

# LAB 1

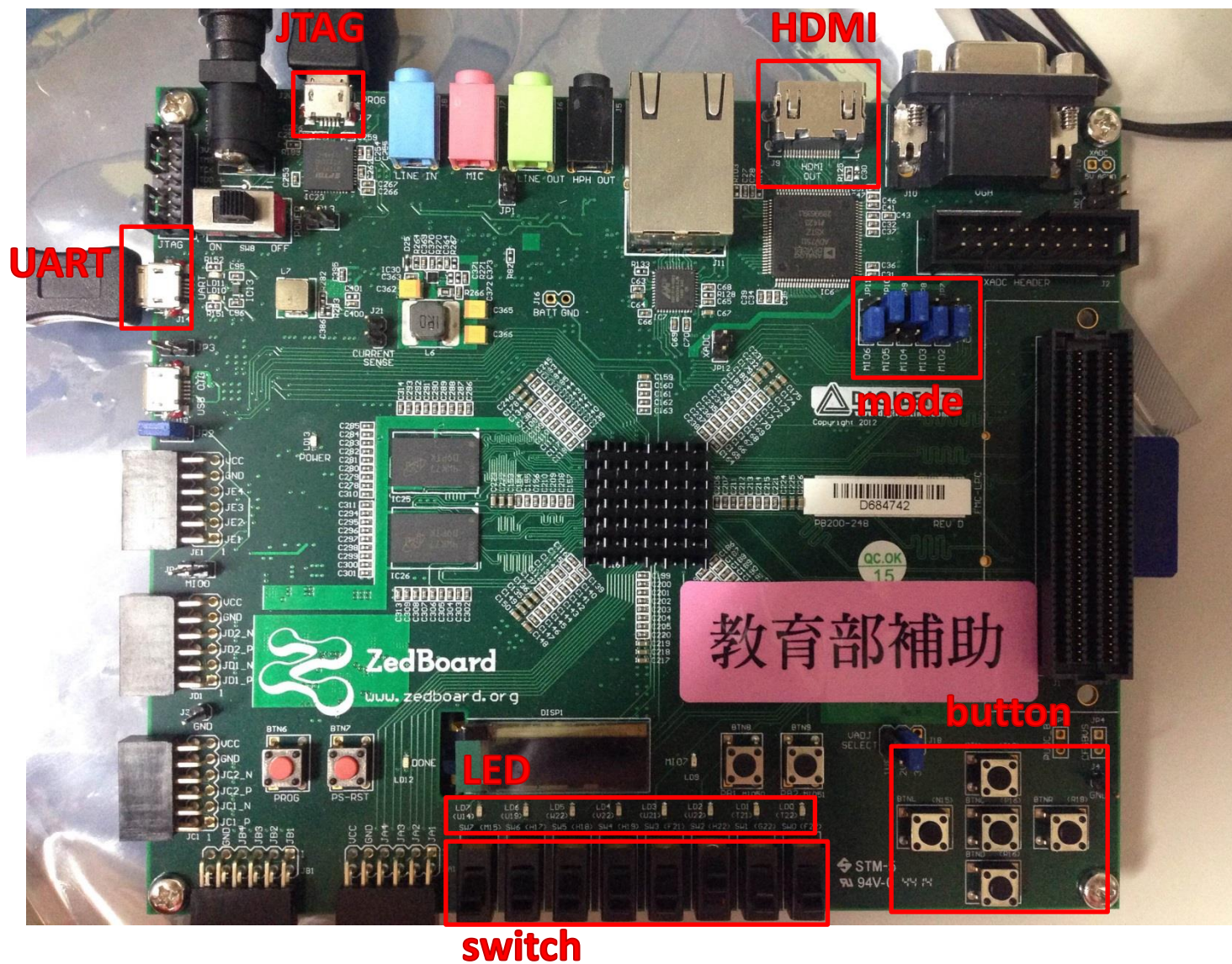
Platform, OS and basic I/O setup

# What would you get?

1. Zedboard
2. MicroUSB to USB cable (male to male)
3. OTG USB cable (male MicroUSB to female USB)
4. SD card
5. 12V power supply
6. Cat 5e lan cable
7. HDMI cable
8. HDMI to DVI adapter
9. MicroUSB to USB cable



# Zedboard



# Environment

- OS: Ubuntu 14.04 64-bit
- Vivado Design Suite
  - Vivado: for hardware design
  - Xilinx SDK: for software design
- Petalinux Tools
  - Linux kernel
- JTAG Driver on Ubuntu
  - <http://svenand.blogdrive.com/archive/172.html#.VqG0Evl9670>

# **LAB 1.1**

Platform setup and development environment setup

# Vivado Design Suite - Download

Vivado Design Tools

ISE Design Tools

Device Models

CAE Vendor Libraries

PetaLinux

2014.2

2014.1

2013.4

2013.3

2013.2


2013.1


2012.4


2012.3


2012.2

## Vivado Design Suite - 2013.4 Full Product Installation

 [All OS Vivado and SDK Full Installer \(TAR/GZIP - 6.81 GB\)](#)  
MD5 SUM Value: 16ad4c3fca376629e0b438e43a25d5ac

 [All OS Vivado Full Installer \(No SDK\) \(TAR/GZIP - 5.36 GB\)](#)  
MD5 SUM Value: 739bbb2c2e526a73c0b325cd54661138

 [Vivado Full Installer for Windows \(No SDK\) \(TAR/GZIP - 4.2 GB\)](#)  
MD5 SUM Value: 308ec43e26638b04f622b52bfca1f345

 [Vivado Full Installer for Linux \(No SDK\) \(TAR/GZIP - 4.22 GB\)](#)  
MD5 SUM Value: 4f359a307e945342fd3a938fd7916d08

### Important Information

Having trouble downloading? The download links above require the installation and use of a browser-based (plug-in) download manager. Your company's policy and/or firewall settings may not permit the download manager to be installed or operate properly. If you wish to bypass the use of the Xilinx download manager, please see [AR# 58485](#).

Download

Includes

Documentation Navigator

System Generator for DSP

Vivado Design Suite (All Editions)

Vivado High Level Synthesis (HLS)

Vivado WebPACK (Free)

Download Type

Full Product Installation

Last Updated

Dec 18, 2013

Documentation

[2013.4 - Release Notes](#)

[2013 - Known Issues](#)


Enablement


[License Solution Center](#)

Order DVD

[ISE Design Suite DVD](#)

## Multi-File Download: Vivado (No SDK) - 2013.4 Full Product Installation

 [All Platforms - Split Installer Base Image - File 1/4 \(ZIP - 1.24 GB\)](#)  
MD5 SUM Value: f043894c9a67415354aaad65d39a67bd

 [Install Data A - File 2/4 \(ZIP - 1.36 GB\)](#)  
MD5 SUM Value: d1dfac582aed4ed2247595306fec5299

Download

Includes

Documentation Navigator

System Generator for DSP

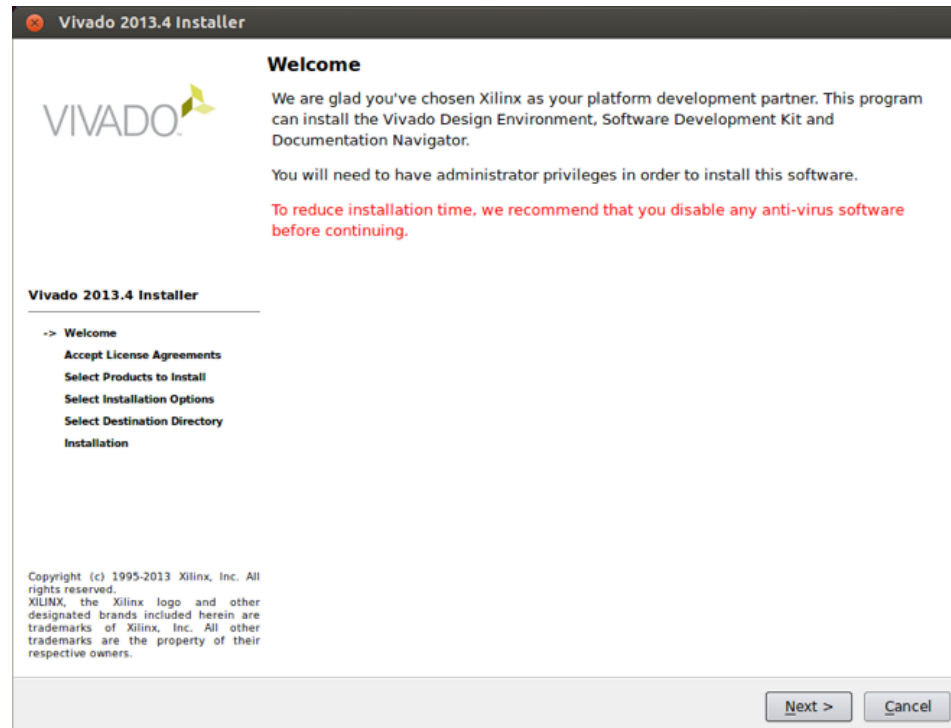
Vivado Design Suite (All Editions)

Vivado High Level Synthesis (HLS)



