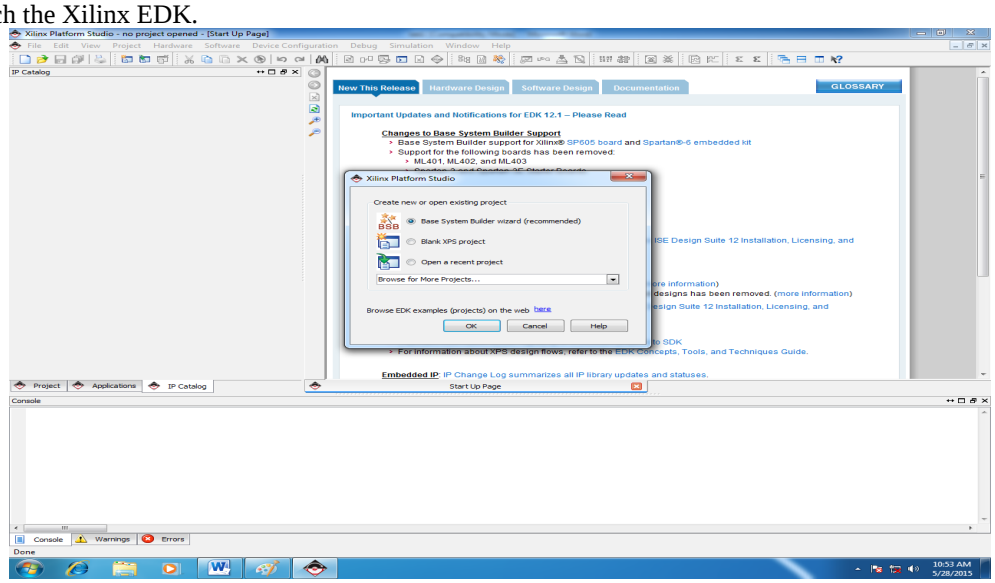
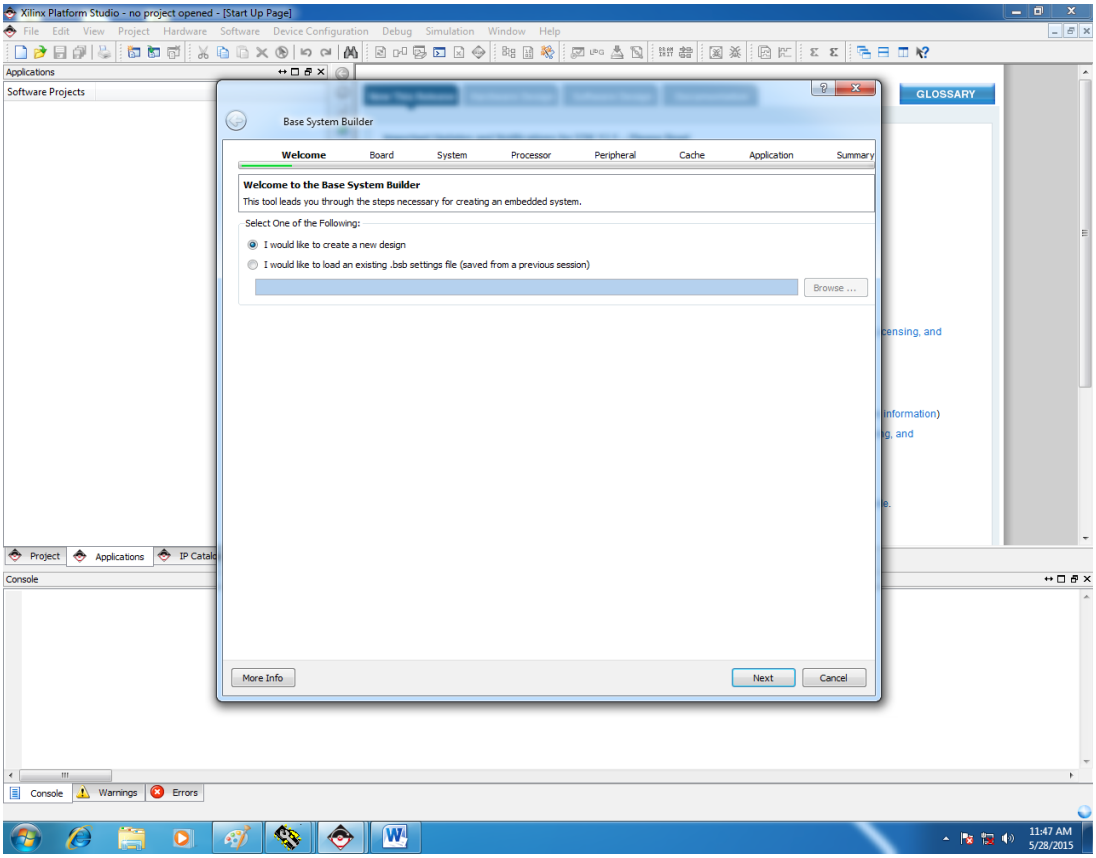
RIPHAH INTERNATIONAL UNIVERSITY

