR memory assigned by EDK before building Linux kernel. We have discussed this in the section "Create buildroot configuration" of Part II (https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-ii/) of this article series. It can take a few minutes to download the kernel image to DDR. Below image shows the download process.



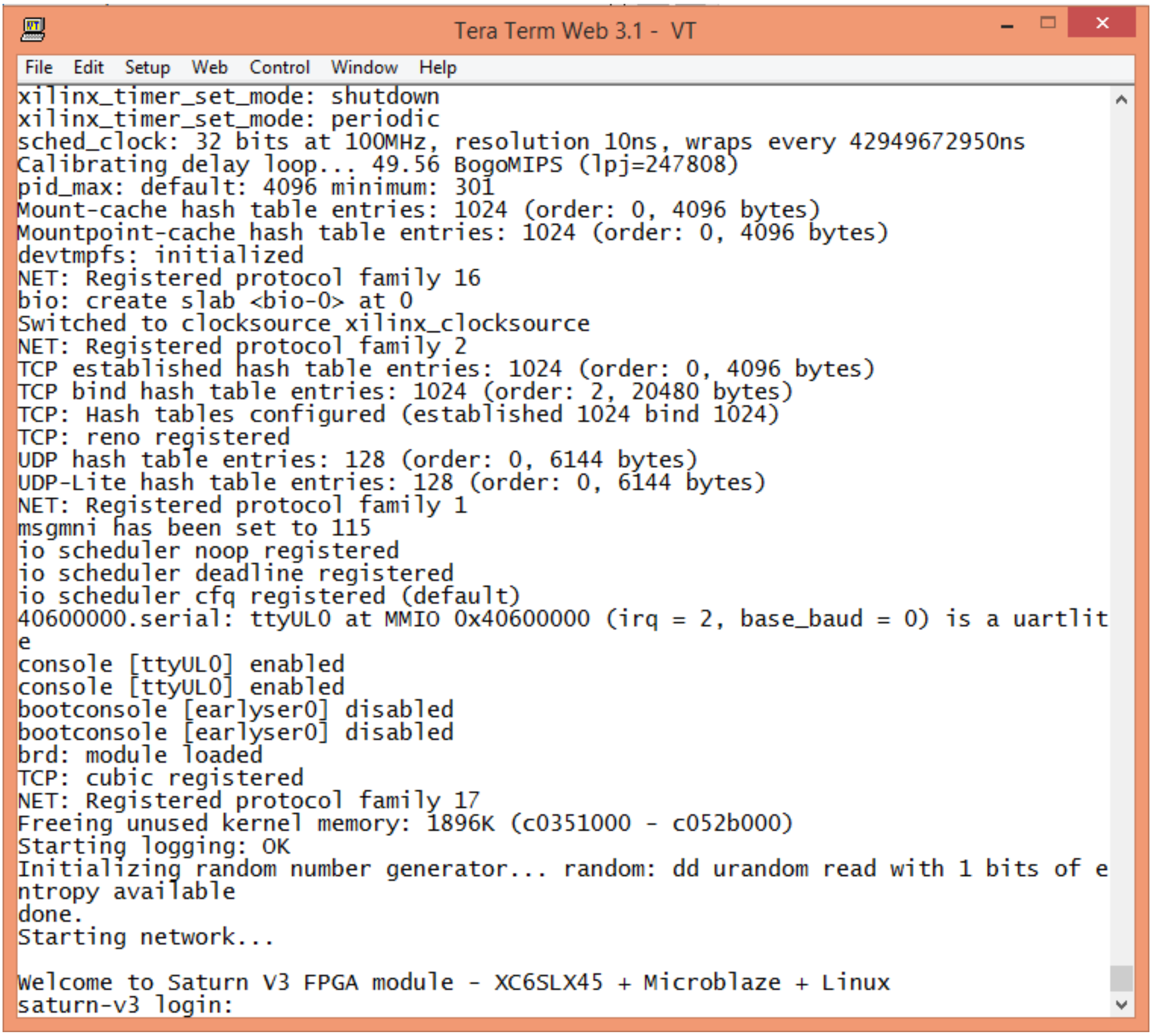
```
XMD% dow simpleImage.saturn_v3
Downloading Program -- simpleImage.saturn_v3
        section, .text: 0xc0000000-0xc0282897
        section, .init.text: 0xc0351000-0xc036a3c3
        section, .init.ivt: 0xc036cae8-0xc036cb0f
        section, __fdt_blob: 0xc0282898-0xc028a897
        section, .rodata: 0xc028b000-0xc031e03f
        section, __ksymtab: 0xc031e040-0xc0322c27
        section, __ksymtab_gpl: 0xc0322c28-0xc0324cdf
        section, __ksymtab_strings: 0xc0324ce0-0xc0333a08
        section, __param: 0xc0333a0c-0xc0333c2b
        section, __modver: 0xc0333c2c-0xc0333fff
        section, __ex_table: 0xc0334000-0xc033547f
        section, .notes: 0xc0335480-0xc03354a3
        section, .sdata2: 0xc03354a4-0xc0335fff
        section, .data: 0xc0336000-0xc035085f
        section, .init.data: 0xc036a3c4-0xc036cae7
        section, .init.setup: 0xc036cb10-0xc036cdf7
        section, .initcall.init: 0xc036cdf8-0xc036d043
        section, .con_initcall.init: 0xc036d044-0xc036d047
        section, .init.ramfs: 0xc036d048-0xc052be83
        section, .bss: 0xc052c000-0xc055ab0b
Download Progress......10.....20.....30......40.....50.....60......70.....80.....90......Done
Setting PC with Program Start Address 0xa4000000
System Reset .... DONE

XMD% _
```