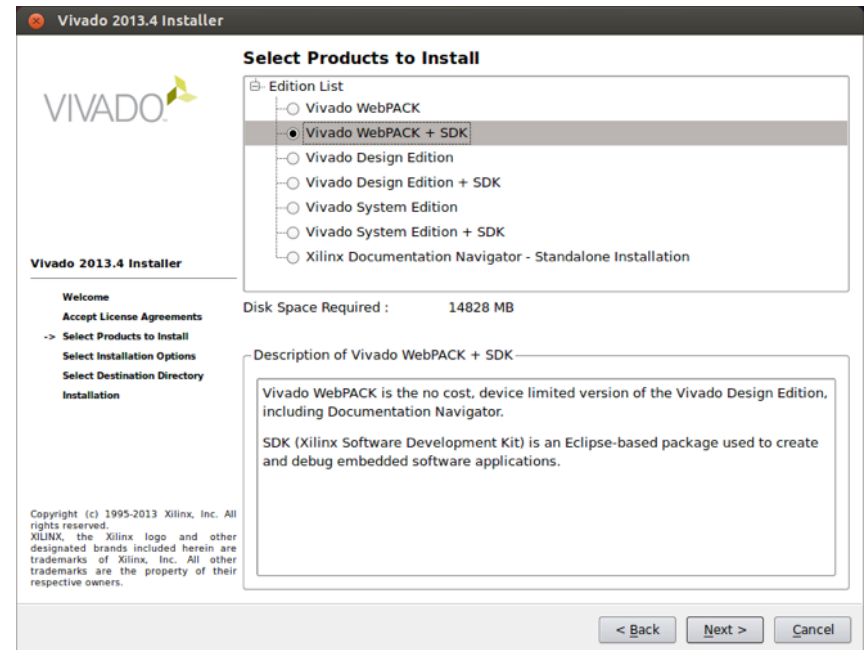
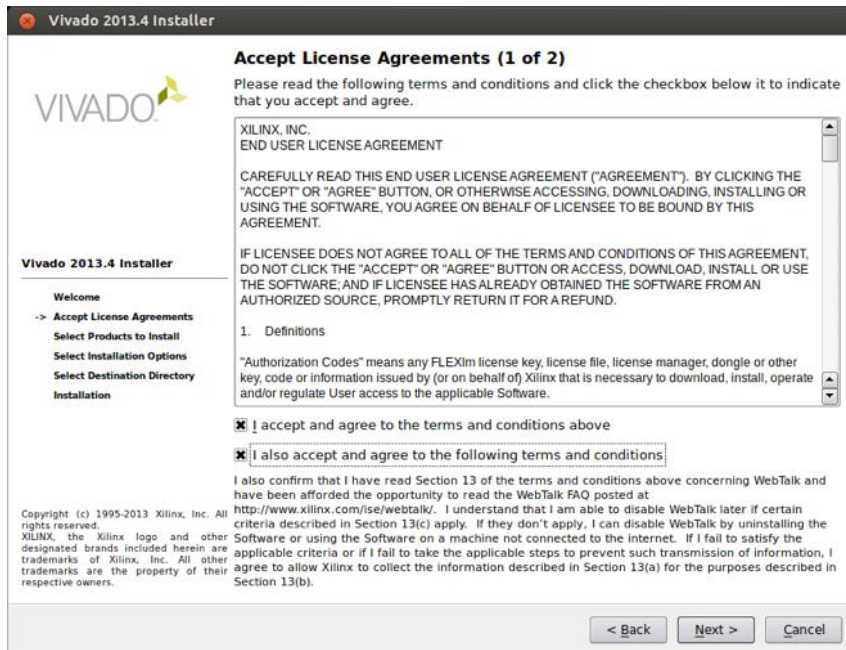
# Vivado Design Suite – Install

- Host> cd Downloads
- Host> tar xvf Xilinx\_Vivado\_SDK\_2013.4\_1210\_1.tar
- Host> sudo mkdir /opt/Xilinx
- Host> cd Xilinx\_Vivado\_SDK\_2013.4\_1210\_1
- Host> sudo ./xsetup



# Vivado Design Suite – Install

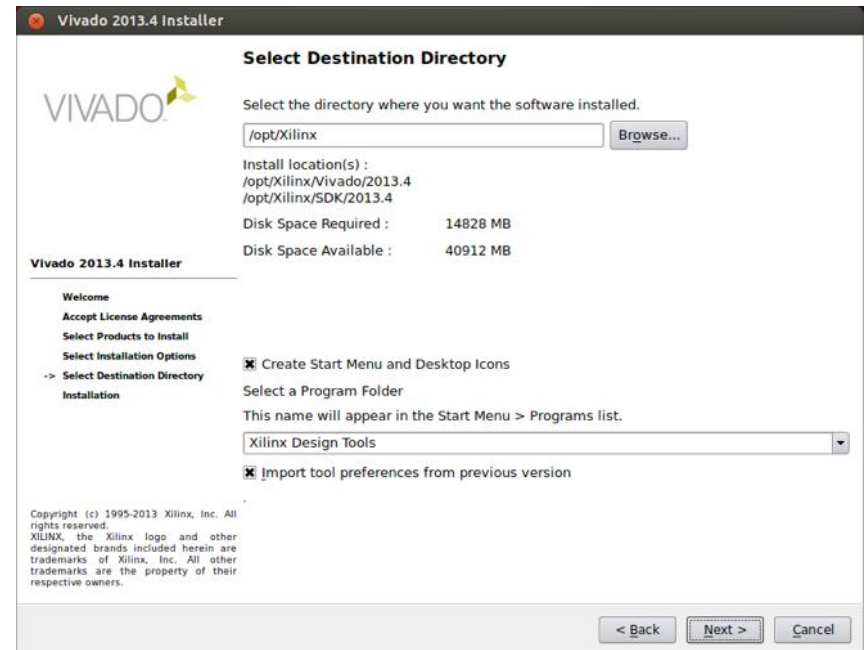
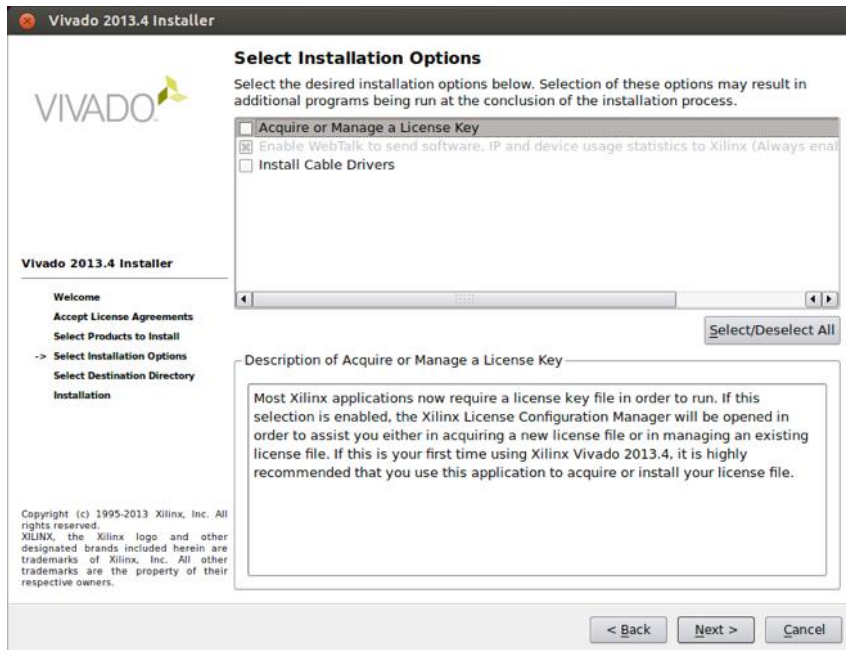
## Install WebPACK + SDK





# Vivado Design Suite – Install

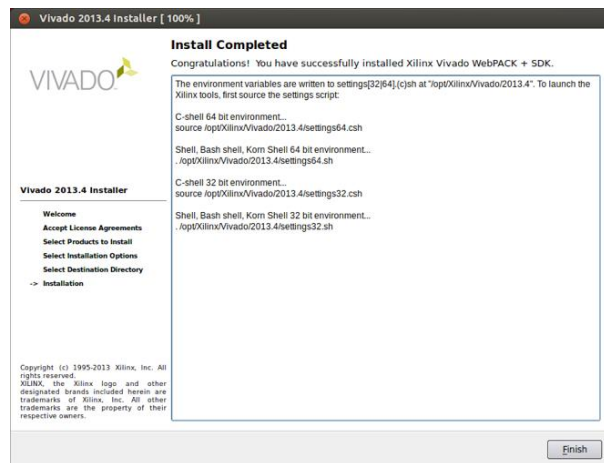
Don't choose two  
installation options



# Vivado Design Suite – Install



- Host> vi ~/.bashrc
  - Add
  - “source /opt/Xilinx/Vivado/2013.4/settings64.sh” (64bit Ubuntu)
  - or
  - “source /opt/Xilinx/Vivado/2013.4/settings32.sh” (32bit Ubuntu)



# Vivado Design Suite – Acquire License

- Host> hostname
- Host> ifconfig

```
Terminal
File Edit View Search Terminal Help
-->
-->hostname
svenand-VirtualBox
-->ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:fd:8f:
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe8f74/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:847395 errors:0 dropped:0 overruns:0 frame:0
          TX packets:190525 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:926686206 (926.6 MB)  TX bytes:13955487 (13.9 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2385 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2385 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:185191 (185.1 KB)  TX bytes:185191 (185.1 KB)

-->
-->
-->
```