Lab 16
MPSoC: Multi Processor System on Chip Design

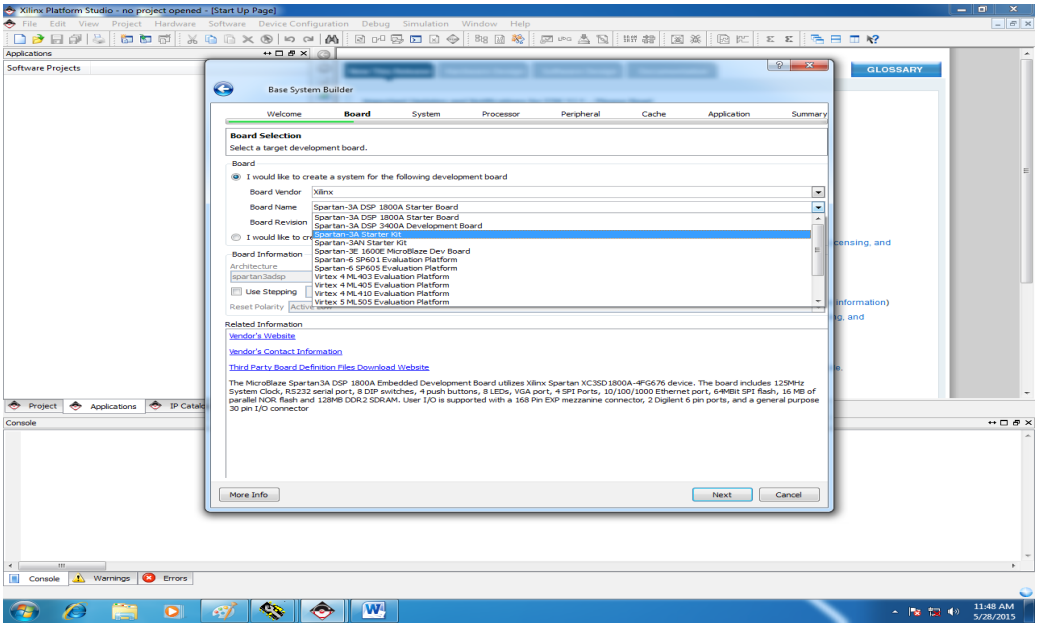
In Lab13, we implemented a uniprocessor system in order to have an overview of how to use the Xilinx EDK tool. Lab13 can be used as base system for the controller of your project (e.g. soccer project group). In this lab, we will develop a dual core system with two microblaze processors. This lab will be particularly useful for the development of the server.

Launch the Xilinx EDK.

