고려대학교
KOREA UNIVERSITY

Computer Systems

Zynq & Zedboard

Prof. Jaeweon Suh

Computer Science Education

Korea University

ARM in FPGAs

- Xilinx Zynq (~2012)
  - Zynq-7000 All Programmable SoCs with Cortex-A9 MPCore

- Altera Arria V & Cyclone V (~2012)
  - Hard processor system (HPS) with Cortex-A9 MPCore

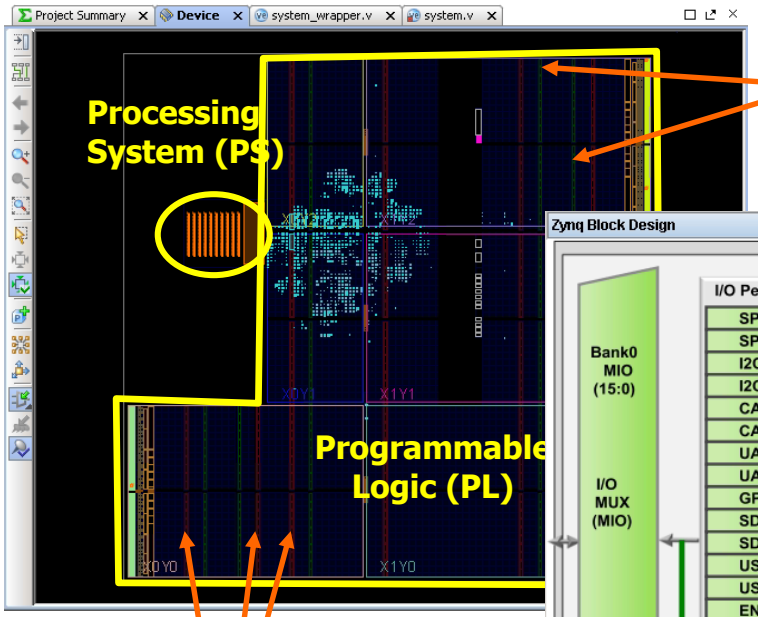
- Actel Smartfusion (~2010)
  - Cortex M3

Zynq-7000 Device Portfolio Summary

	Zynq-7000 AP SoC Devices	7Z010	7Z020	7Z030	7Z045	7Z100 new
Processing System	Processor Core	Dual ARM® Cortex™-A9 MPCore™				
	Processor Extensions	NEON™ & Single / Double Precision Floating Point				
	Memory	L1 Cache 32KB I / D, L2 Cache 512KB, on-chip Memory 256KB				
	External Memory Support	DDR3, DDR2, LPDDR2, 2x QSPI, NAND, NOR				
	Peripherals	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 2x SD/SDIO, 2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b GPIO				
Programmable Logic	Logic Cells (Slices)	30K LC	85K LC	125K LC	350K LC	444K LC
	Block RAM	60	140	265	545	755
	DSP Blocks	80	220	400	900	2020
	PCI Express® (Root Complex or Endpoint)	-		Gen2 x4	Gen2 x8	
	Agile Mixed Signal (XADC)	2x 12bit 1Msps A/D Converter				
I/O	Processor System IO	130				
	Multi Standards 3.3V IO	100	200	100	250	250
	Multi Standards High Performance 1.8V IO	-	-	150	150	150
	Multi Gigabit Transceivers	-	-	4	16	16

- Block RAM: dual-port, 36Kbit (can be configured 32K x 1, 16K x 2, 8K x 4, 4K x 9 (or 8), 1K x 36 (or 32), or 512 x 72 (or 64)
- 140 Block RAMs = 140 x 36Kb = 630KB

