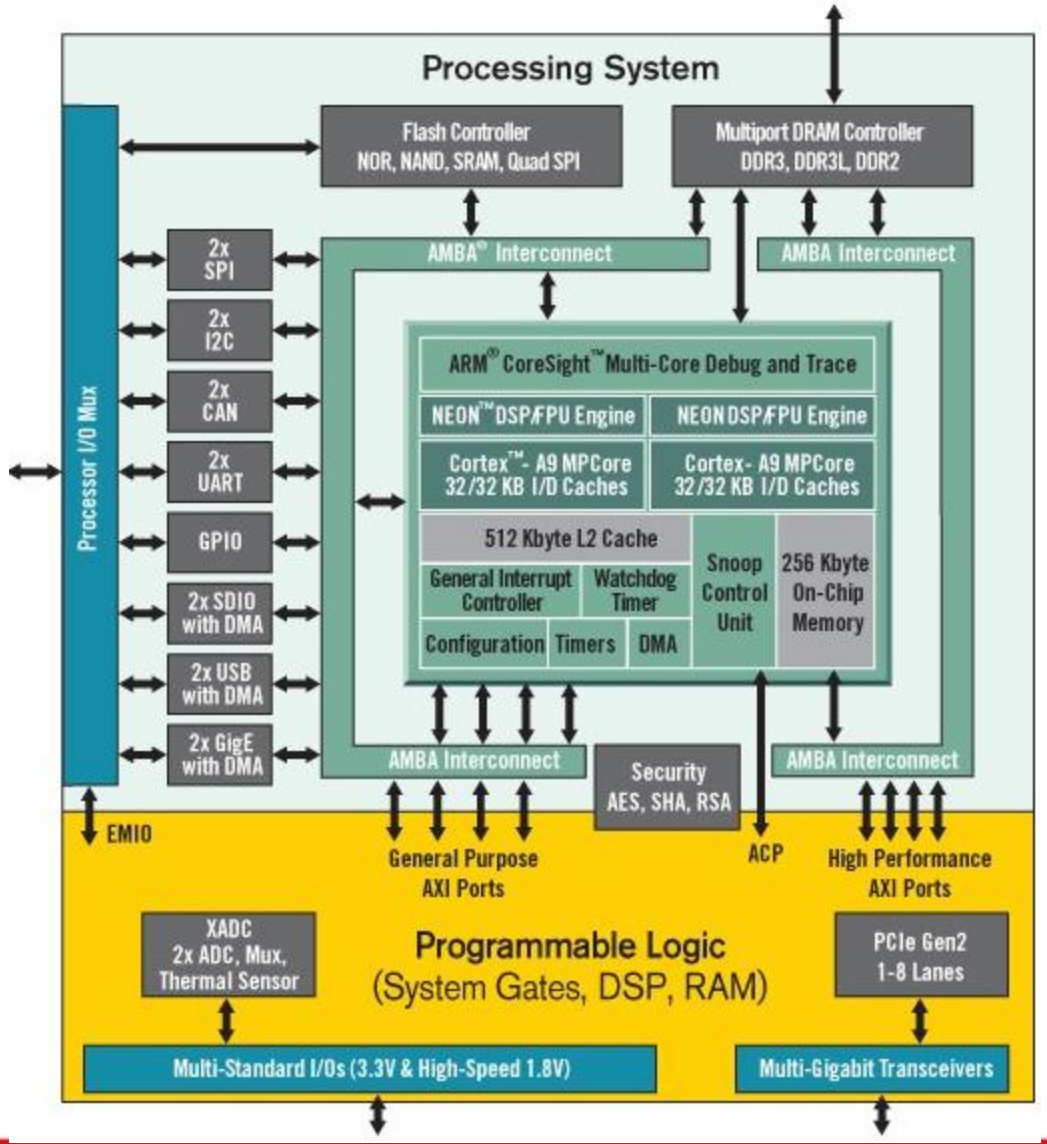
ZYNQ平台嵌入式软件开发

Accelerating Your Success™

Zynq-7000 All Programmable SoC

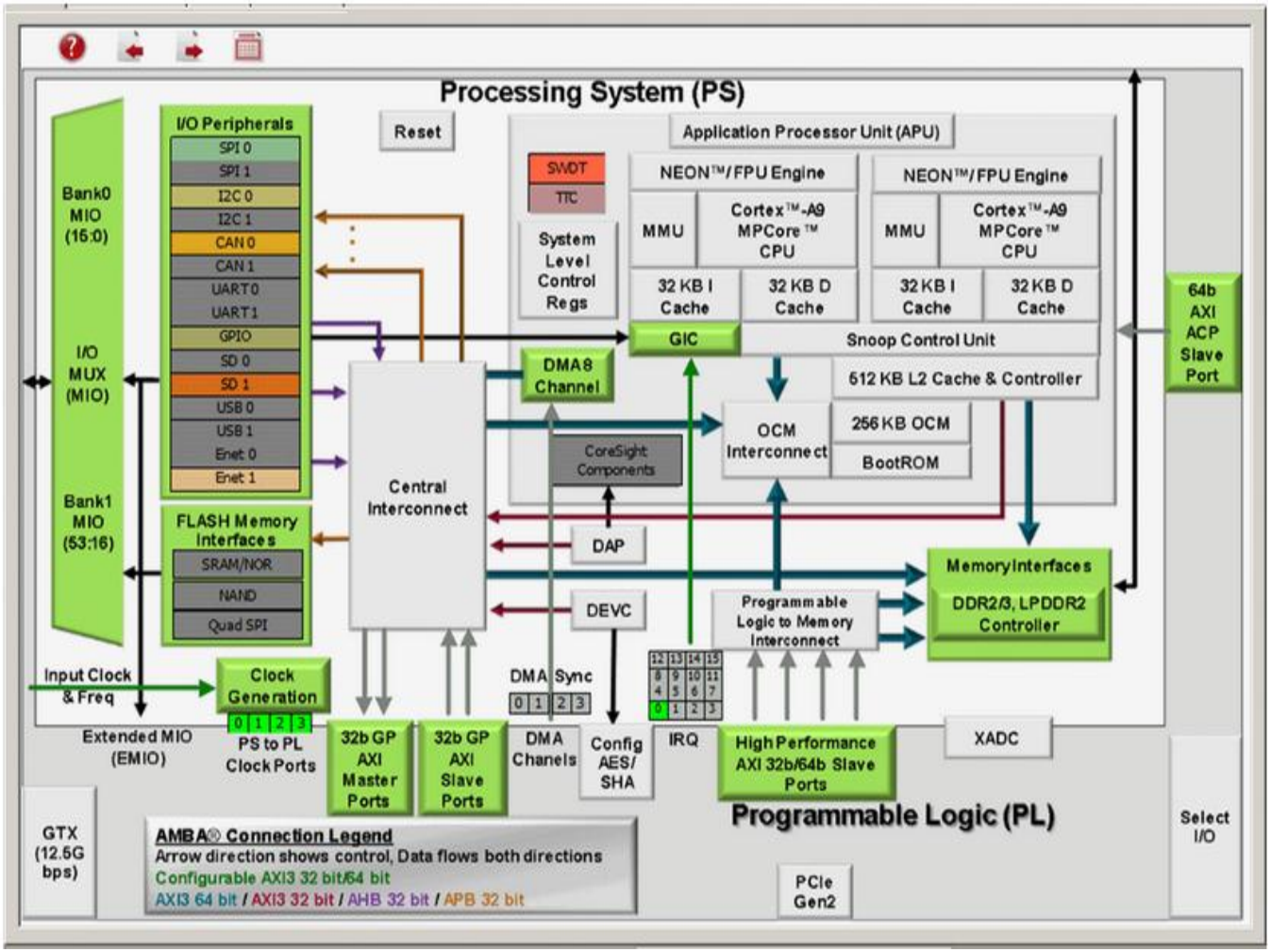


- 双核 ARM Cortex™-A9 MPCore
  - 高达 1GHz
  - 可通过 NEON 扩展及单双精度浮点单元进行增强
  - 32kB 指令及 32kB 数据 L1 缓存
- 统一的 512kB L2 缓存
- 256kB 片上存储器
- DDR3、DDR3L、DDR2 以及 LPDDR2 动态存储控制器
- 2 个 QSPI、NAND Flash 以及 NOR 闪存控制器
- 2 个 USB2.0 (OTG)、2 个 GbE、2 个 CAN2、0B、2 个 SD/SDIO、2 个 UART、2 个 SPI、2 个 I2C、4 个32b GPIO
- 一级引导载入程序、AES 和 SHA 256b 解密的RSA 认证以及安全引导认证
- 双 12 位 1Msps 数模转换器
  - 最多接受17对差分模拟输入
- Advanced Low Power 28nm Programmable Logic