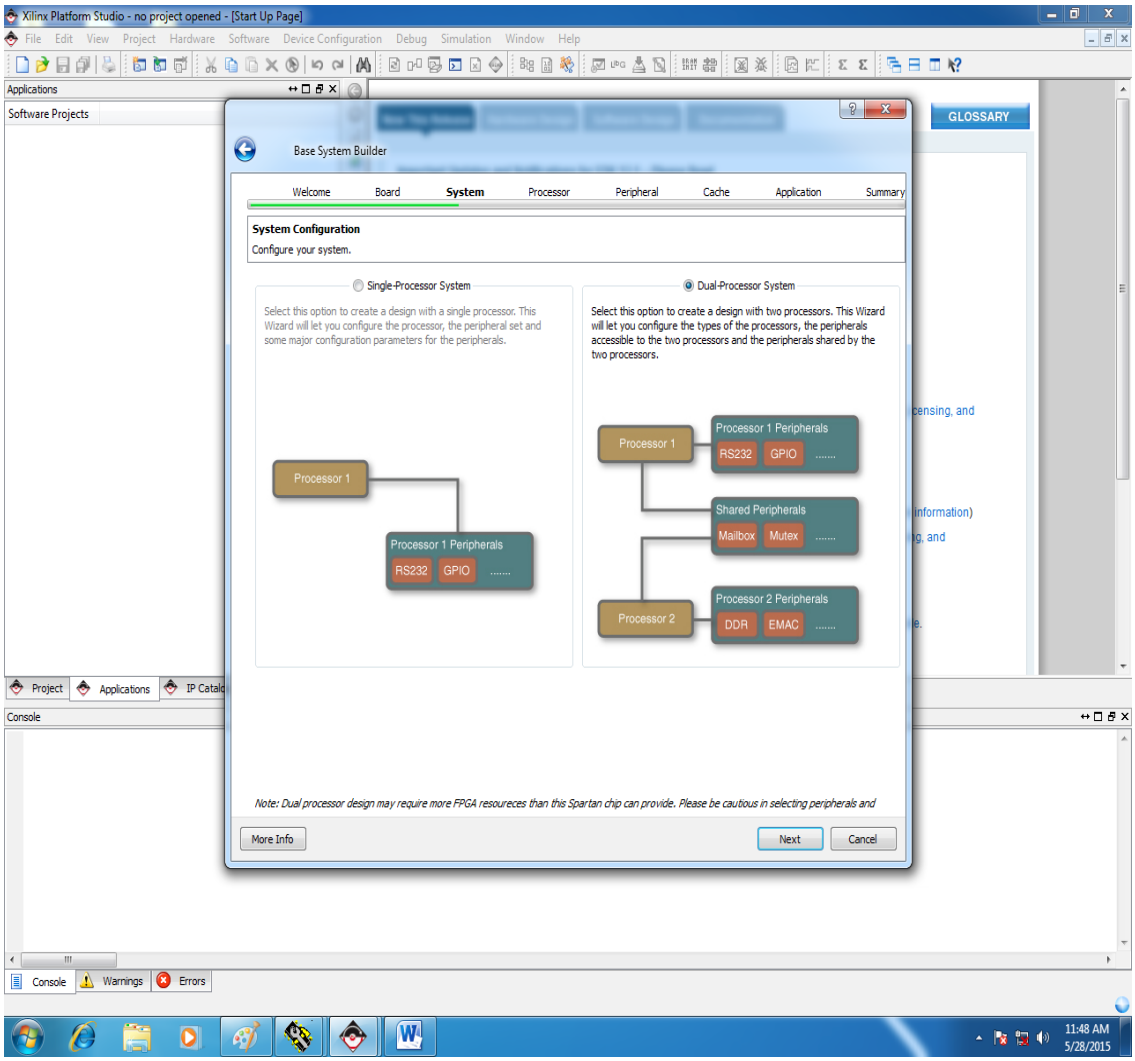




Select FGPA Board

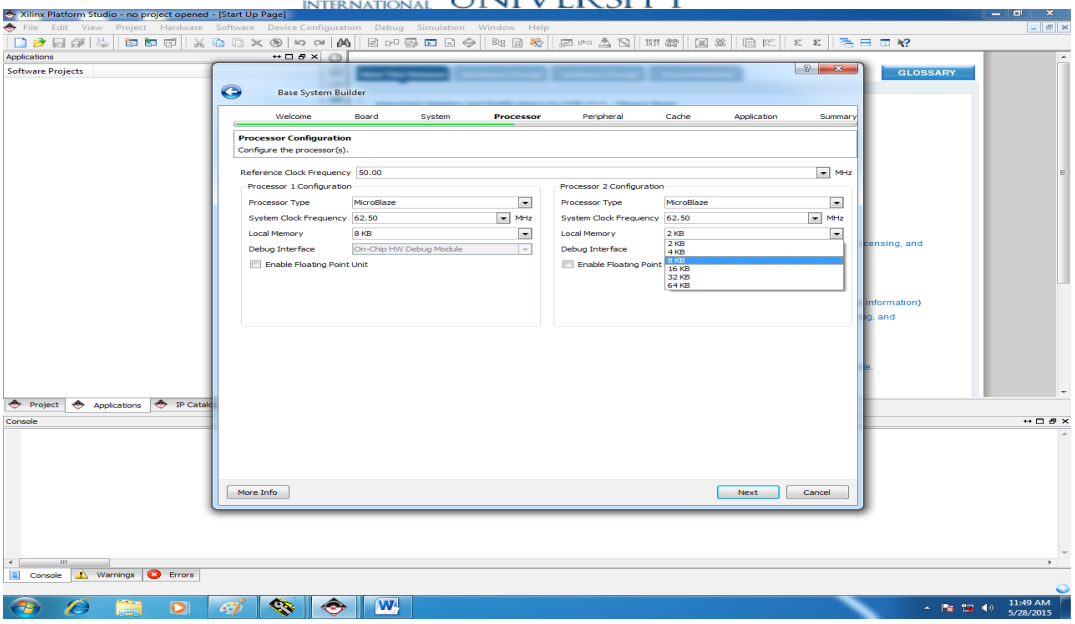


Select Dual Core System