# Vivado Design Suite – Acquire License


## Check PetaLinux Tools and ISE WebPACK



### Certificate Based Licenses



Product	Type	License	Available Seats	Status	Subscription End Date
<input checked="" type="checkbox"/> ISE WebPACK License	Certificate - No Charge	Node	1/1	Current	None
<input checked="" type="checkbox"/> PetaLinux Tools License	Certificate - Evaluation	Node	1/1	Current	365 days
<input type="checkbox"/> PetaLinux Tools License, Floating License	Certificate - Evaluation	Floating	1/1	Current	365 days
<input type="checkbox"/> Vivado HLS Evaluation License	Certificate - Evaluation	Node	1/1	Current	30 days



Input your hostname and MAC address

**Add a host...**

1. Host Name : svenand-VirtualBox

2. Operating System :  Linux 64-bit

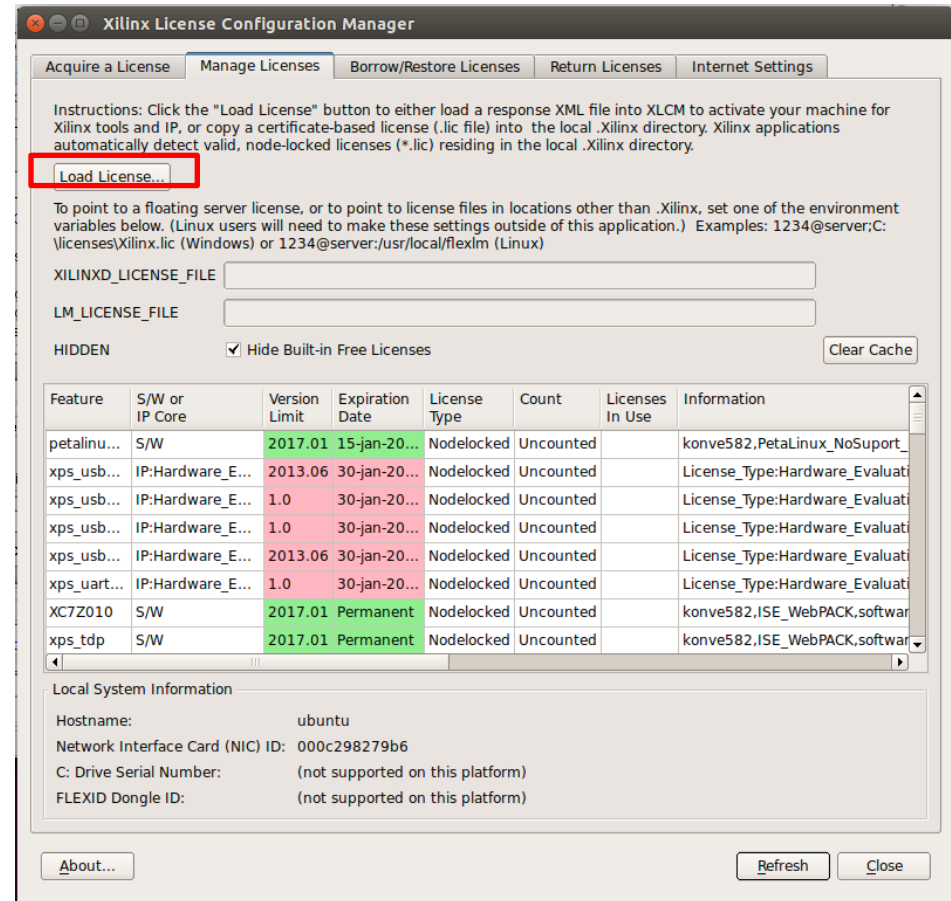
3. Host ID Type :  Ethernet MAC

4. Host ID Value : 080027fd8fxx 

Cancel Add

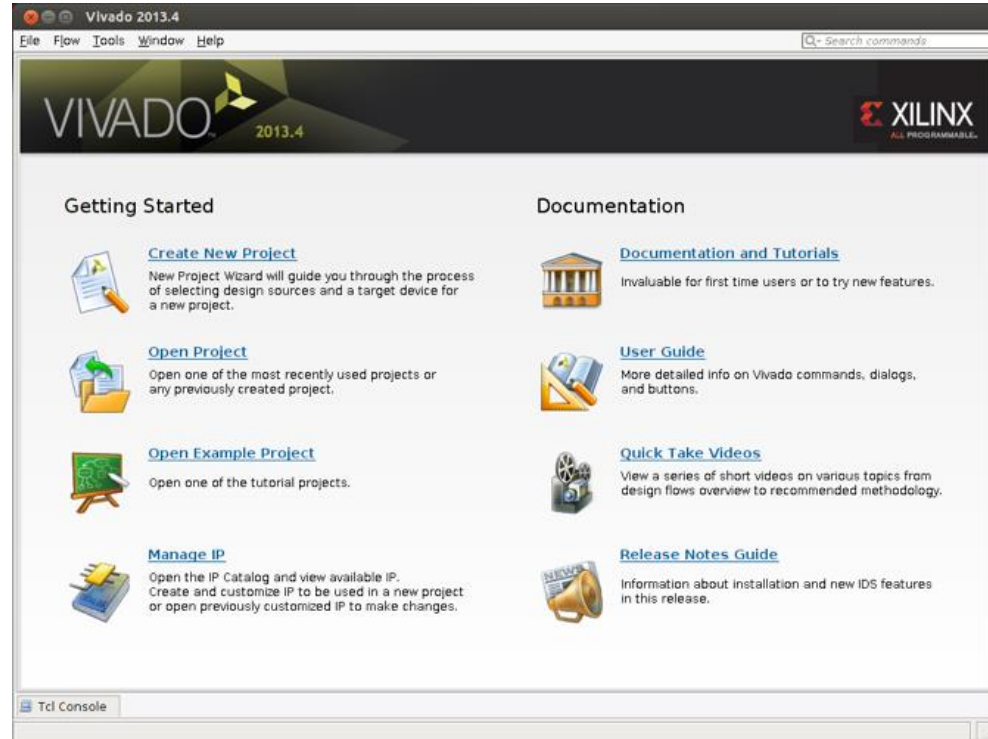
# Vivado Design Suite – Manage License

- Download the license from the e-mail
- Host > vivado &
- Help -> Manage License
- Click *Load License*



# Vivado Design Suite – Start Vivado and SDK

- Host> vivado &
- Host> xsdk &
- 'GLIBCXX\_3.4.9' not found:
  - Solution:  
<http://ubuntuforums.org/showthread.php?t=808045>





# PetaLinux Tools - Download

[Vivado Design Tools](#)[ISE Design Tools](#)[Device Models](#)[CAE Vendor Libraries](#)[PetaLinux](#)


## Version


[2014.2](#)[2013.10](#)[2013.04](#)[2012.12](#)

## PetaLinux - 2013.10 Installation Files


 [PetaLinux 2013.10 Installation archive for Zynq and MicroBlaze](#)  
(BSP - 754.68 MB)  
MD5 SUM Value: **b308f364d880a2bde99a72d9d500fbed**


 [PetaLinux 2013.10 Board Support Package for Xilinx ZC702 evaluation kit](#) (BSP - 49.09 MB)  
MD5 SUM Value: **9159340e9167da0c4b1bf32f70a20cd4**


 [PetaLinux 2013.10 Board Support Package for Avnet/Digilent ZedBoard](#) (BSP - 60.22 MB)  
MD5 SUM Value: **bd4ea658e06638a1db63cd1a0510e518**

 [PetaLinux 2013.10 Board Support Package for Xilinx ZC702 evaluation kit \(AMP reference design\)](#) (BSP - 49.64 MB)  
MD5 SUM Value: **76e0c1ea40506f63250c32cee6af5218**

 [PetaLinux 2013.10 Board Support Package for Xilinx ZC706 evaluation kit](#) (BSP - 60.32 MB)  
MD5 SUM Value: **2f34970f50065b973c446b55ebdd191a**

 [PetaLinux 2013.10 Board Support Package for Xilinx ZC706 evaluation kit \(AMP reference design\)](#) (BSP - 61 MB)  
MD5 SUM Value: **20ae8531c9fc2c8d9bfb626fe6360f81**

 [PetaLinux 2013.10 Board Support Package for Digilent Atlys board](#) (BSP - 21.66 MB)  
MD5 SUM Value: **9dbc0a5d18bd7ccfa7bb9ea23cadf1a6**

 [PetaLinux 2013.10 Board Support Package for Xilinx ML605 evaluation kit](#) (BSP - 22.65 MB)  
MD5 SUM Value: **9e095ed7775dab53a3db5a6a2b2f3599**