Zynq

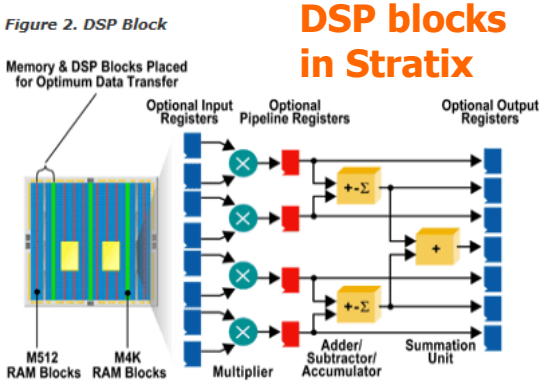


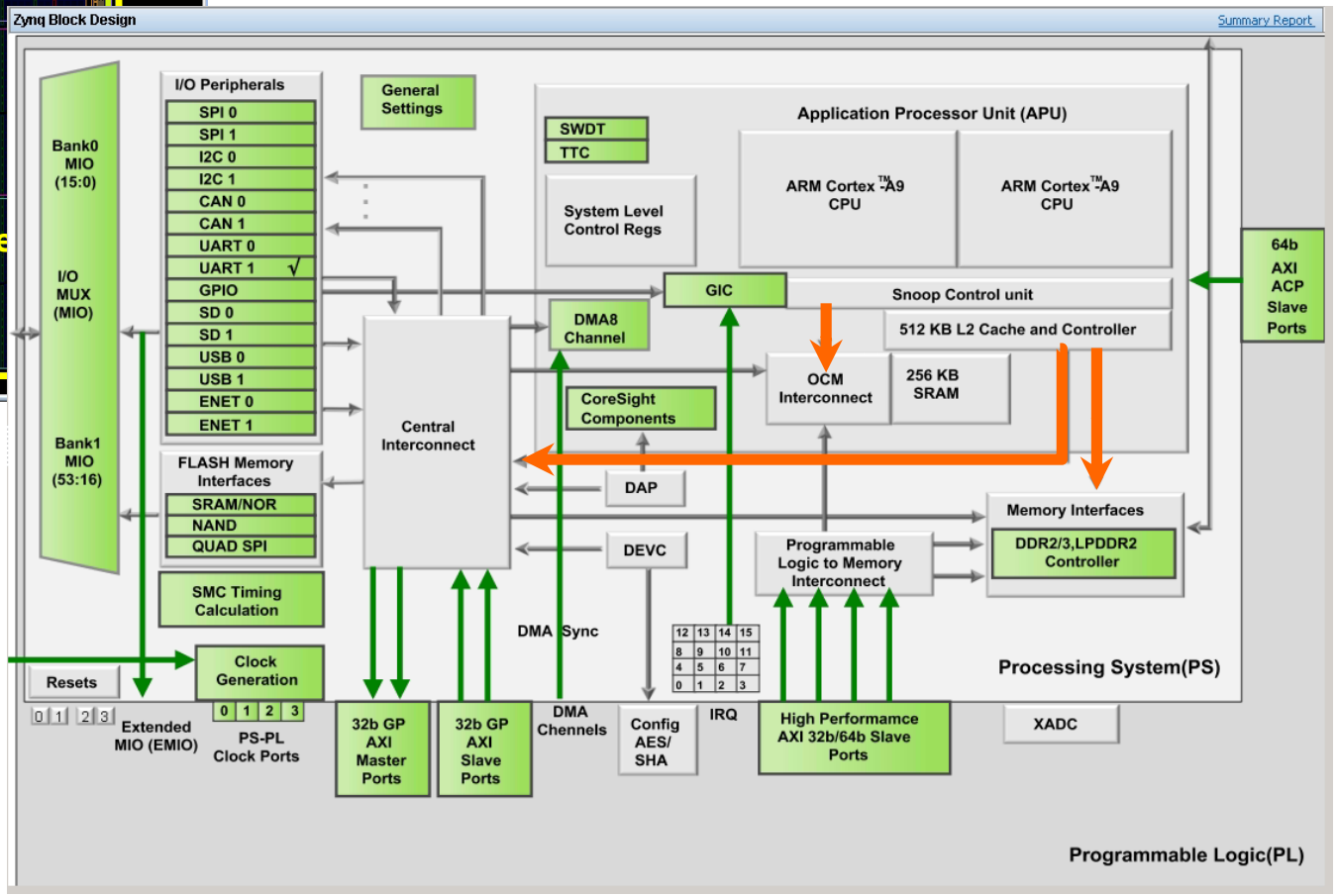
DSP (Digital Signal Processing) Blocks in green

Block RAMs in purple

XC7Z020-1CLG484C

DSP blocks in Stratix



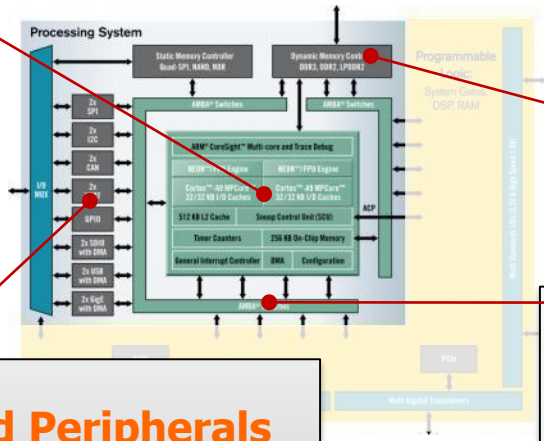


Korea Univ

Processing System (PS)

Processor Core Complex

- Dual ARM Cortex-A9 MPCore with NEON™ extensions
- Single / Double Precision Floating Point support
- Up to 1 GHz operation