(https://numato.com/help/wp-content/uploads/2016/03/XMD_download_Kernel_Image1.png)

Now we are ready to boot Linux. Open your favorite serial terminal emulation software (hyperterminal, PUTTY, Teraterm etcc..) and open the serial port corresponds to Saturn's FT2232H channel B. Set baud rate to 115200 and set any handshacking to off/none. Once the terminal software is setup, run the following command at XMD commandline to boot the kernel.

```
cmd>con 0xa4000000
```

Make sure to replace 0xa4000000 with the DDR base address/Kernel base address of your system if it is different. If all steps so far went as planned, you should see Linux booting in the serial terminal software as shown below.



```
xilinx_timer_set_mode: shutdown
xilinx_timer_set_mode: periodic
sched_clock: 32 bits at 100MHz, resolution 10ns, wraps every 42949672950ns
Calibrating delay loop... 49.56 BogoMIPS (lpj=247808)
pid_max: default: 4096 minimum: 301
Mount-cache hash table entries: 1024 (order: 0, 4096 bytes)
Mountpoint-cache hash table entries: 1024 (order: 0, 4096 bytes)
devtmpfs: initialized
NET: Registered protocol family 16
bio: create slab <bio-0> at 0
Switched to clocksource xilinx_clocksource
NET: Registered protocol family 2
TCP established hash table entries: 1024 (order: 0, 4096 bytes)
TCP bind hash table entries: 1024 (order: 2, 20480 bytes)
TCP: Hash tables configured (established 1024 bind 1024)
TCP: reno registered
UDP hash table entries: 128 (order: 0, 6144 bytes)
UDP-Lite hash table entries: 128 (order: 0, 6144 bytes)
NET: Registered protocol family 1
msgmni has been set to 115
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
40600000.serial: ttyUL0 at MMIO 0x40600000 (irq = 2, base_baud = 0) is a uartlit
e
console [ttyUL0] enabled
console [ttyUL0] enabled
bootconsole [earlyser0] disabled
bootconsole [earlyser0] disabled
brd: module loaded
TCP: cubic registered
NET: Registered protocol family 17
Freeing unused kernel memory: 1896K (c0351000 - c052b000)
Starting logging: OK
Initializing random number generator... random: dd urandom read with 1 bits of e
ntropy available
done.
Starting network...

Welcome to Saturn V3 FPGA module - XC6SLX45 + Microblaze + Linux
saturn-v3 login:
```

(https://numato.com/help/wp-content/uploads/2016/03/SaturnSpartan6_Booting_Linux3.png)

The default user is root and no password set for root user. Please note that since the files system is initramfs which is sitting in LPDDR, any changes that you make will not persist.

Congratulations if you followed the steps in this article series and was able to boot Linux successfully. Those who could not successfully boot, please don't give up and keep trying. You should be able to tackle this without much difficulties. If you have any questions, feel free to ask below in the comments section.

Since Linux kernel image is directly downloaded to DDR memory, you will loose everything when power cycling the board. To make the board boot in to Linux every time you power up the board, we need to put the FPGA bit file and kernel image in the SPI flash. Saturn has 128Mbit (16 Mbyte) SPI flash memory onboard. This memory is large enough to hold the FPGA bit file and Kernel image (You will still have some space left for custom data). Visit Part IV (https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-iv/) of this article series to see how this is done.

Read Part IV of this article series here. (https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-iv/)

(https://twitter.com/home?status=https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-iii/) f (https://www.facebook.com/sharer/sharer.php?u=https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-iii/) (https://pinterest.com/pin/create/button/?url=https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-iii/&media=&description=) G+ (https://plus.google.com/share?url=https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-iii/) in (https://www.linkedin.com/shareArticle?mini=true&url=https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-iii/)

Was this helpful?

👍 4 Yes

👎 No

3 Comments

Mehdi says:

REPLY (HTTPS://NUMATO.COM/KB/SATURN-MICROBLAZE-AND-LINUX-HOW-RUN-LINUX-SATURN-SPARTAN-6-FPGA-MODULE-PART-III/?REPLYTOCOM=86#RESPOND)

Is it possible to boot (any configuration of) linux on MicroBlaze without LPDDR RAM(Mimas Spartan6)?