Download Type  
Installation Files

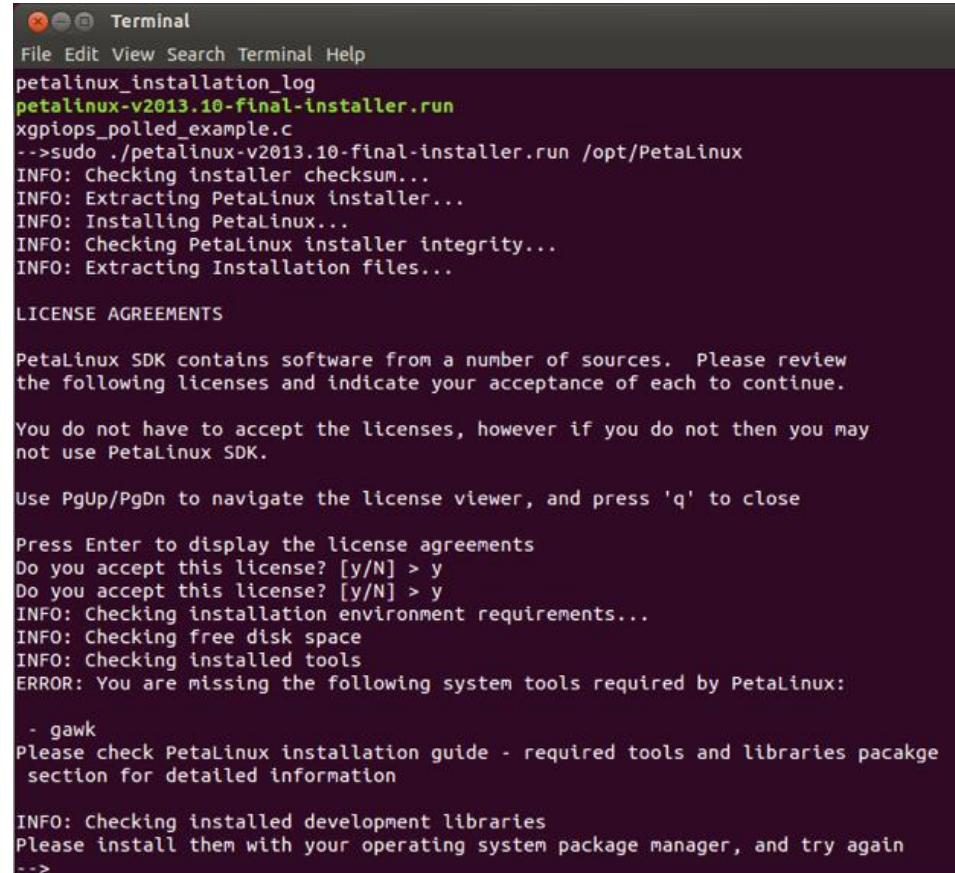
Last Updated  
Dec 2, 2013

Answers  
[Release Notes and Known Issues](#)

Documentation  
[PetaLinux SDK User Guide](#)

# PetaLinux Tools - Install

- Host> cd ~/Downloads
- Host> sudo ./petalinux-v2013.10-final-installer.run /opt/PetaLinux
- Host> vi ~/.bashrc
  - Add
  - “source
  - /opt/PetaLinux/petalinux-v2013.10-final/settings.sh”



```
Terminal
File Edit View Search Terminal Help
petalinux_installation_log
petalinux-v2013.10-final-installer.run
xgpiops_polled_example.c
-->sudo ./petalinux-v2013.10-final-installer.run /opt/PetaLinux
INFO: Checking installer checksum...
INFO: Extracting PetaLinux installer...
INFO: Installing PetaLinux...
INFO: Checking PetaLinux installer integrity...
INFO: Extracting Installation files...

LICENSE AGREEMENTS

PetaLinux SDK contains software from a number of sources. Please review
the following licenses and indicate your acceptance of each to continue.

You do not have to accept the licenses, however if you do not then you may
not use PetaLinux SDK.

Use PgUp/PgDn to navigate the license viewer, and press 'q' to close

Press Enter to display the license agreements
Do you accept this license? [y/N] > y
Do you accept this license? [y/N] > y
INFO: Checking installation environment requirements...
INFO: Checking free disk space
INFO: Checking installed tools
ERROR: You are missing the following system tools required by PetaLinux:

- gawk
Please check PetaLinux installation guide - required tools and libraries package
section for detailed information

INFO: Checking installed development libraries
Please install them with your operating system package manager, and try again
-->
```

# PetaLinux Tools – Required Package

- Host> sudo apt-get install gawk
- Host> sudo apt-get install bison
- Host> sudo apt-get install flex
- Host> sudo apt-get install zlib1g-dev
- Host> sudo apt-get install tofrodos
- Host> sudo apt-get install libstdc++6:i386
- Host> sudo apt-get install libncurses5:i386
- Host> sudo apt-get install libncurses5w-dev:i386
- Host> sudo apt-get install gcc
- Host> sudo ln -s /usr/bin/make /usr/bin/gmake

Tool/Library	YUM/RPM Package for RHEL/CentOS/Fedora	APT Package for Debian/Ubuntu	RPM Package for SuSE
dos2unix	dos2unix	tofrodos	dos2unix
ip	iproute	iproute	iproute2
gawk	gawk	gawk	gawk
gcc	gcc	gcc	gcc
git	git	git-core	git-core
make	gnutls-devel	make	make
netstat	net-tools	net-tools	net-tools
ncurses	ncurses-devel	ncurses-dev	ncurses-devel
tftp server	tftp-server	tftpd	tftp-server
zlib	zlib-devel	zlib1g-dev	zlib-devel
flex	flex	flex	flex
bison	bison	bison	bison
32bit libs	libstdc++-4.4.6-4.el6.i686 glibc.i686 libgcc.i686	ia32-libs lib32ncursesw5	32-bit runtime environment