High BW Memory

- Internal
  - L1 Cache – 32KB/32KB (per Core)
  - L2 Cache – 512KB Unified
- On-Chip Memory of 256KB
- Integrated Memory Controllers (DDR3, DDR2, LPDDR2, 2xQSPI, NOR, NAND Flash)

Integrated Memory Mapped Peripherals

- 2x USB 2.0 (OTG) w/DMA
- 2x Tri-mode Gigabit Ethernet w/DMA
- 2x SD/SDIO w/DMA
- 2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 32b GPIO

AMBA Open Standard Interconnect

- High bandwidth interconnect between Processing System and Programmable Logic
- ACP port for enhanced hardware acceleration and cache coherency for additional soft processors

Address Map

Table 4-1: **System-Level Address Map**

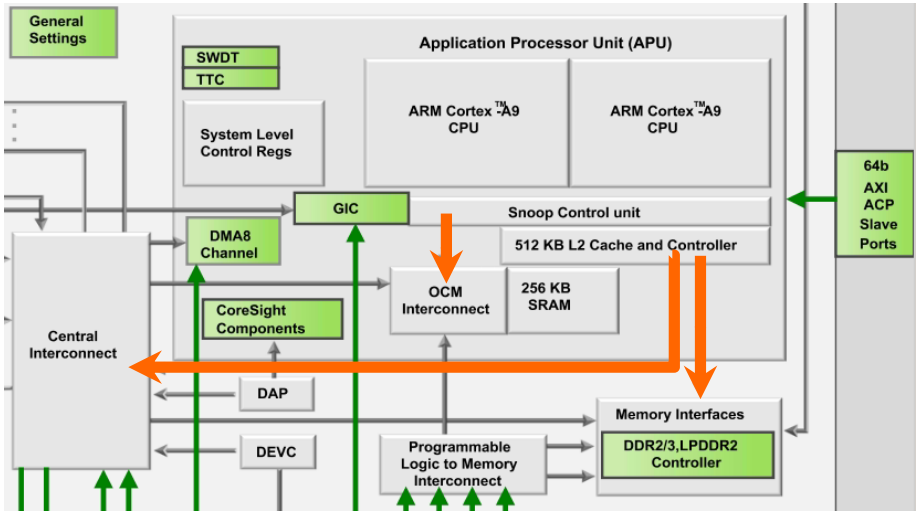
Address Range	CPUs and ACP	AXI_HP	Other Bus Masters[1]	Notes
0000_0000 to 0003_FFFF[2]	OCM	OCM	OCM	Address not filtered by SCU and OCM is mapped low
	DDR	OCM	OCM	Address filtered by SCU and OCM is mapped low
	DDR			Address filtered by SCU and OCM is not mapped low
				Address not filtered by SCU and OCM is not mapped low
0004_0000 to 0007_FFFF	DDR			Address filtered by SCU
				Address not filtered by SCU
0008_0000 to 000F_FFFF	DDR	DDR	DDR	Address filtered by SCU
		DDR	DDR	Address not filtered by SCU[3]
0010_0000 to 3FFF_FFFF	DDR	DDR	DDR	Accessible to all interconnect masters
4000_0000 to 7FFF_FFFF	PL		PL	General Purpose Port #0 to the PL, M_AXI_GP0
8000_0000 to BFFF_FFFF	PL		PL	General Purpose Port #1 to the PL, M_AXI_GP1
E000_0000 to E02F_FFFF	IOP		IOP	I/O Peripheral registers, see Table 4-6
E100_0000 to E5FF_FFFF	SMC		SMC	SMC Memories, see Table 4-5
F800_0000 to F800_0BFF	SLCR		SLCR	SLCR registers, see Table 4-3
F800_1000 to F880_FFFF	PS		PS	PS System registers, see Table 4-7
F890_0000 to F8F0_2FFF	CPU			CPU Private registers, see Table 4-4
FC00_0000 to FDFF_FFFF[4]	Quad-SPI		Quad-SPI	Quad-SPI linear address for linear mode
FFFC_0000 to FFFF_FFFF[2]	OCM	OCM	OCM	OCM is mapped high
				OCM is not mapped high

Table 4-6: **I/O Peripheral Register Map**

Register Base Address	Description
E000_0000, E000_1000	UART Controllers 0, 1
E000_2000, E000_3000	USB Controllers 0, 1
E000_4000, E000_5000	I2C Controllers 0, 1
E000_6000, E000_7000	SPI Controllers 0, 1
E000_8000, E000_9000	CAN Controllers 0, 1
E000_A000	GPIO Controller
E000_B000, E000_C000	Ethernet Controllers 0, 1
E000_D000	Quad-SPI Controller
E000_E000	Static Memory Controller (SMC)
E010_0000, E010_1000	SDIO Controllers 0, 1

Application Processing Unit (APU)