Zynq-7000 操作系统支持

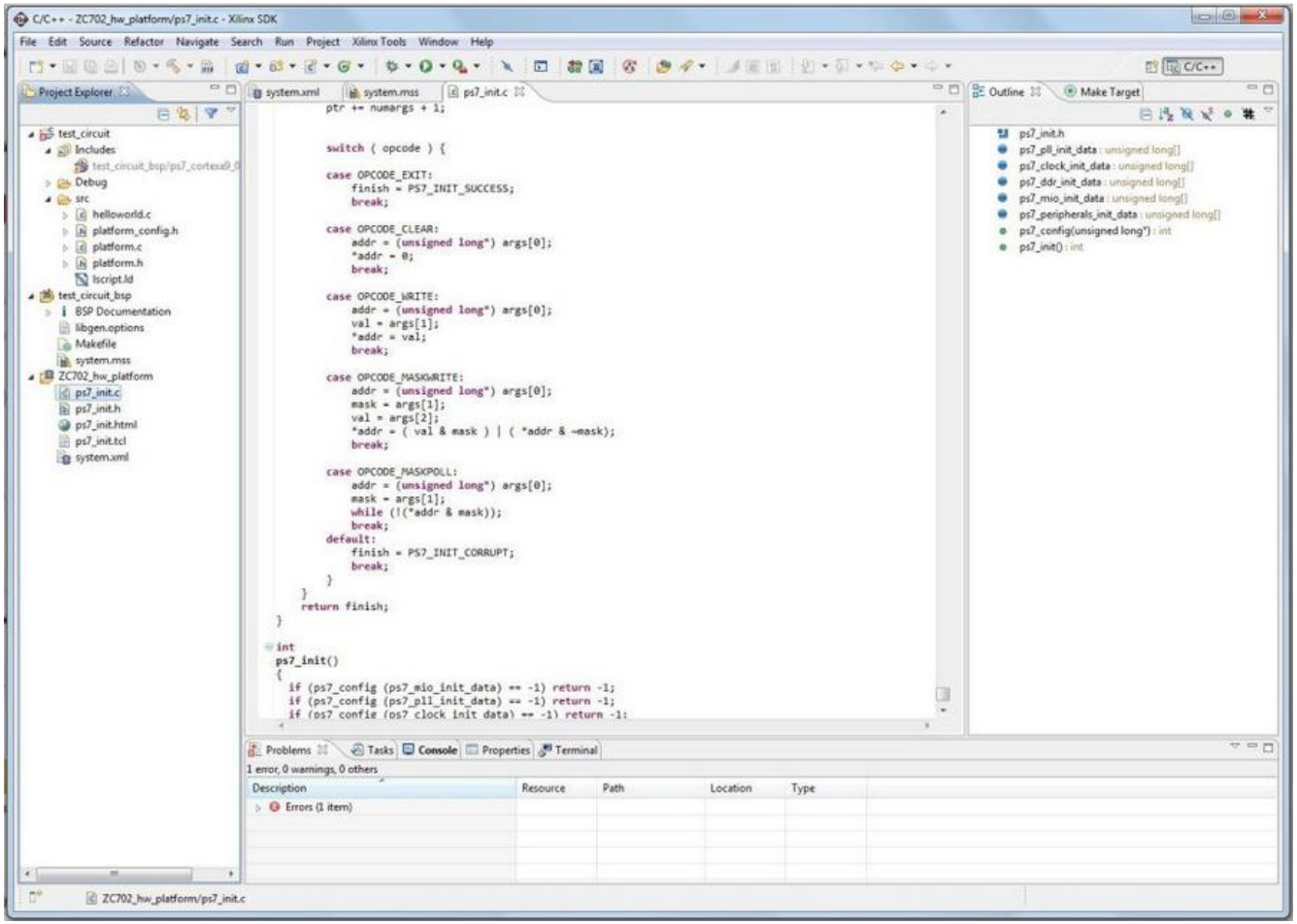
- **开源OS**
  - Linux 3.8.0
  - Andriod 2.3
  - FreeRTOS
- **商业OS**
  - Adeneo Embedded - Windows Embedded Compact 7
  - eSOL - uITRON/T-Kernel
  - ExpressLogic - ThreadX
  - Micrium - uC/OS
  - Wind River - Linux
  - ENEA - OSE
  - SYSGO - PikeOS
  - ETAS - RTA-OS
  - iVeia - Android
  - Xilinx - PetaLinux
  - Quadros - Quadros
  - Wind River - VxWorks
  - Green Hills Software - INTEGRITY