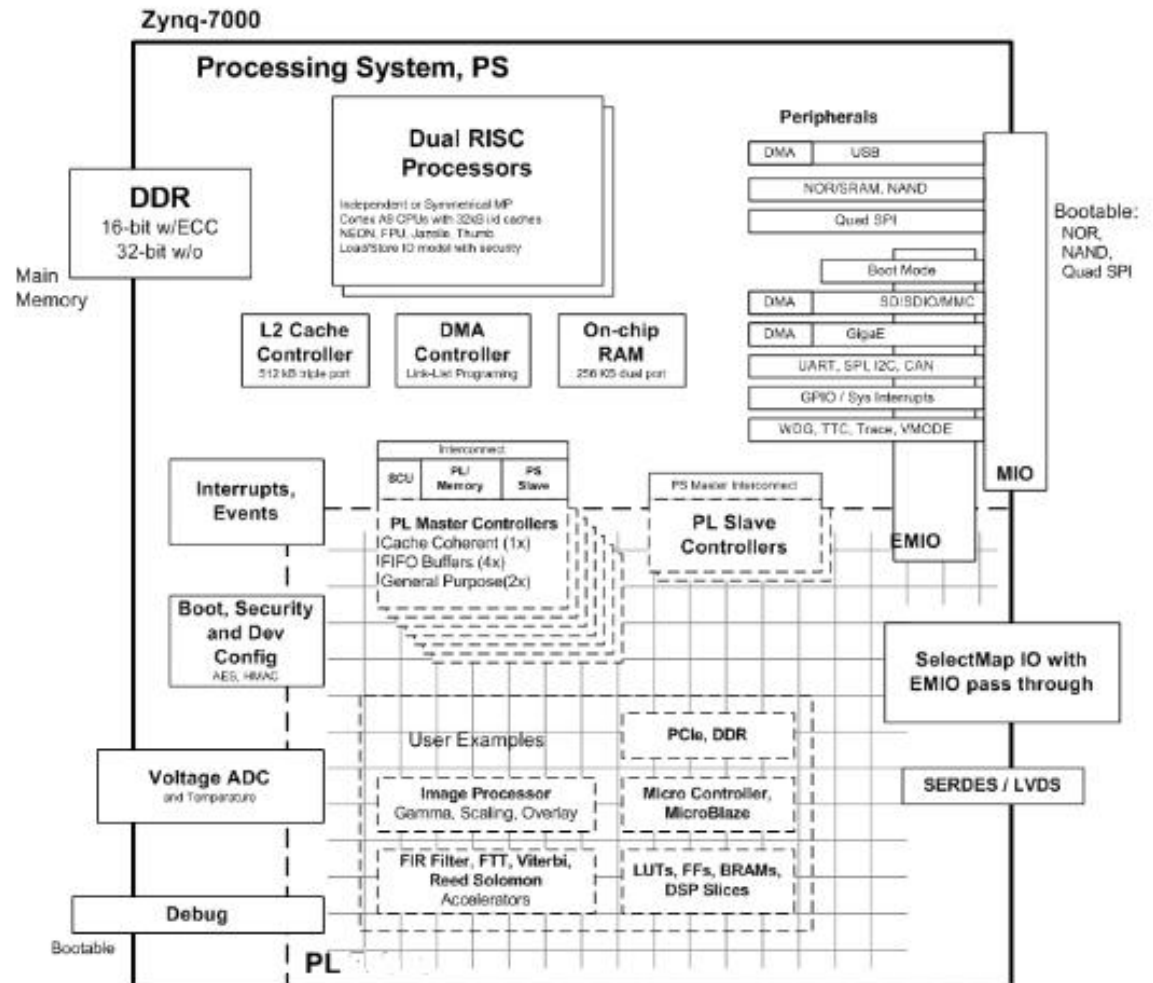
# PetaLinux Tools – Create Project

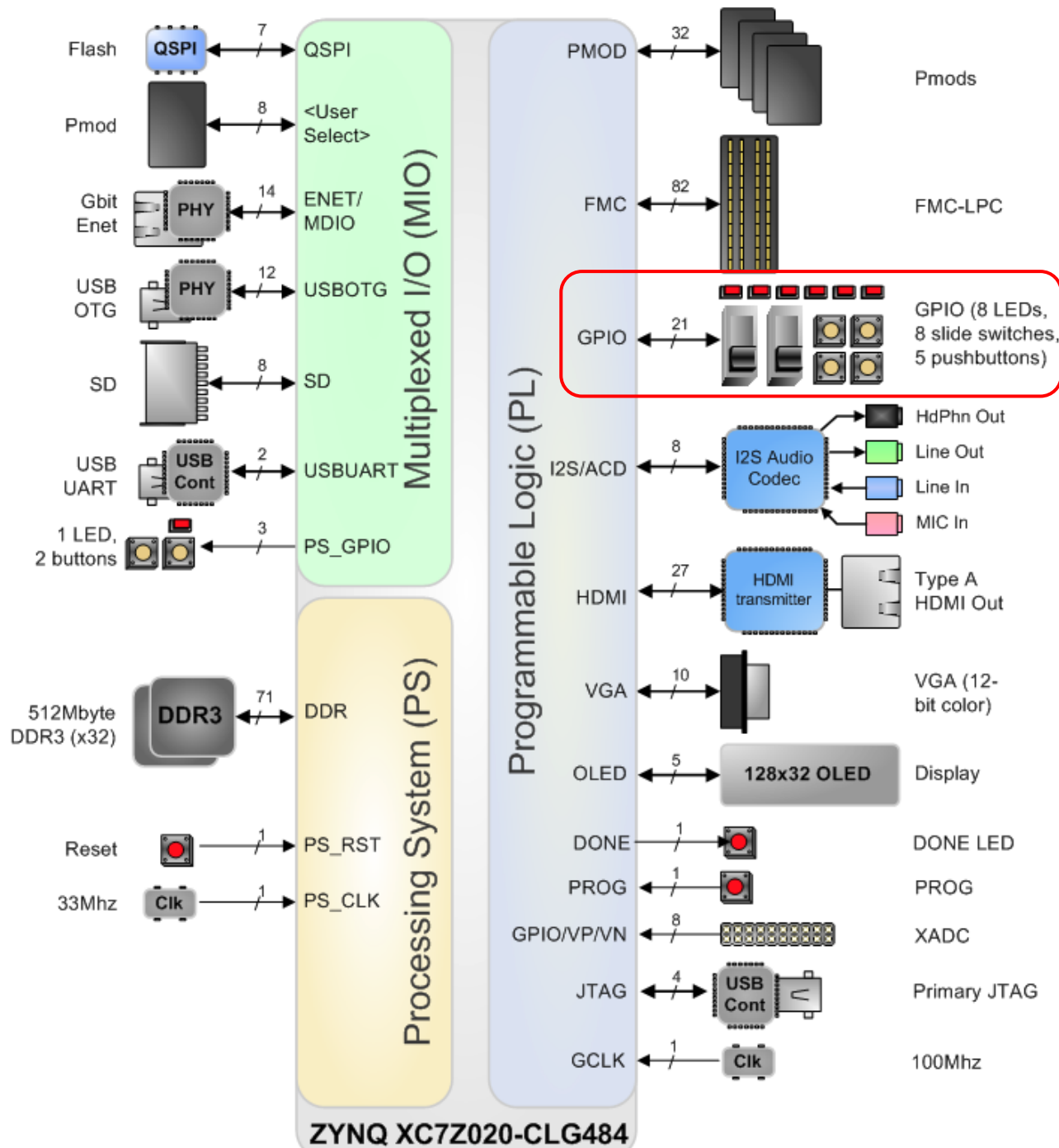
- Host> petalinux-create -t project --name software

```
petalinux@ubuntu:~/emblnx/labs/lab6$ petalinux-create -t project --name software
INFO: Create project: software
INFO: New project successfully created in /home/petalinux/emblnx/labs/lab6/software
```

# Zynq 7000 AP SoC PS and PL Block Diagram

- PS: **P**rocessing system
  - Dual ARM Cortex-A9 processor based
- PL: **P**rogrammable logic
  - Artix™ FPGA







# **LAB 1.2**

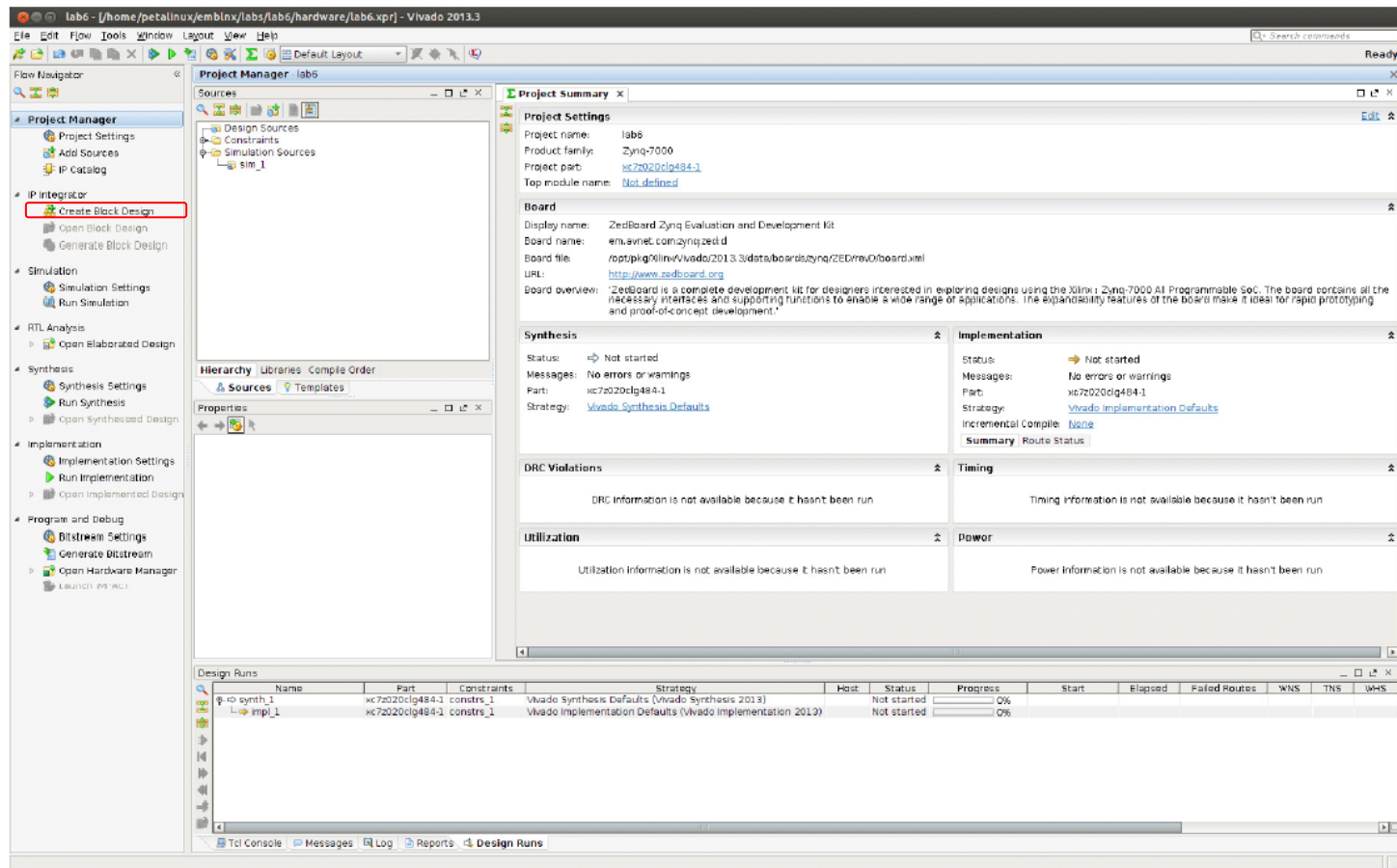
Bare-metal design

# Create Vivado Project

- Host> vivado &
- Click **Create New Project** and click **Next**
- Enter **Project Name** and click **Next**
- Select **RTL Project** and click **Next**
- Under the Specify area select **Boards**
- Select **Zynq** from the Library drop-down list
- Select **Zedboard Zynq Evaluation and Development Kit** and click **Next**
- Click **Finish**

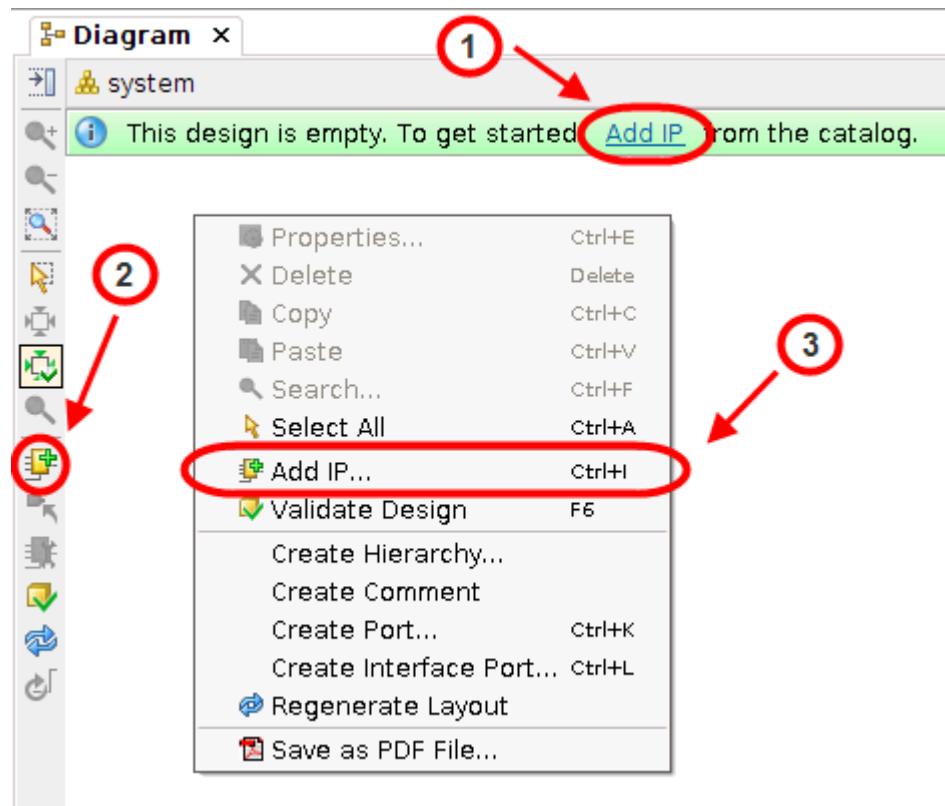
# Create Block Design

- Click **Create Block Design**
- Enter **Design name**, such as system, and click **OK**.