- Cortex-A9 MPCore
  - ARMv7
  - NEON 128b SIMD coprocessor
  - 32KB L1I$ and 32KB L1D$
  - 512KB shared L2
- Snoop Control Unit (SCU)
  - Maintains coherency between L1Ds and L2

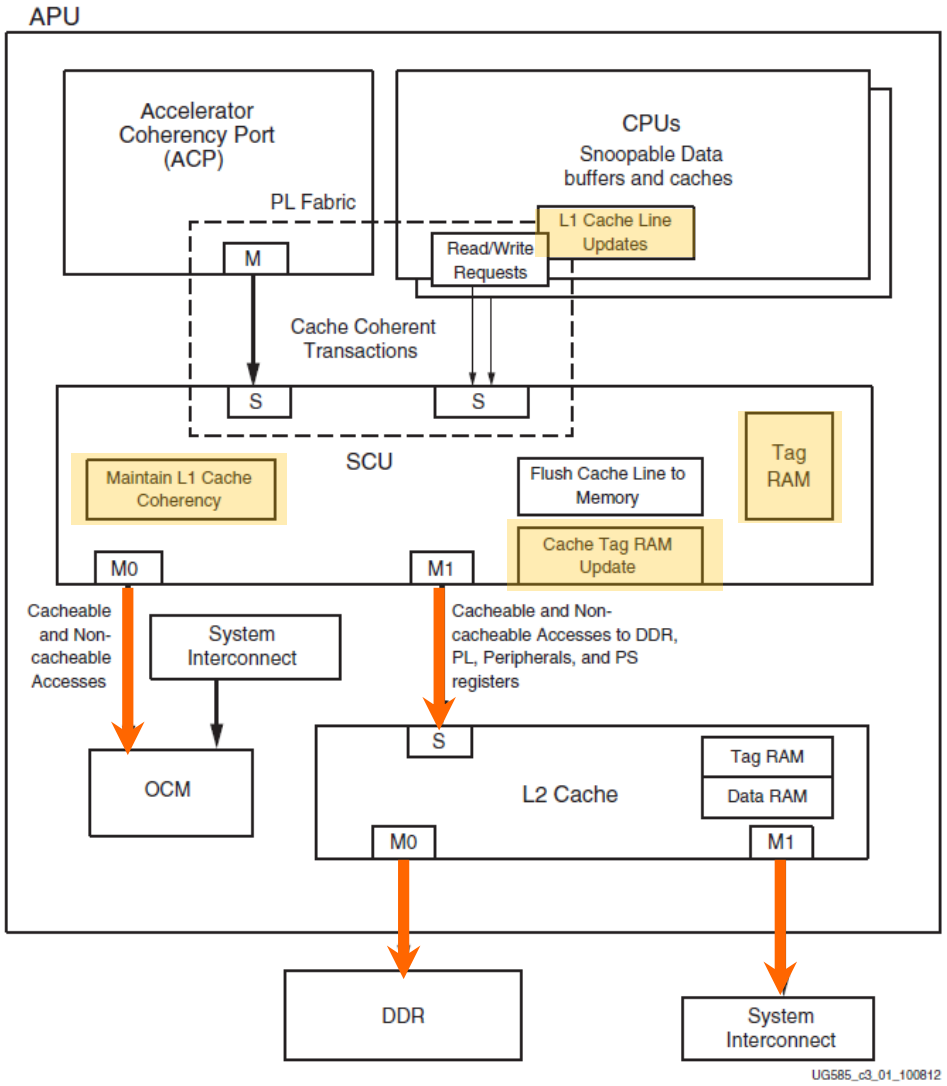


- On-Chip Memory (OCM)
  - 256KB On-Chip SRAM, accessible by CPUs, PL, and central interconnect
  - 128KB BootROM, not user-visible (Zynq-7000 EPP TRM, p678)
- Accelerator Coherency Port (ACP)
  - From PL (master) to PS (slave)
  - Can access the L2 and the OCM
  - Transactions are data coherent with L1 and L2 $