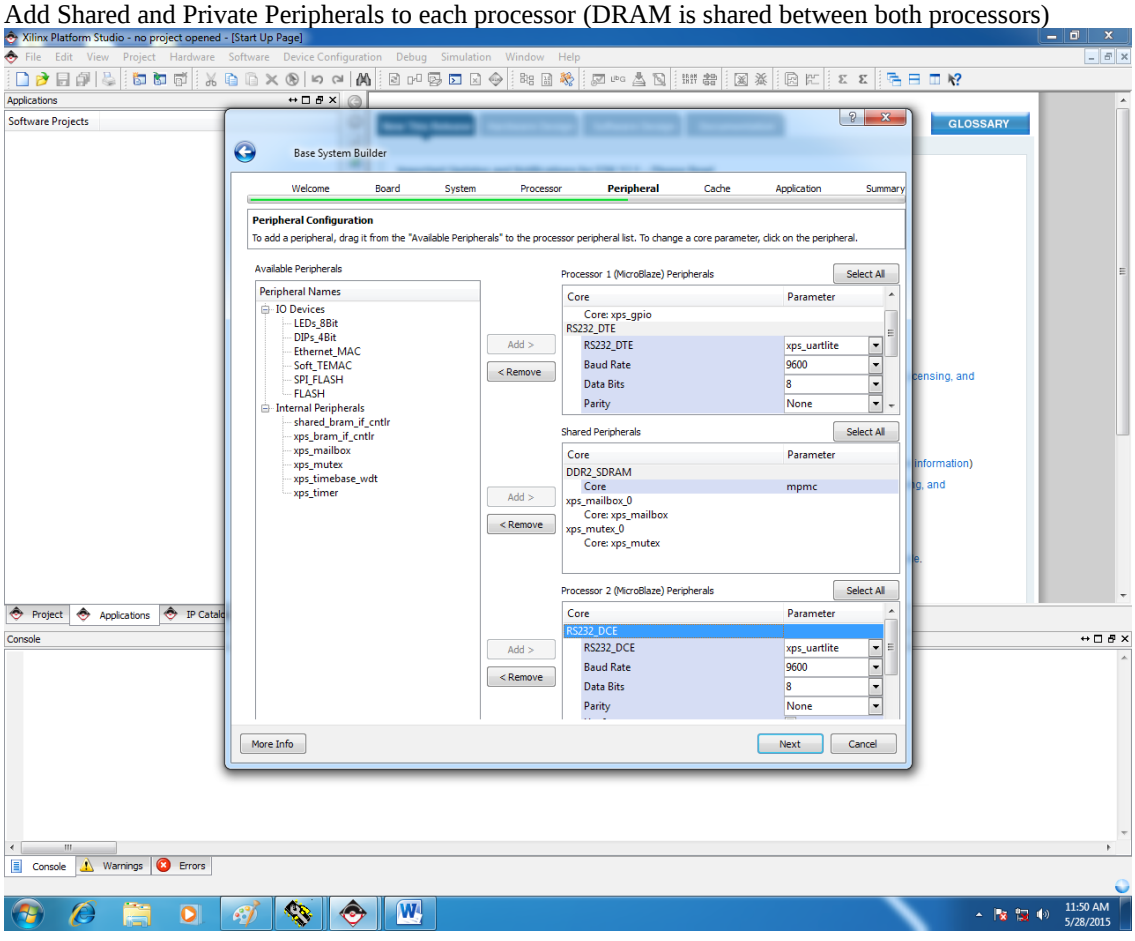


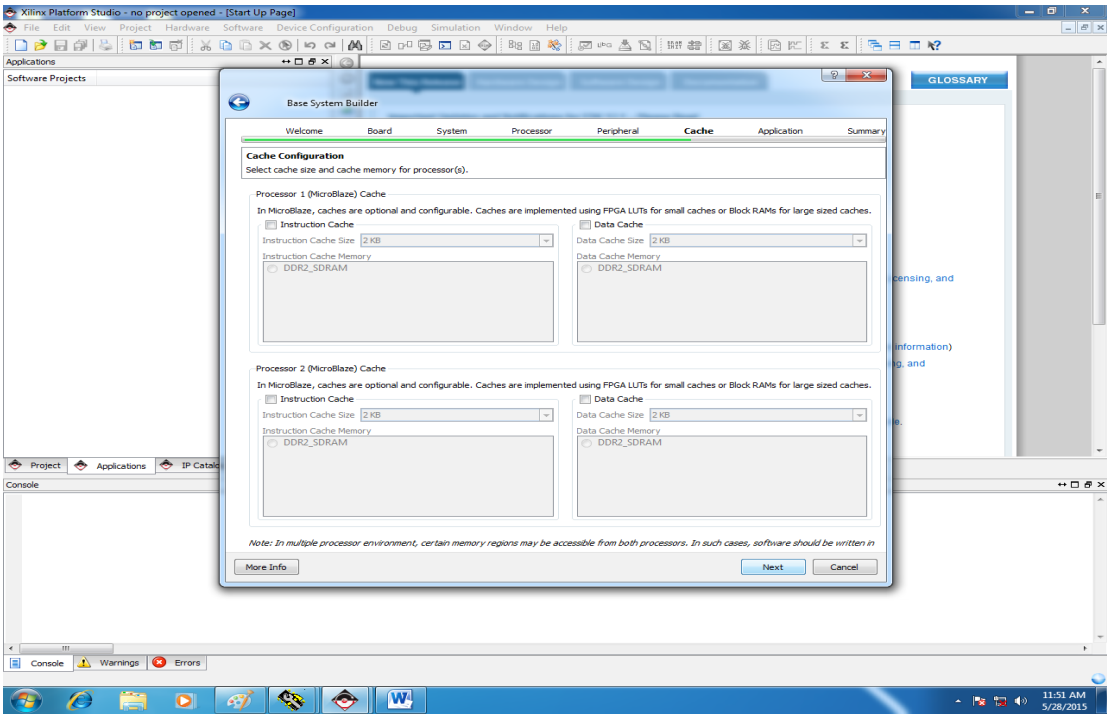
Add the interprocessor communication cores (*Mutex* & *Mailbox*)