# Create Processing System


- Click **Add IP**
- Search **zynq** and double click **ZYNQ7 Processing System**
- Click **Run Block Automation** and select **/processing\_system7\_0** and Click **OK**



# Configure Processing System

- Double-click the **ZYNQ7 Processing System** block
- (Optional) Deselect all the I/O peripherals except **UART 1**
  - De-select **USB0, ENET 0** and **SD 0** under **MIO Configuration** → **I/O Peripherals**
  - De-select **GPIO MIO** under **MIO Configuration** → **I/O Peripherals** → **GPIO**
  - Deselect **Quad SPI Flash** under **MIO Configuration** → **Memory Interfaces**
  - Deselect **Timer 0** under **MIO Configuration** → **Application Processor Unit**
- Check and Select **M\_AXI\_GP0 interface** under **PS-PL Configuration** → **GP Master AXI**
- Check and Select **FCLK\_RESET0\_N** under **General** → **Enable Clock Resets**
- Check and Select **FCLK\_CLK0** under **Clock Configuration** → **PL Fabric Clocks**

# Create Switch GPIO

- Click the Add IP icon 
- Double-click the **AXI GPIO**
- Click on the **AXI GPIO** block to select it, and in the properties tab, change the name to **sw\_8bit**
- Double click on the **AXI GPIO** block
- Click on **Generate Board Based IO Constraints**, and under **Board Interface**, for **GPIO**, click on **Custom** to view the dropdown menu options, and select **sws\_8bits**
- Click **OK**



# Configure AXI Interface

- Click ***Run Connection Automation*** and select ***/SWs\_8Bits/S\_AXI*** and Click ***OK***
- Double-click the ***AXI Interconnect*** block
- Select ***3*** as the number of master interfaces
- Click ***OK***
- Connect
  - ***AXI Interconnect M01\_ARESETN*** → ***Proc Sys Reset peripheral\_aresetn***
  - ***AXI Interconnect M02\_ARESETN*** → ***Proc Sys Reset peripheral\_aresetn***
  - ***AXI Interconnect M01\_ACLK*** → ***ZYNQ7 Processing System FCLK\_CLK0***
  - ***AXI Interconnect M02\_ACLK*** → ***ZYNQ7 Processing System FCLK\_CLK0***

# Create Button and LED GPIOs

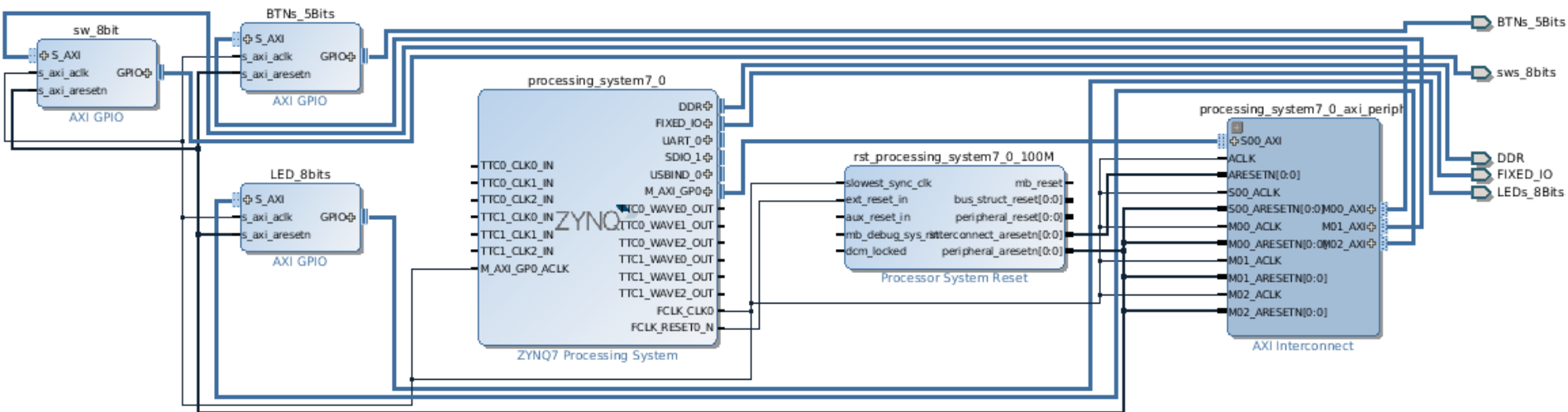
- Create two more GPIO for Buttons and LEDs.
  - BTN<sub>s</sub>\_5Bits (select the *board interface* as ***BTN<sub>s</sub>\_5Bits***)
  - LED<sub>s</sub>\_8Bits (select the *board interface* as ***LED<sub>s</sub>\_8Bits***)
- Follow the previous steps for each additional peripherals (***BTN<sub>s</sub>\_5Bits*** and ***LED<sub>s</sub>\_8Bits***)

Peripheral Name	Board Interface	S_AXI Connection	s_axi_aclk	s_axi_aresetn
BTN <sub>s</sub> _5Bits	BTN <sub>s</sub> _5Bits	M01_AXI	FCLK_CLK0	peripheral_aresetn
LED <sub>s</sub> _8Bits	LED <sub>s</sub> _8Bits	M02_AXI	FCLK_CLK0	peripheral_aresetn



# Connect GPIOs to External Pins

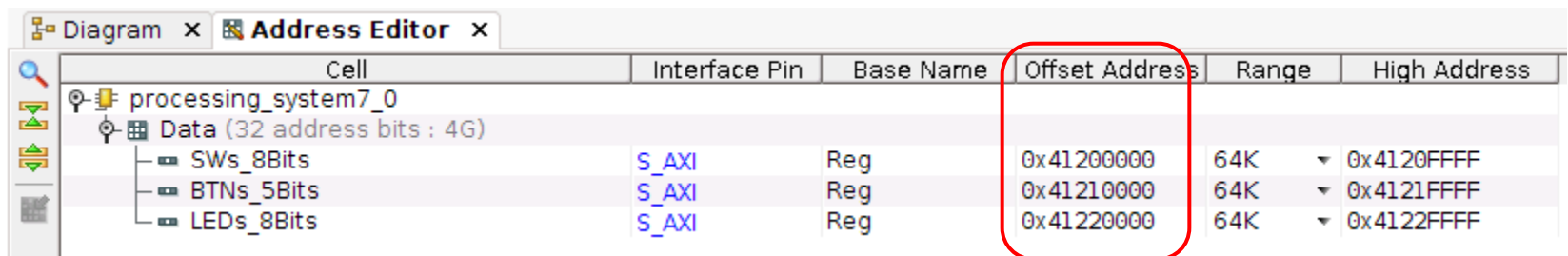
- Right-click the ***GPIO*** port of the ***SWs\_8Bits*** instance and select ***Make External***
- Select the ***GPIO*** port and change the name to ***SWs\_8Bits*** in the ***External Interfaces Properties Name*** field
- Click ***Run Connection Automation*** and select ***/BTN\_5Bits/GPIO***
- Select ***BTN\_5Bits*** in the ***Select Board Interface***
- Click ***OK***
- Similarly, use ***Run Connection Automation*** to create an external port for the ***LEDs\_8Bits*** instance

# Block Design



# Assign Address

- Select the **Address Editor** tab and click 
- Click 
- Select **File** → **Save Block Design**
- Select **Tools** → **Validate Design**



The screenshot shows the 'Address Editor' window with a table of memory addresses. The table has columns for Cell, Interface Pin, Base Name, Offset Address, Range, and High Address. The 'Offset Address' column is highlighted with a red box.