- DAP: Debug Access Port
- GIC: Generic Interrupt Controller

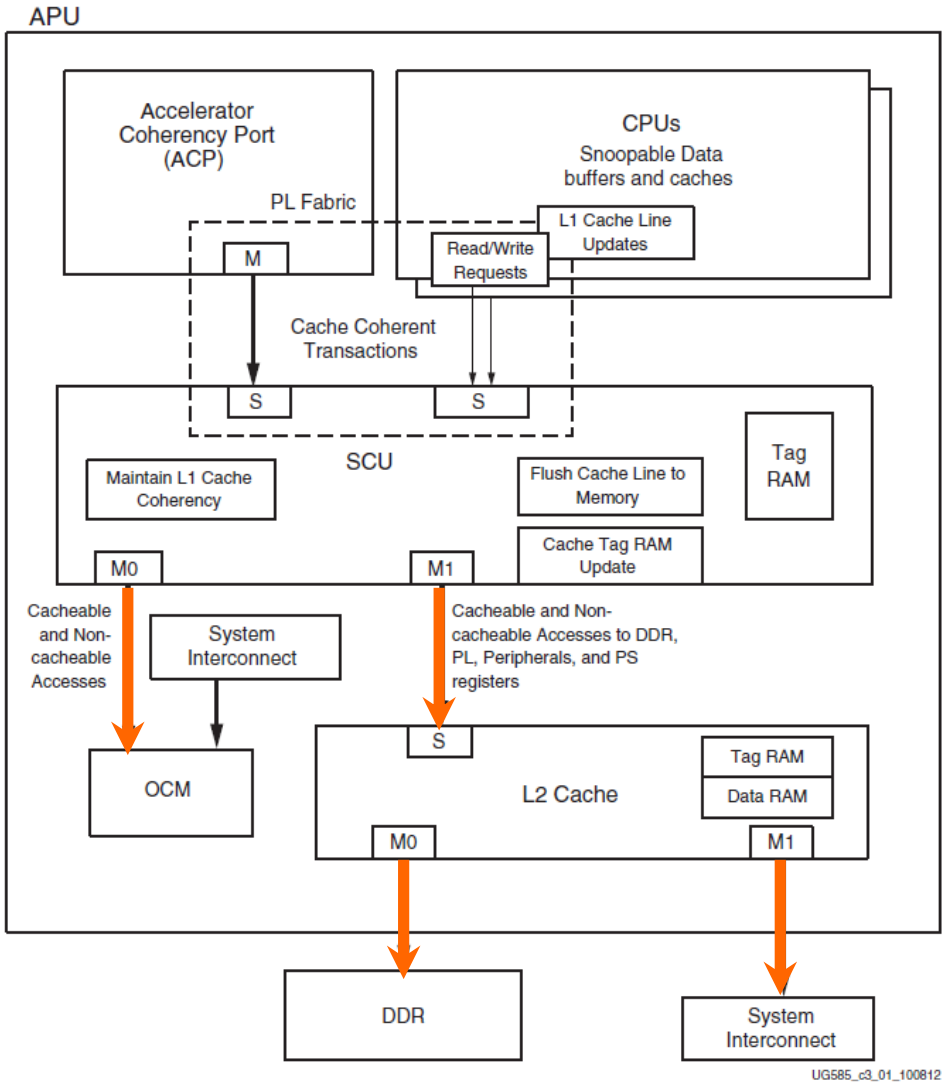
SCU

- Snoop Control Unit (SCU) is responsible for maintaining L1 cache coherency between 2 processors
  - SCU supports the MESI snooping
  - It has 4-way associative Tag RAMs, acting as a local directory that lists coherent cache lines held in L1D$
  - The local directory allows the SCU to check if data is in the L1D$, so accesses can be filtered only to the processor that is sharing the data
  - L1D$s are 4-way set associate with write-back/write-allocation and MESI



SCU

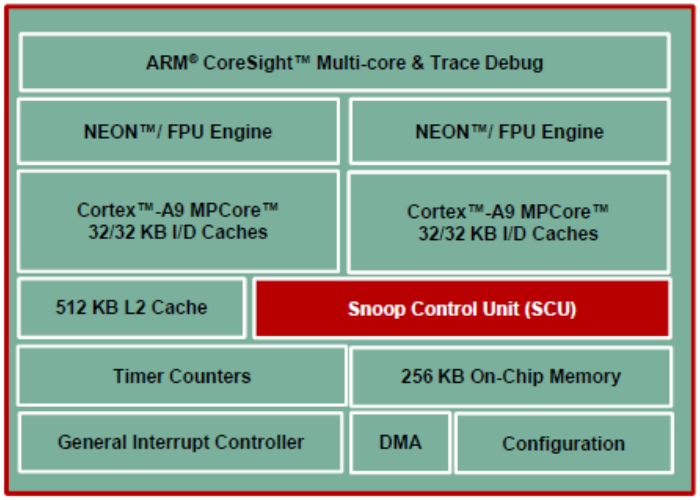
- All accesses through L2 cache controller can be routed to the DDR or other slaves in the PS or PL, depending on their addresses

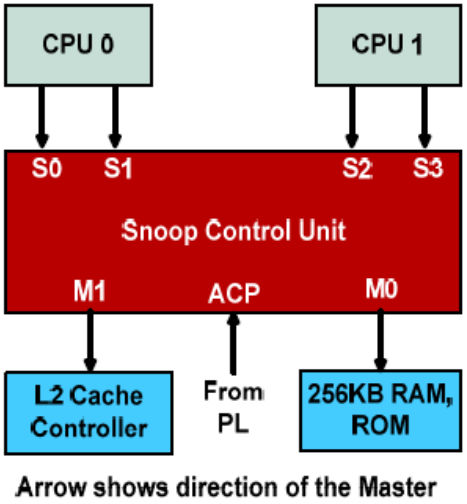


SCU

Snoop Control Unit (SCU)