Zynq-7000 开发工具 ---- XPS



- XPS 可实现完整的配置和定制操作
- 管理 Zynq-7000 AP SoC 外设 IO 引脚启动配置和初始化
- 器件安全、回读引导和比特流管理措施
- 项目专用配置感知
- 无需用户掌握具体硬件知识
- 利用可展开的软件图自动构建一级引导载入程序

Zynq-7000 开发工具 ---- XPS

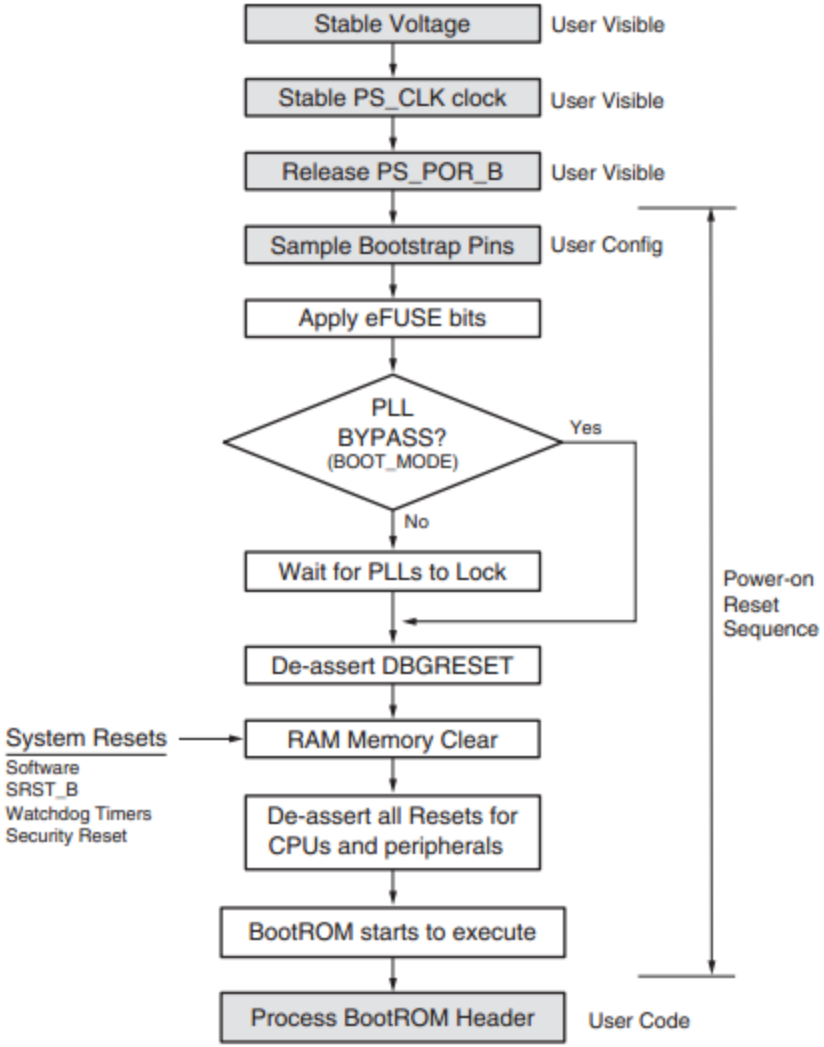


- 使用 Eclipse CDT 开发环境 (version 3.8)
- 完整的集成设计环境 (IDE) 可用于连接 Vivado 和 ISE 嵌入式硬件设计环境
- 实现完整的软件设计和调试流程支持，包括全新多内核和硬件/软件调试功能
- Xilinx 版 Mentor Sourcery CodeBench Lite (version 2012.09) 支持
- 定制库和器件驱动程序
- 裸机与 Linux 开发
- 支持同构和异构开发的多内核

Zynq-7000 嵌入式软件开发流程

- 利用ISE工具完成硬件设计
- 利用SDK工具生成FSBL（First Stage Boot Loader）
- 利用JTAG工具下载调试FSBL的功能
- 利用SDK工具生成由FSBL、Bitstream、SSBL(Second Stage Boot Loader)组成的BOOT.BIN
- 将BOOT.BIN、OS映像文件、内存文件系统等文件拷贝到系统设计指定的存储设备中（QSPI\NAND\NOR Flash、SD卡），调试操作系统基本功能
- 开发调试用户态功能程序

Zynq-7000 启动流程



- **MMU, Icache, Dcache, L2 cache are all disabled**
- **Both processors are in the supervisor state**
- **ROM code is masked and inaccessible**
- **192 KB of OCM is accessible starting at address 0x0 while 64 KB is accessible starting at address 0xFFFF0000**
- **CPU0 branches into the stage 1 image if no failure takes place**
- **CPU1 is in a WFE state while executing code located at address 0xFFFFFE00 to 0xFFFFFFF0**