Cell	Interface Pin	Base Name	Offset Address	Range	High Address
processing_system7_0					
Data (32 address bits : 4G)					
SWs_8Bits	S_AXI	Reg	0x41200000	64K	0x4120FFFF
BTNs_5Bits	S_AXI	Reg	0x41210000	64K	0x4121FFFF
LEDs_8Bits	S_AXI	Reg	0x41220000	64K	0x4122FFFF

# Generate Bitstream

- Right Click on the block diagram file, ***system.bd***, and select ***Create HDL Wrapper***, then click ***Copy and Overwrite***
- Click ***Run Synthesis*** then click ***Save***
- Select ***Open Synthesized Design*** and click ***OK***
- Click on ***Generate Bitstream*** then click ***Yes***
- Select ***Open Implemented Design*** option and click ***OK***



# Export to Xilinx SDK

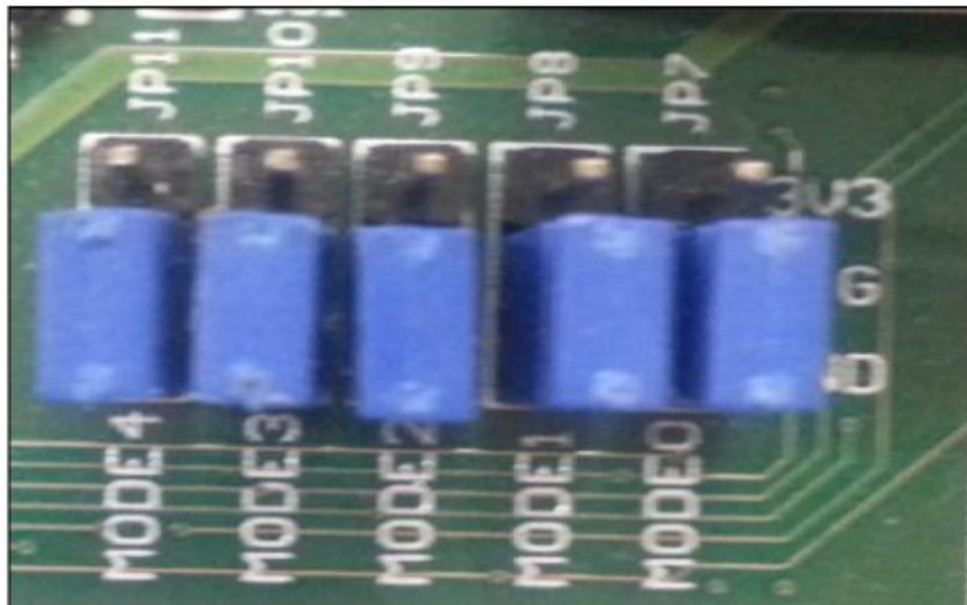
- Click ***File*** → ***Export*** → ***Export Hardware for SDK***
- Check the ***Launch SDK*** and ***Include bitstream*** box and click ***OK***
- Click ***Yes***

# Write a Bare-metal Application

- In SDK, select ***File*** → ***New*** → ***Application Project***
- Name the project ***Lab\_1.2.1*** and, select ***Create New*** in the ***Board Support Package*** section and type the name ***standalone\_bsp*** and click ***Next***
- Select ***Empty Application*** and click ***Finish***
- The library will automatically generated at  
~\lab1.2.1\lab1.2.1.sdk\SDK\SDK\_Export\standalone\_bsp\ps7\_cortexa9\_0\include directory
- Create a new C program at ***Lab\_1.2.1*** in the project view
- Finish the program

# Program FPGA

- Make sure that the jumper settings are set to JTAG Mode
- Select **Xilinx Tools** → **Program FPGA**
- Check the **Bitstream** path is pointing to  
~\lab1.2.1\lab1.2.1.sdk\SDK\SDK\_Export\hw\_platform\_0\system.bit and click **Program**
- Right click on project **Lab\_1.2.1** and select **Run As** → **Launch on Hardware (GDB)**



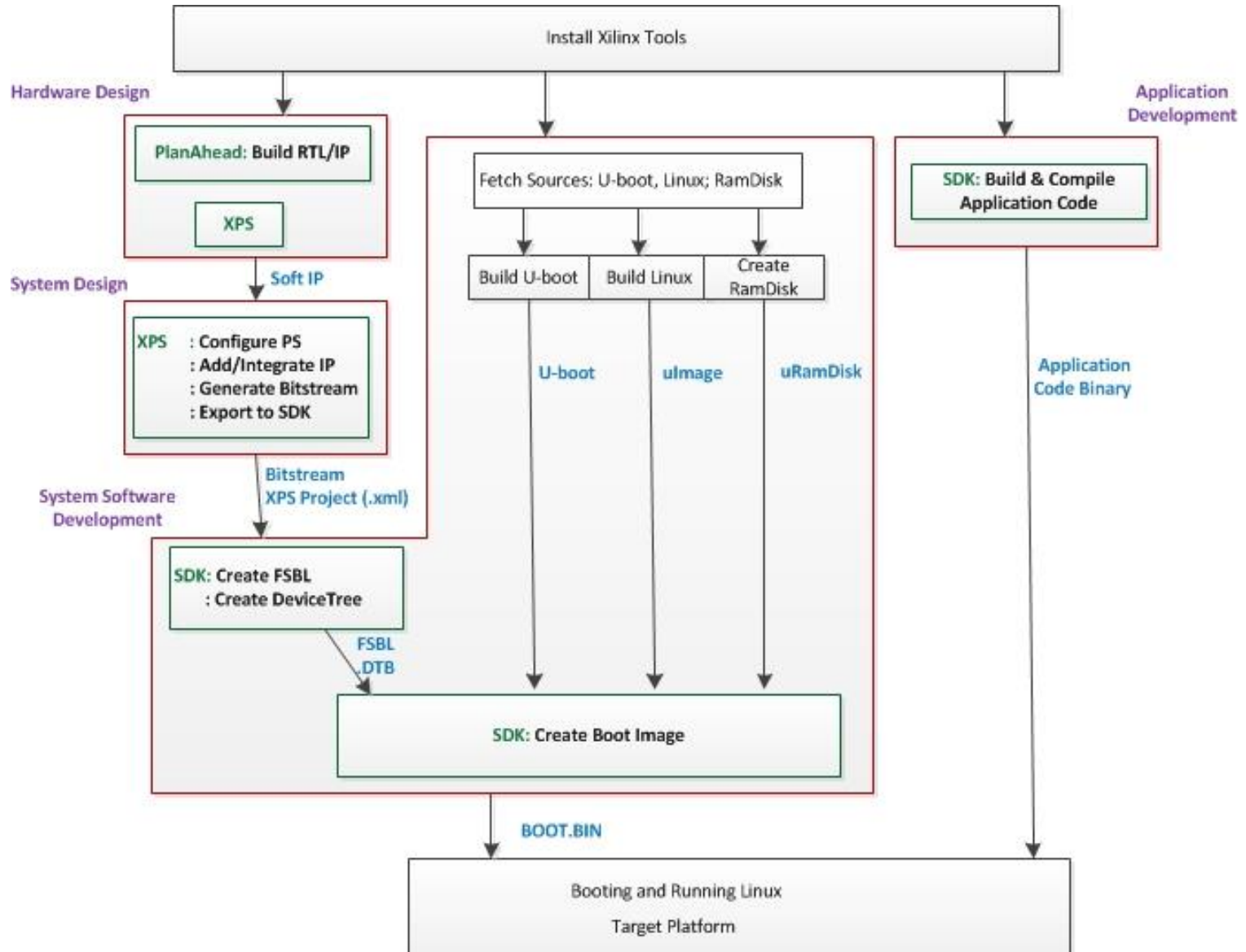
# Observe UART Output

- Start the *GtkTerm* at host
- Select */dev/ttyACM0* at *Port* and *115200* at *Baud Rate*