- **The Snoop Control Unit (SCU) connects two Cortex-A9 processors to the system memory/peripherals via AXI interfaces in Zynq-7000 EPP**
  - Provides L1 data cache coherency between the Cortex-A9 processors
  - Arbitrates between Cortex-A9 processors requesting L2 cache accesses
  - Provides access to the on-chip ROM and RAM
  - Manages Accelerator Coherence Port (ACP) accesses





Arrow shows direction of the Master

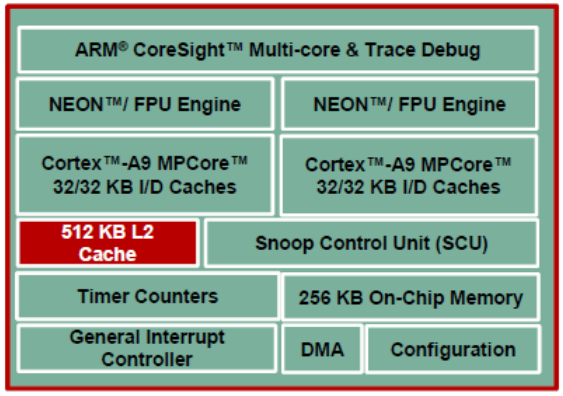
L2 Cache Controller

2012 Xfest

L2 Cache Controller

13

- **High performance L2 cache controller**
  - 512KB of cache
  - Supports lockdown by line for routines that might need low latency
  - 8-way set associative
  - Parity support
  - Write-through and write-back support



- **L2 cache controller AXI interfaces**
  - One AXI master interface for the DDR controller
  - One AXI master interface for all slave devices in the PL and PS
  - One AXI slave interface for the Snoop Control Unit

Follow @avnetxfest | Tweet this event: #avtxfest

AVNET

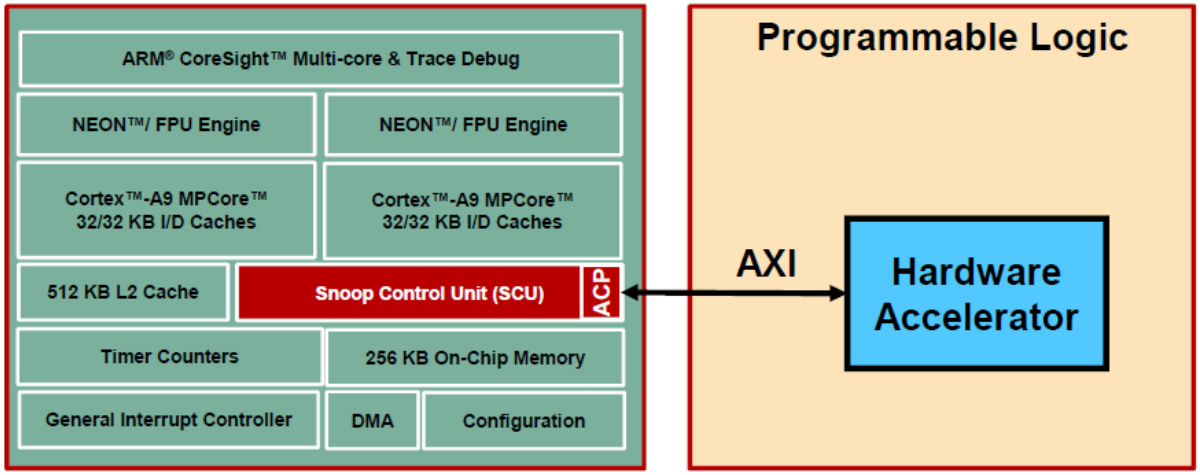
Korea Univ

ACP

2012 Xfest

Accelerating Software with Hardware

- **Accelerator Coherence Port (ACP)**
  - The Accelerator Coherence Port is a slave AXI 64-bit port on the SCU connected to the PL (150MHz in a -1 device)
  - ACP port can be used by accelerators in the PL to access the ARM L1/L2 caches and maintain coherency
  - Cache accesses could be used as a means to share data between the A9 CPUs and hardware accelerator in the PL with low latency