November 3, 2017 at 12:11 pm

Rohit Singh says:

REPLY (HTTPS://NUMATO.COM/KB/SATURN-MICROBLAZE-AND-LINUX-HOW-RUN-LINUX-SATURN-SPARTAN-6-FPGA-MODULE-PART-III/?REPLYTOCOM=87#RESPOND)

No. You need substantial amount of RAM which FPGA’s on-die RAM cannot provide.

November 6, 2017 at 6:26 am

David Flores says:

REPLY (HTTPS://NUMATO.COM/KB/SATURN-MICROBLAZE-AND-LINUX-HOW-RUN-LINUX-SATURN-SPARTAN-6-FPGA-MODULE-PART-III/?REPLYTOCOM=246#RESPOND)

links of parts II and IV are wrong—

must be:
https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-ii/ (https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-ii/)
https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-iv/ (https://numato.com/kb/saturn-microblaze-and-linux-how-run-linux-saturn-spartan-6-fpga-module-part-iv/)

November 8, 2018 at 4:45 am

Leave A Comment

Comment

Name *

Email *

Website

POST COMMENT

Knowledge Base Category

Callisto Kintex 7 USB 3.1 FPGA Module (https://numato.com/kb-category/callisto-kintex-7-usb-3-1-fpga-module/) (2)

Embedded Linux (https://numato.com/kb-category/embedded-linux/) (12)

FPGA Applications (https://numato.com/kb-category/fpga-applications/) (13)

Galatea PCI Express Spartan 6 FPGA Module (https://numato.com/kb-category/galatea-pci-express-spartan-6-fpga-module/) (4)

Getting Started With FPGA (https://numato.com/kb-category/getting-started-with-fpga/) (32)

Intercore SDK Framework (https://numato.com/kb-category/intercore-sdk-framework/) (1)

Mimas A7 Mini FPGA Development Board (https://numato.com/kb-category/mimas-a7-mini-fpga-development-board/) (3)

Mimas Artix 7 FPGA Development Board (https://numato.com/kb-category/mimas-artix-7-fpga-development-board/) (10)

Narvi Spartan 7 FPGA Module (https://numato.com/kb-category/narvi-spartan-7-fpga-module/) (2)

Nereid Kintex 7 PCI Express FPGA Board (https://numato.com/kb-category/nereid-kintex-7-pci-express-fpga-board/) (3)

Neso Artix 7 FPGA Module (https://numato.com/kb-category/neso-artix-7-fpga-module/) (10)

Opsis: FPGA-based open video platform (https://numato.com/kb-category/opsis-fpga-based-open-video-platform/) (4)

Prodigy Series Automation Devices (https://numato.com/kb-category/prodigy-series-automation-devices/) (1)

Proteus Kintex 7 FPGA Development Module (https://numato.com/kb-category/proteus-kintex-7-fpga-development-module/) (1)

Quick Start Guides (https://numato.com/kb-category/quick-start-guides/) (2)

Rhea Device Management Tool (https://numato.com/kb-category/rhea-device-management-tool/) (2)

Saturn Spartan 6 FPGA Module (https://numato.com/kb-category/saturn-spartan-6-fpga-module/) (7)

Skoll Kintex 7 FPGA Module (https://numato.com/kb-category/skoll-kintex-7-fpga-module/) (6)

Styx Xilinx Zynq FPGA Module (https://numato.com/kb-category/styx-xilinx-zynq-fpga-module/) (8)

Tagus – Artix 7 PCI Express Development Board (https://numato.com/kb-category/tagus-artix-7-pci-express-development-board/) (1)

- Telesto MAX 10 FPGA Module (https://numato.com/kb-category/telesto-max-10-fpga-module/) (5)
- Tenagra FPGA System Management Software (https://numato.com/kb-category/tenagra-fpga-system-management-software/) (3)
- Theia Android Application (https://numato.com/kb-category/theia-android-application/) (1)
- USB GPIO Modules (https://numato.com/kb-category/usb-gpio-modules/) (2)
- USB Relay Modules (https://numato.com/kb-category/usb-relay-modules/) (1)
- Vivado Design Suit (https://numato.com/kb-category/vivado-design-suit/) (4)
- Waxwing Spartan 6 FPGA Development Board (https://numato.com/kb-category/waxwing-spartan-6-fpga-development-board/) (3)
- White Papers (https://numato.com/kb-category/white-papers/) (1)
- Working With Xilinx EDK (https://numato.com/kb-category/working-with-xilinx-edk/) (1)
- XO-Bus Framework (https://numato.com/kb-category/xo-bus/) (2)

Don't miss out on new articles! Subscribe to get valuable insights.

email address

Subscribe

(https://twitter.com/numatolab)

(https://www.facebook.com/numato/)

(https://www.youtube.com/user/NumatoLab)

(https://plus.google.com/+Numatosystems)

Privacy Policy Terms of Use