Zynq-7000 First Stage Boot Loader

- **功能简介**
  - 由BootROM将其从外部存储设备拷贝到OCM运行
  - 完成硬件初始化，为SSBL提供可工作的硬件环境
  - 解析BOOT.BIN文件，加载Bitstream、SSBL
  - AMP环境下需要加载CPU1上运行的ELF文件
  - CPU0进入到SSBL中执行
- **FSBL工程根据XPS的XML文档选择需要的BSP驱动，并生成ps7_init.c文件**
- **相关文档：**
  http://www.xilinx.com/support/documentation/user_guides/ug821-zynq-7000-swdev.pdf

Zynq-7000 Bootgen and BIF file format

Bootgen

- A standalone tool for creating a bootable image suitable for the Zynq-7000 AP SoC processor. The program assembles the boot image by prefixing a header block to a list of partitions. Each partition can be optionally encrypted and authenticated.

BIF

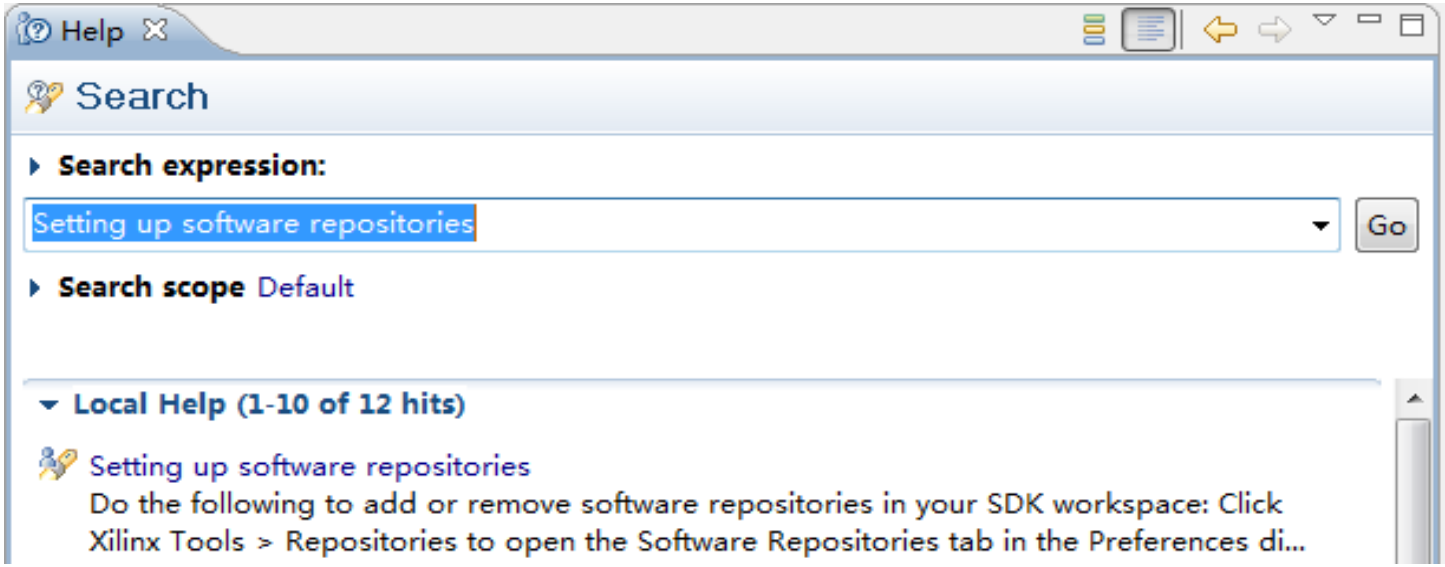
- The BIF file specifies each component of the boot image, in order of boot, and allows optional attributes to be applied to each image component. Each image component is usually mapped to a partition, but in some cases an image component can be mapped to more than one partition if the image component is not contiguous in memory.