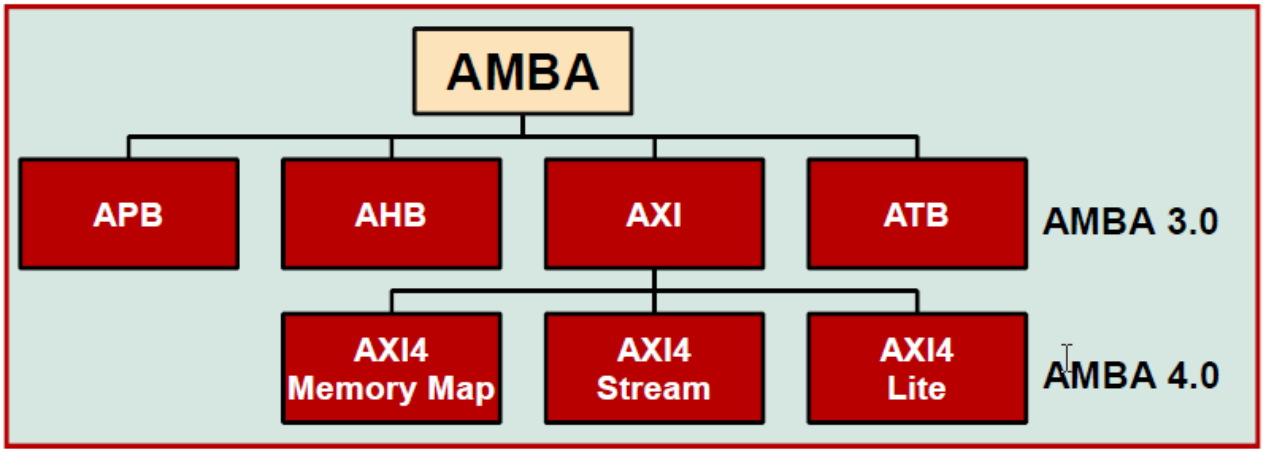
ACP allows PL masters to share and access processor's cache hierarchy