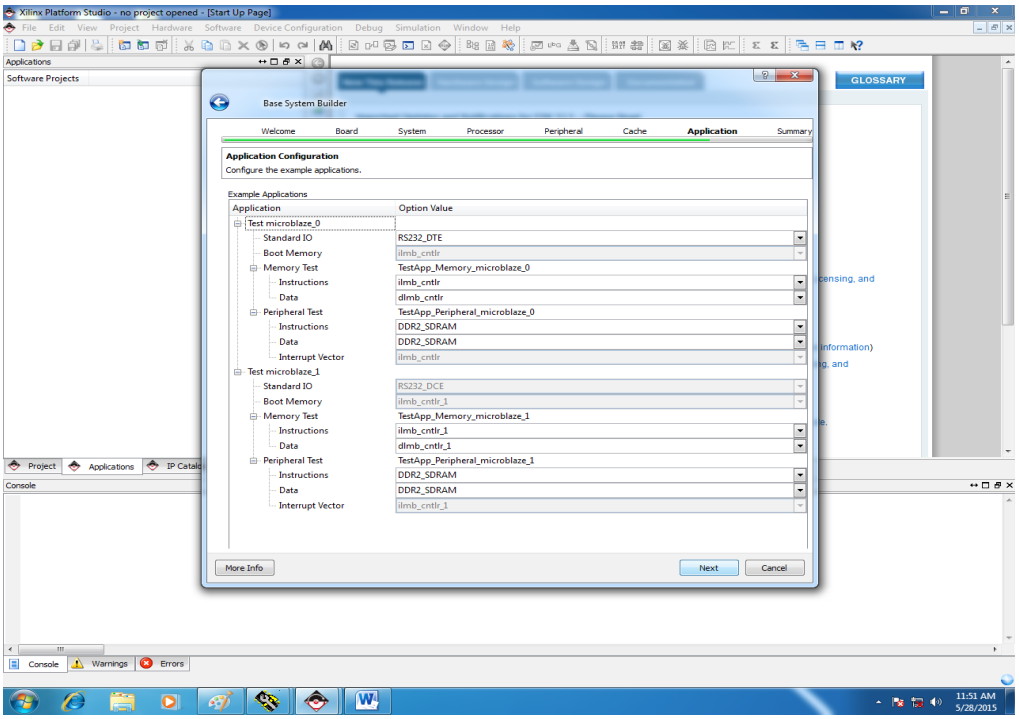
The Mutex core is used in the system to provide synchronization when accessing shared resources. The core has a configurable number of mutexes and has a write to lock scheme. The Mailbox core is used to pass messages between processor cores in FIFO fashion. The mailbox core offers an interrupt line if a core wants to indicate the presence of data.