相关文档

http://www.xilinx.com/support/documentation/user_guides/ug821-zynq-7000-swdev.pdf

SDK中BSP驱动调试

- 执行Clean Project后，工程中的BSP代码会被清除，后续Build Project命令会从SDK安装目录拷贝新的BSP代码
- 通过创建并添加本地Repository，可以在不影响其他Workspace的情况下进行修改调试BSP代码。
- 添加新的Repo的步骤可通过以下方式搜索：
  - SDK中按F1，打开帮助窗口，并点击Search链接
  - 搜索框中输入Setting up software repositories，选择第一个链接



Zynq-7000 DeviceTree

- The ARM architecture has become a major headache in the Linux community: Even though the processors share the same compiler and many functionalities, each embodiment (i.e. chip) has its own addresses for the registers, and a slightly different configuration. On top of that, each board has its own set of external components. The result is a wild forest of header files, patches and special configuration parameters in the kernel tree, each combination matching a specific board with a specific chip containing an ARM processor. In short, it has turned out to be an ugly and unmaintainable pile of hacks which nobody is really fond of.

More info Refer to: http://xillybus.com/tutorials/device-tree-zynq-1

- **Device Tree in Linux Kernel**
  - http://www.wiki.xilinx.com/Device+Tree+Generator

Zynq-7000 Linux on Zynq

- **U-boot**
  - https://github.com/Xilinx/u-boot-xlnx
- **Ramdisk**
  - http://www.wiki.xilinx.com/Build+and+Modify+a+Rootfs
- **Kernel**
  - https://github.com/Xilinx/linux-xlnx
- **Drivers:**
  - http://www.wiki.xilinx.com/Linux+Drivers