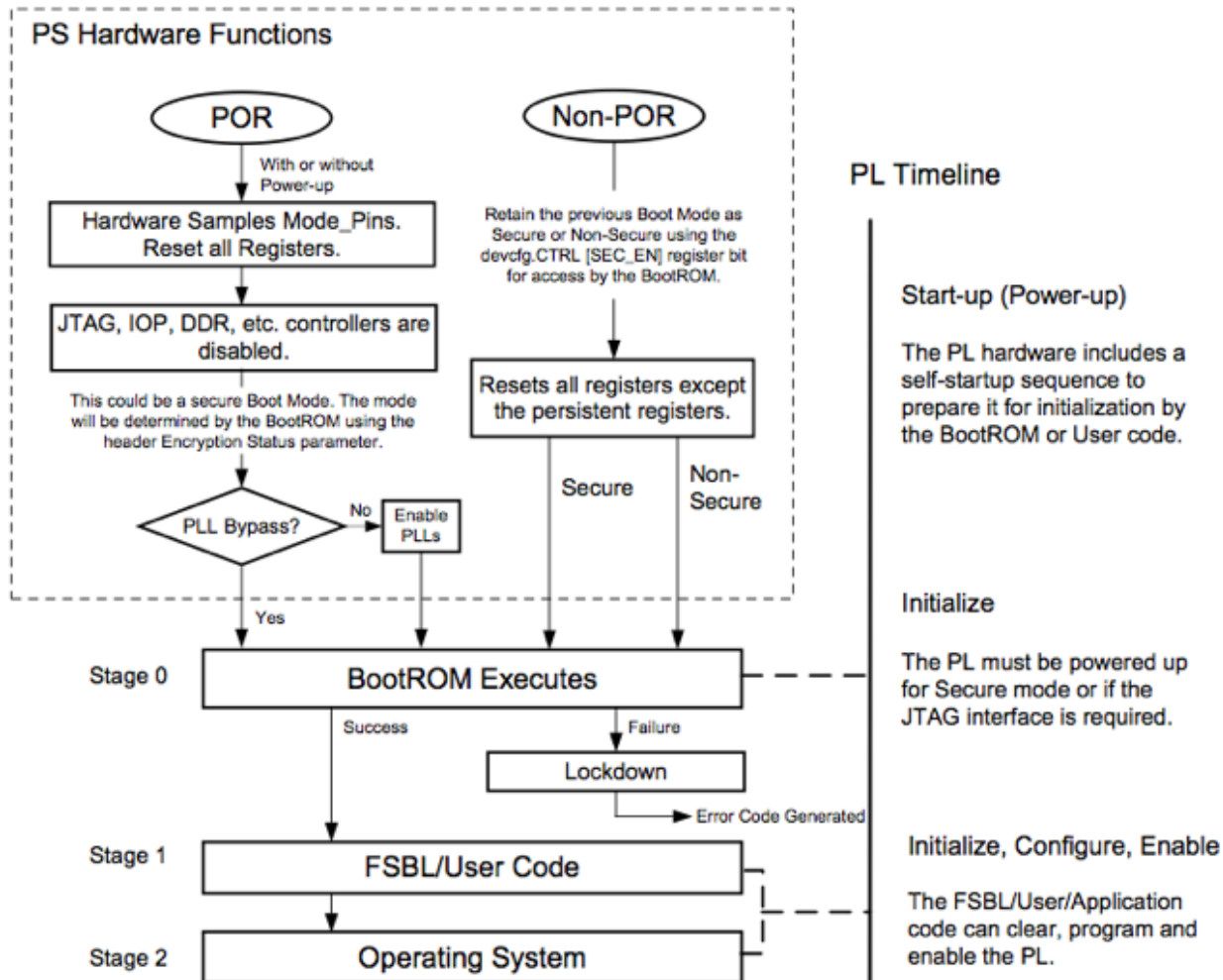
# **LAB 1.2.2**

Hello world program on basic petalinux

# Design Flow



# Zynq Boot Flow



# Hardware Design

- In Vivado, create a new design with only **ZYNQ7 Processing System** block or using the hardware design from Lab 1.2.1
  - The following components are the minimal requirement to boot up Linux:
  - **ENET 0** (optimal but suggested) and **SD 0** under **MIO Configuration** → **I/O Peripherals**
  - **Quad SPI Flash** under **MIO Configuration** → **Memory Interfaces**
  - **Timer 0** under **MIO Configuration** → **Application Processor Unit**
  - (Please check the setting of above item is the same as it in default **ZYNQ7 Processing System** block )
- **Run Synthesis** and/or **Generate Bitstream** again if you revise the design or create a new design
- Click **File** → **Export** → **Export Hardware for SDK**
- Check the **Launch SDK** box and click **OK**
- Click **Yes**



# Add PetaLinux SDK Repository

- In SDK, select ***Xilinx Tools*** → ***Repositories***
- Select ***Repositories*** then click ***New***
- Browse and Select `~/opt/petalinuxv2013.3.10-final/components/edk_user_repository` then click ***OK***
- Click ***OK***

# Create PetaLinux BSP (Board Support Package)

- Select ***File*** → ***New*** → ***Board Support Package***
- Select ***hw\_platform\_0*** at ***Hardware Platform*** field and ***petalinux*** at ***Board Support Package OS*** field, and then click ***Finish***
- Select ***petalinux*** in the ***Board Support Package Settings*** window and check the following options
  - ***stdin: ps7\_uart\_1***
  - ***stdout: ps7\_uart\_1***
  - ***main\_memory: ps7\_ddr\_0***
  - ***flash\_memory: ps7\_qspi\_0***
  - ***sdio: ps7\_sd\_0***
  - ***ethernet: ps7\_ethernet\_0***
- Click ***OK***

# Create First Stage Boot Loader (FSBL)

- Select ***File*** → ***New*** → ***Application Project***
- Enter ***zynq\_fsbl*** as the ***Project name***
- Set the follow options:
  - ***Hardware Platform: hw\_platform\_0***
  - ***Target Software OS Platform: standalone***
  - ***Board Support Package: Create New: zynq\_fsbl\_bsp***
- Click ***Next***
- Select ***zynq FSBL*** in the ***Available Templates*** box
- Click ***Finish***

# Create and Configure Petalinux Project

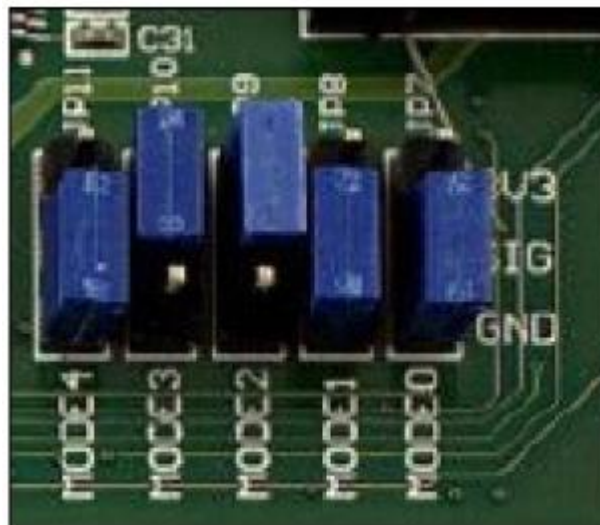
- Start a **Terminal** at host
- Host> cd lab1.2.2
- Host> petalinux-create -t project --name software
- Host> cd ~/lab1.2.2/lab1.2.2.sdk/SDK/SDK\_Export/petalinux\_bsp\_0
- Host> petalinux-config --get-hw-description -p ~/lab1.2.2/software
- Host> cd ~/lab1.2.2/software
- Host> petalinux-config
- Select **Host name** and change to your group number
- Select **System boot device** and change to **SD card**
- Select **OK** then select **Exit** and select **Yes**
- Host> petalinux-build
- Uboot would be generated at *~/lab1.2.2/software/images/linux*

# Create Boot Image

- In SDK, right-click the **zynq\_fsbl** project and select **Create Boot Image**
- Click **Add** then click **Browse** and select  
`~/lab1.2.2/software/images/linux/u-boot.elf`
- Check the order of boot: bootloader -> bitstream -> u-boot
- Click **OK**
- Click **Create Image**
- **mv**  
`~/lab1.2.2/lab1.2.2.sdk/SDK/SDK_Export/zynq_fsbl/bootimage/output.bin`  
`~/lab1.2.2/lab1.2.2.sdk/SDK/SDK_Export/zynq_fsbl/bootimage/BOOT.BIN`
- Copy  
`~/lab1.2.2/lab1.2.2.sdk/SDK/SDK_Export/zynq_fsbl/bootimage/BOOT.BIN` and `~/lab1.2.2/software/images/linux/image.ub` to SD card

# Boot the OS on Zedboard

- Set the jumper to SD card mode (*JP9* and *JP10* are set to High)
- Plug in SD card
- Start the ***GtkTerm*** at host
- Select ***/dev/ttyACM0*** at ***Port*** and ***115200*** at ***Baud Rate***
- U-Boot-PetaLinux> run sdboot
- The user name and password are both *root* for petalinux



# Design a Hello World Program

- At host
  - Host> cd /tftpboot
  - Host> vi lab1.2.2.c
  - Write a Hello World program
  - Host> arm-xilinx-linux-gnueabi-gcc lab1.2.2.c -o lab1.2.2
  - Setup the Host IP and use LAN cable to connect Host and Target
    - Host> ifconfig eth0 192.168.0.1
- At Zedboard
  - Group X> ifconfig eth0 192.168.0.10
  - Group X> tftp -g 192.168.0.1 -r lab1.2.2
  - Group X> chmod 755 lab1.2.2
  - Group X> ./lab1.2.2

# Download the Hello World Program to Zedboard

- At Host
  - Set up the Host IP and use LAN cable to connect Host and Target
    - Host> ifconfig eth0 192.168.0.1
- At Zedboard
  - Group X> ifconfig eth0 192.168.0.10
  - Group X> tftp -g 192.168.0.1 -r lab1.2.2
  - Group X> chmod 755 lab1.2.2
  - Group X> ./lab1.2.2



# Mount From SD card

- `root@GroupID:~# mount /dev/mmcblk0p1 /mnt`
- `root@GroupID:~# cd /mnt/`
- `root@GroupID:~# ./lab1.2.2`

# **LAB 1.3**

Basic I/O related program on petalinux

# Hint

- Two ways to access
  - Map the physical address of I/O to virtual address
  - Enable User space I/O and map the I/O under */dev* to application's address space

# Lab Assignment

- Today's work (10%)
  - Install vivado & xsdk
- Lab 1.2.1 (30%)
  - Design hardware for lighting on the LEDs with switching on the corresponding switches
  - Show the button status from UART
- Lab 1.2.2 (30%)
  - Change host name of petalinux to you group number as "Group #"
  - Show "Hello Group #" under OS
  - Minimize the file size of BOOT.BIN
- Lab 1.3 (20%)
  - Use LEDs to show the result of add, subtract, multiply and divide
  - Switches as input number
  - Four Buttons denote add, subtract, multiply and divide
- Bonus (10%)
  - Implement Lab 1.3 in bare-metal design
- Report (10%)

# Deadline

- Demo:
  - Lab1.1 & 1.2.1: 3/3
  - Lab1.2.2 & 1.3: 3/10
- Report: 3/10 23:59
- Discussion
  - [https://2016\\_embedded\\_system\\_lab.hackpad.com/](https://2016_embedded_system_lab.hackpad.com/)