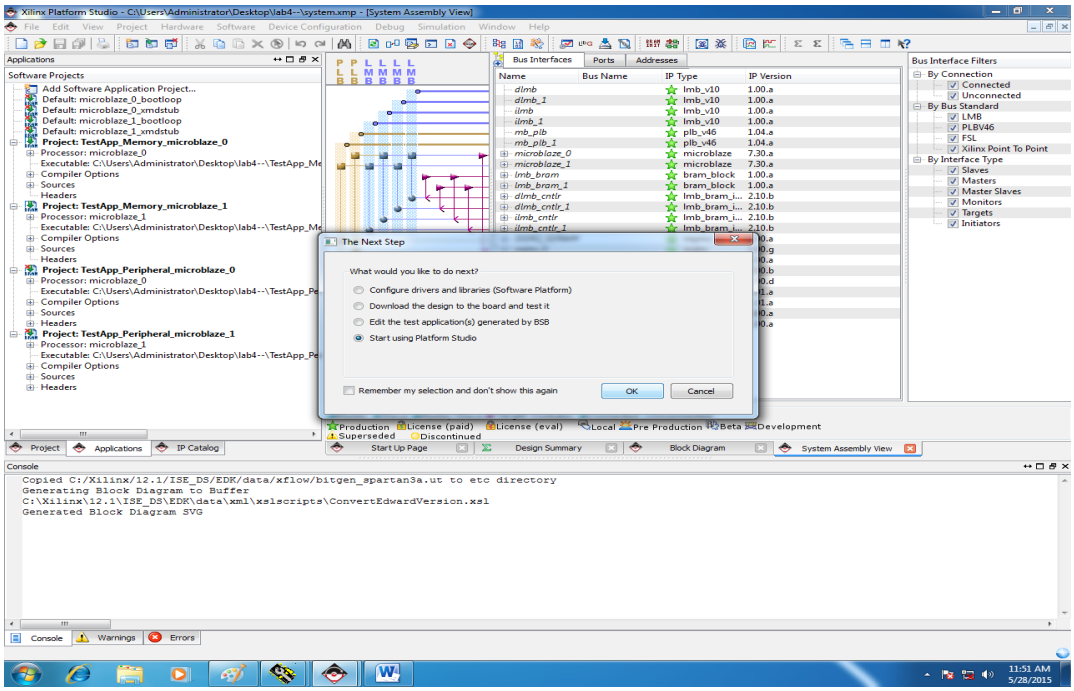


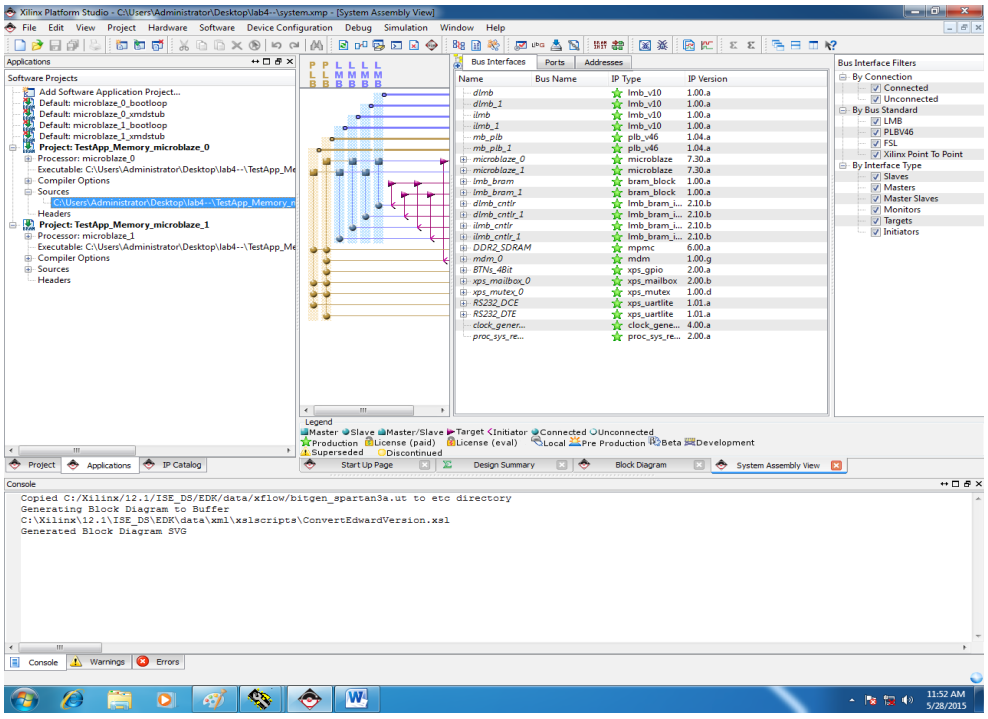
Add Shared and Private Peripherals to each processor (DRAM is shared between both processors)