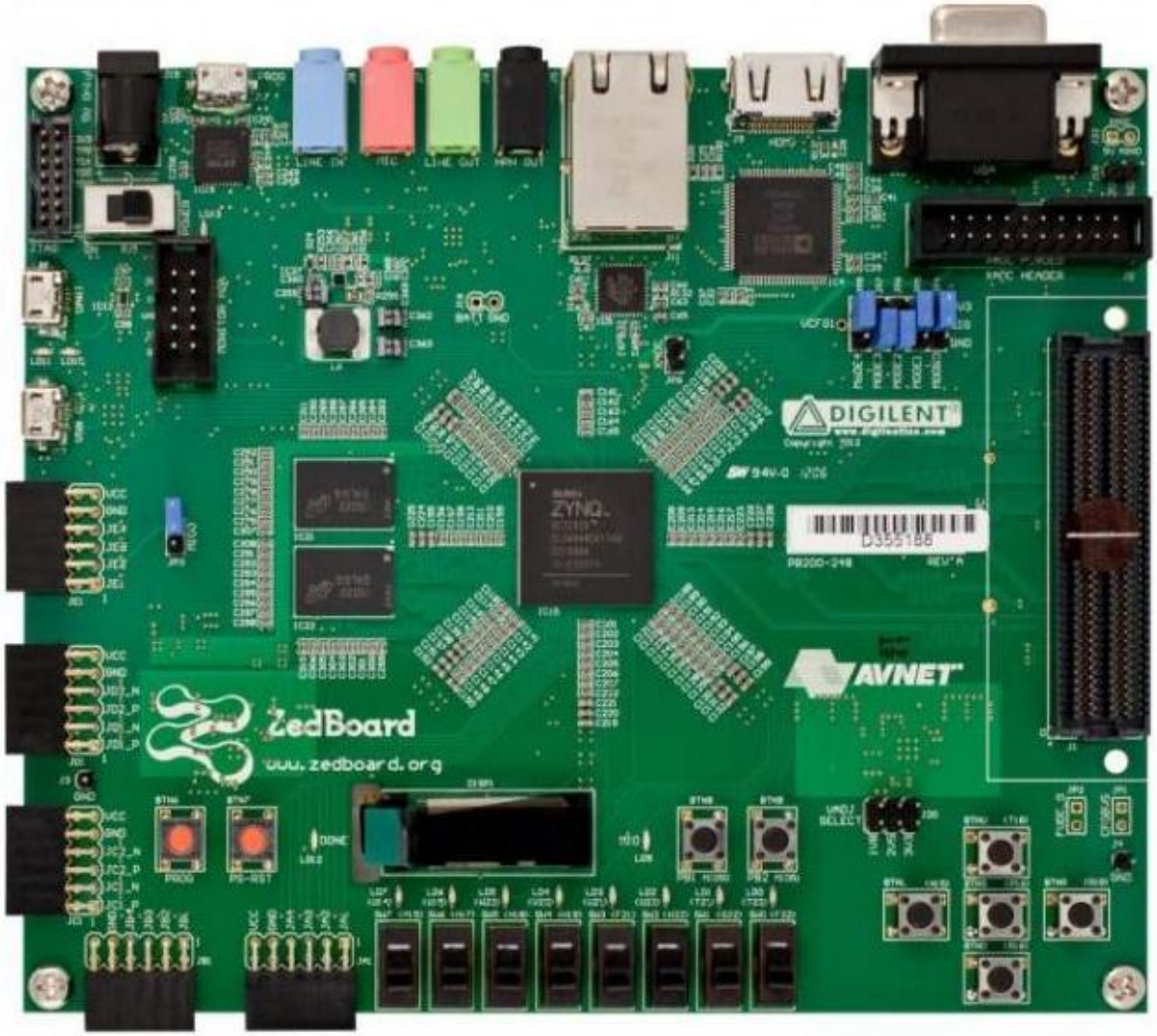
Zynq-7000 Multi-OS Support

- **AMP**
  - The Zynq AP SoC can be configured to run independent software stacks on each of its processor cores. Such approaches can be configured as either unsupervised AMP (where one of the operating systems is responsible for system management and coordination), or as an AMP configuration created using ARM TrustZone architecture which provides a light weight virtual machine of sorts, on which Linux and RTOS code can independently execute.
- **Example:**
  - Xapp1078 : **Linux / Bare-metal AMP**
  - Xapp1079 : **Bare-metal / Bare-metal AMP**
- **More info Refer to:**
  - http://www.wiki.xilinx.com/Multi-OS+Support+(AMP+%26+Hypervisor)

Zynq-7000 Power Management

- **Cpufreq**
  - The cpufreq framework is used to scale the CPU frequency.
- **Cpuidle**
  - The cpuidle framework manages CPU idle levels.
- **Suspend**
  - The suspend framework provides the interface to enter sleep states, like the well known 'suspend to disk/RAM' on laptops.
- **Wake on UART**
- **Wake on GPIO**
- **Wake on Debugger**
- **More info Refer to:**
  - http://www.wiki.xilinx.com/Zynq+Power+Management