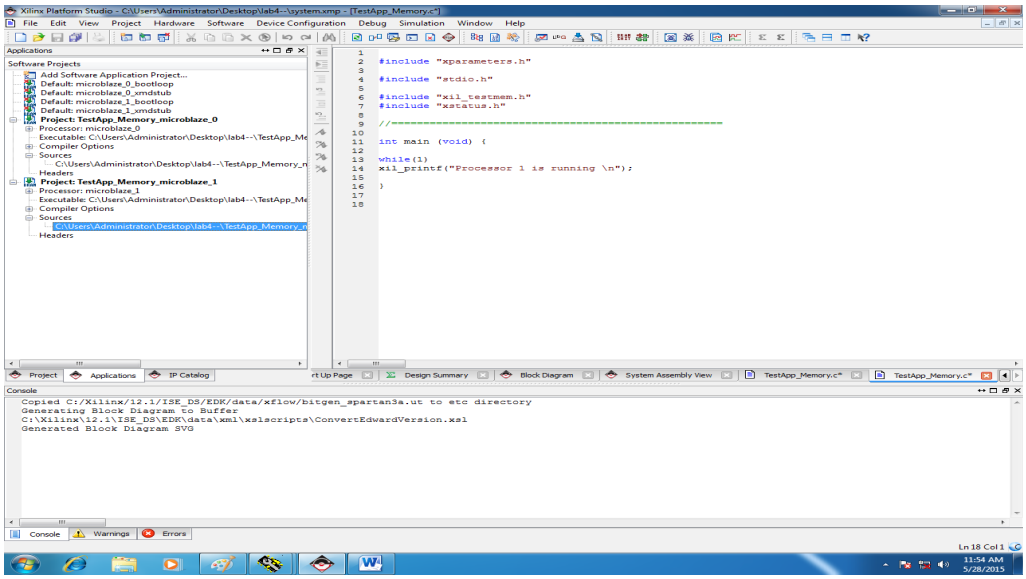

Xilinx Platform Studio - no project opened - [Start Up Page]
Base System Builder
Welcome
Board
System
Processor
Peripheral
Cache
Application
Summary
Cache Configuration
Select cache size and cache memory for processor(s).
Processor 1 (MicroBlaze) Cache
In MicroBlaze, caches are optional and configurable. Caches are implemented using FPGA LUTs for small caches or Block RAMs for large sized caches.
Instruction Cache
Data Cache
Instruction Cache Size
Data Cache Size
2 KB
Instruction Cache Memory
Data Cache Memory
DDR2_SDRAM
Processor 2 (MicroBlaze) Cache
In MicroBlaze, caches are optional and configurable. Caches are implemented using FPGA LUTs for small caches or Block RAMs for large sized caches.
Instruction Cache
Data Cache
Instruction Cache Size
Data Cache Size
2 KB
Instruction Cache Memory
Data Cache Memory
DDR2_SDRAM
Note: In multiple processor environment, certain memory regions may be accessible from both processors. In such cases, software should be written in
More Info
Next
Cancel
GLOSSARY
Applications
Software Projects
Project
Applications
Console
Warnings
Errors
11:51 AM
5/28/2015


Xilinx Platform Studio - no project opened - [Start Up Page]
Base System Builder
Welcome
Board
System
Processor
Peripheral
Cache
Application
Summary
Application Configuration
Configure the example applications.
Example Applications
Application
Option Value
Test microblaze_0
Standard IO
RS232_DTE
Boot Memory
ilmb_cntlr
Memory Test
TestApp_Memory_microblaze_0
Instructions
ilmb_cntlr
Data
dlmb_cntlr
Peripheral Test
TestApp_Peripheral_microblaze_0
Instructions
DDR2_SDRAM
Data
DDR2_SDRAM
Interrupt Vector
ilmb_cntlr
Test microblaze_1
Standard IO
RS232_DCE
Boot Memory
ilmb_cntlr_1
Memory Test
TestApp_Memory_microblaze_1
Instructions
ilmb_cntlr_1
Data
dlmb_cntlr_1
Peripheral Test
TestApp_Peripheral_microblaze_1
Instructions
DDR2_SDRAM
Data
DDR2_SDRAM
Interrupt Vector
ilmb_cntlr_1
More Info
Next
Cancel
GLOSSARY
Applications
Software Projects
Project
Applications
Console
Warnings
Errors
11:51 AM
5/28/2015

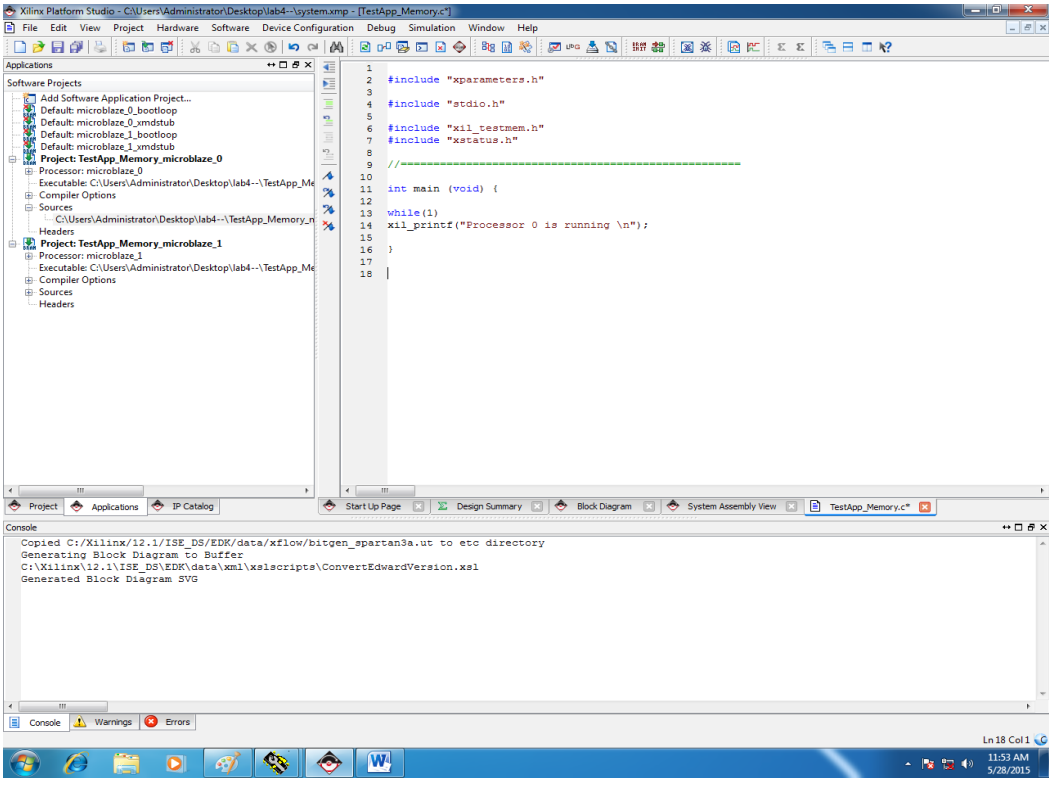

Xilinx Platform Studio - C:\Users\Administrator\Desktop\lab4--\system.xmp - [System Assembly View]
The Next Step
What would you like to do next?
Configure drivers and libraries (Software Platform)
Download the design to the board and test it
Edit the test application(s) generated by BSB
Start using Platform Studio
Remember my selection and don't show this again
OK
Cancel
Copied C:/Xilinx/12.1/ISE_DS/EDK/data/xflow/bitgen_spartan3a.ut to etc directory
Generating Block Diagram to Buffer
C:\Xilinx\12.1\ISE_DS\EDK\data\xml\xslscripts\ConvertEdwardVersion.xsl
Generated Block Diagram SVG


Xilinx Platform Studio - C:\Users\Administrator\Desktop\lab4--\system.xmp - [System Assembly View]
Copied C:/Xilinx/12.1/ISE_DS/EDK/data/xflow/bitgen_spartan3a.ut to etc directory
Generating Block Diagram to Buffer
C:\Xilinx\12.1\ISE_DS\EDK\data\xml\xslscripts\ConvertEdwardVersion.xsl
Generated Block Diagram SVG

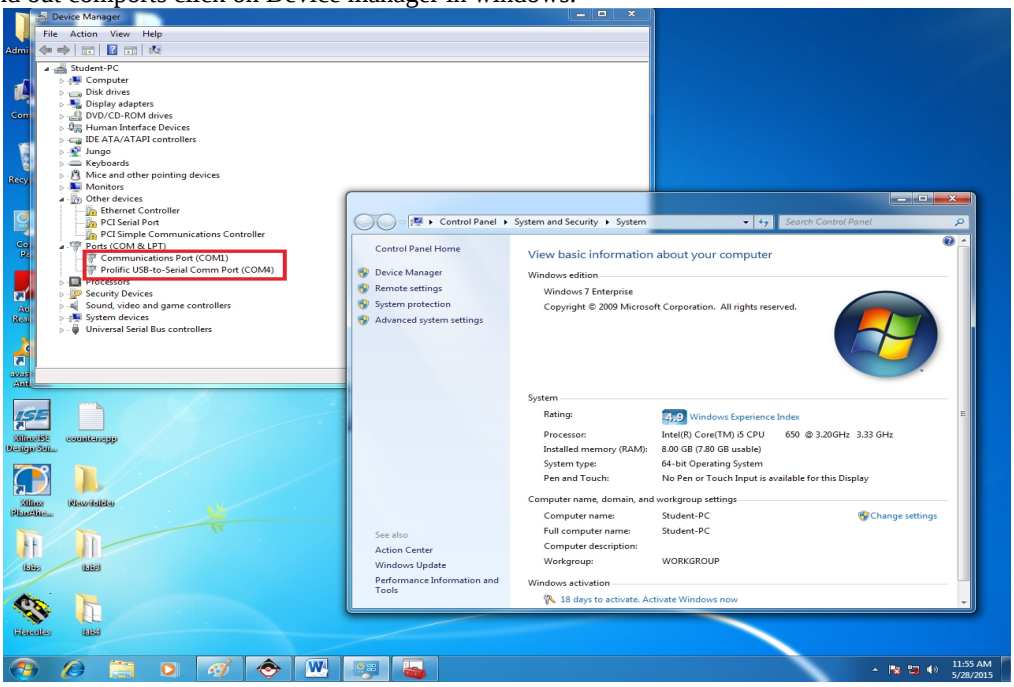
Write simple code for Processor 1



And same code but with different ID for processor two



To find out comports click on Device manager in windows.



Open Two instance of Serial Terminal each for single processor and open appropriate Comm Port.