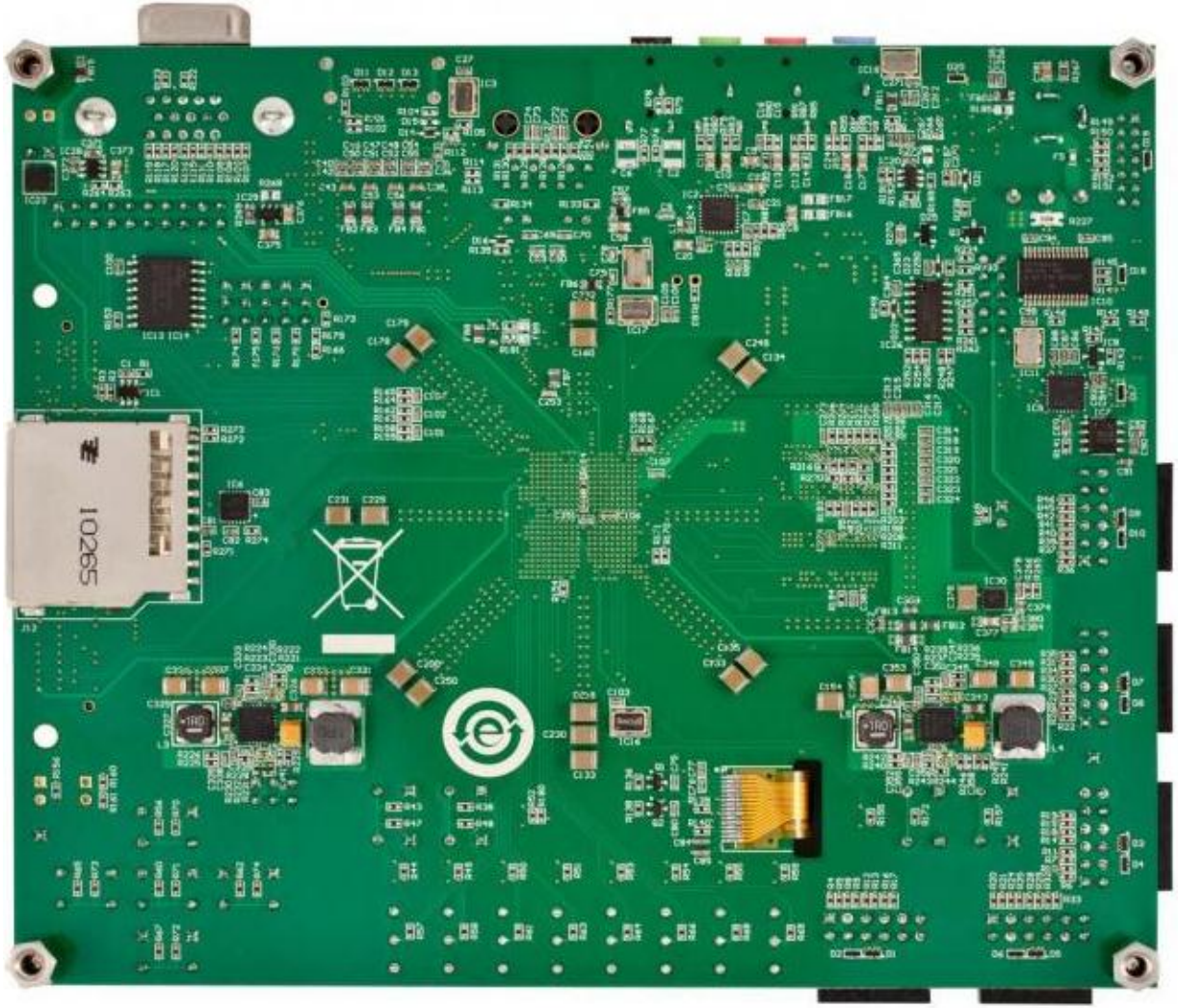
Zynq-7000 ZedBoard



Key Features

- **Zynq Processor**
  - Dual ARM® Cortex™-A9 MPCore™
  - Up to 667 MHz operation
  - NEON™ Processing / FPU Engines
- **Memory**
  - 512 MB DDR3 memory (1066 Mbps)
  - 256 Mb Quad SPI Flash
  - Full size SD/MMC card cage
  - 4 GB SD Card Included
- **Connectivity**
  - 10/100/1000 Ethernet
  - USB OTG (Device/Host/OTG)
  - USB UART
- **Expansion**
  - FMC (Low Pin Count)
  - (5) Pmod™ headers (2x6)
- **Video/Display**
  - HDMI output (1080p60 + audio)
  - VGA connector
  - 128 x 32 OLED
  - User LEDs (9)
- **User Inputs**
  - Slide switches (8)
  - Push button switches (7)

Zynq-7000 ZedBoard



Key Features

- **Audio**
  - 24-bit stereo audio CODEC
  - Stereo line in/out
  - Headphone
  - Microphone input
- **Analog**
  - Xilinx XADC header
  - Supports 4 analog inputs
  - 2 Differential / 4 Single-ended
- **Debug/Programming**
  - On-board USB JTAG programming port
  - ARM Debug Access Port (DAP)
- **Power**
  - 12V DC input @ 3.0 A (Max)
- **Dimensions**
  - Length: 6.3 inches
  - Width: 5.3 inches
- **Certification**
  - CE and RoHS certified

Zynq-7000 ZedBoard Resource

Reference Design

- AMP on Zedboard
- LCD & Touch Panel
- HDMI Video Output
- Motor Controller
- Ubuntu\Andriod\WinCE On Zedboard
- OpenCV
- More and more…

Online Resource

- http://www.zedboard.org
- http://xilinx.eetrend.com/tag/2213
- http://www.google.com
- http://www.baidu.com

Zynq-7000 ZedBoard 书籍

- 本书分为基础篇和进阶篇两部分。

  基础篇中主要介绍了ZedBoard板、Zynq器件、开发工具链、Zynq的体系结构及其启动过程等基本的器件与板卡情况，同时考虑了软件工程师的知识结构。在第8章中介绍了FPGA的原理，以及硬件加速的原理，便于软件工程师理解FPGA。在第9章中通过基础实验带领大家浅尝Zynq的开发过程，并带领大家用ZedBoard板搭建出一个单板计算机系统。

  进阶篇中，首先介绍了利用虚拟平台QEMU进行软件开发的流程。在第11章中详细分析了Zynq中处理器与硬件逻辑的连接关系。第12章主要介绍了在Zynq上利用AutoESL进行软硬件协同设计的理念与过程。第13章占了全书四分之一的篇幅，介绍了使用Zynq实现逻辑、驱动、操作系统、高清视频显示、OpenCV、网络摄像机等12个应用案例，这是本书的精华之一。在第14章，将13章中提到的独立案例再融合成4个系统案例，这4个系统案例都会应用到前面章节的知识，以此帮助大家尽快掌握在Zynq上进行系统级设计的方法。

Zynq-7000 网络资源

- Xilinx官网
  - http://china.xilinx.com/support/
  - http://wiki.xilinx.com/
  - http://forums.xilinx.com
- ARM
  - http://infocenter.arm.com/help/index.jsp

Thank you☺