AMBA

2012 Xfest

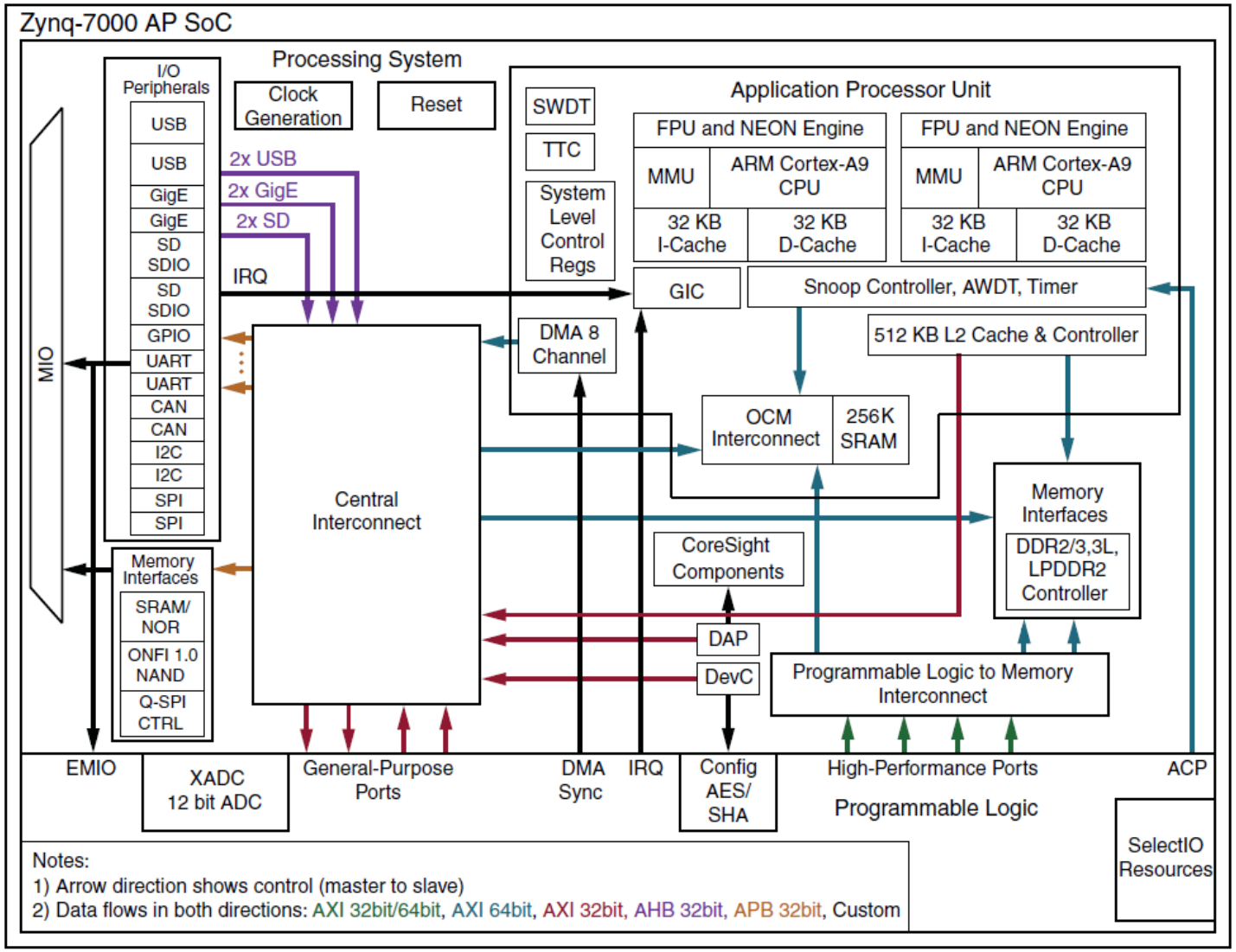
Introduction to AMBA®

- **The Advanced Microcontroller Bus Architecture (AMBA) is used as the on-chip bus in system-on-a-chip (SoC) designs**
  - AMBA was introduced by ARM in 1996



AXI - Advanced eXtensible Interface
AHB - Advanced High-performance Bus
APB - Advanced Peripheral Bus
ATB - Advanced Trace Bus (CoreSight on-chip debug and trace)

AMBA Buses in Zynq

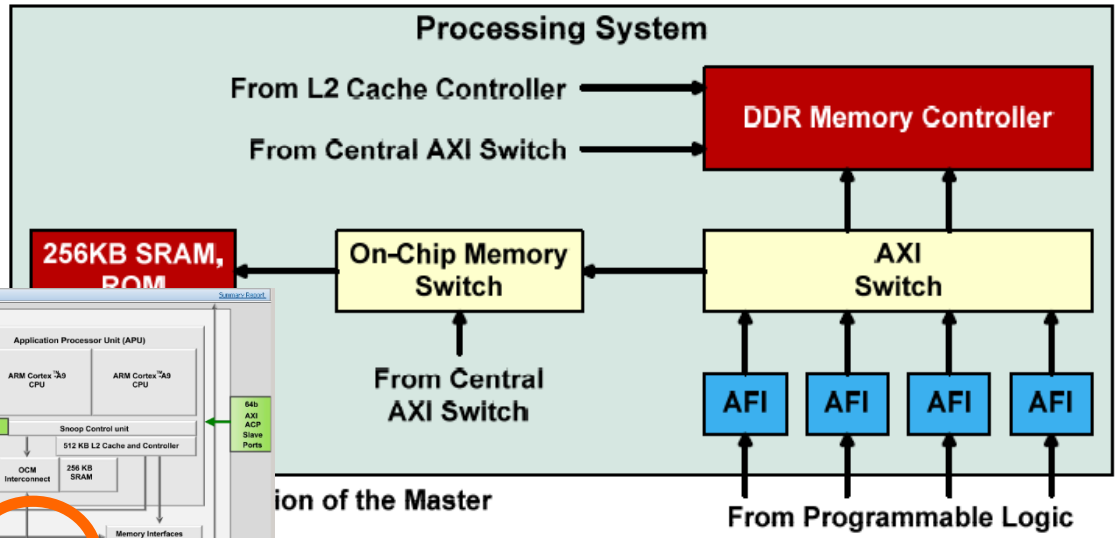


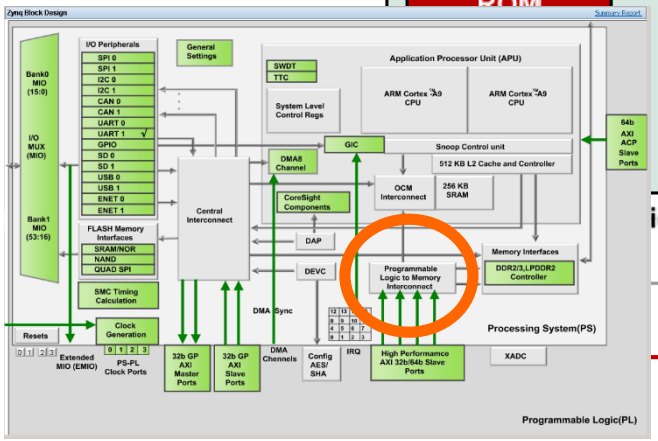
PL-to-PS AXI Ports

2012 Xfest

PL to PS High Performance AXI Ports

- **PL uses 4 high performance AXI ports to access the PS memory**
  - Each AXI port from PL can be configured for 32-bit or 64-bit operation
    - 150MHz operation in a -1 speed grade part
  - AXI FIFO Interfaces (AFI) are large (1KB, 128 x 64-bit) FIFOs to smooth large data transfers (idea is not to stall PL)
  - Clock domain crossing between the PL and PS is managed in the PS





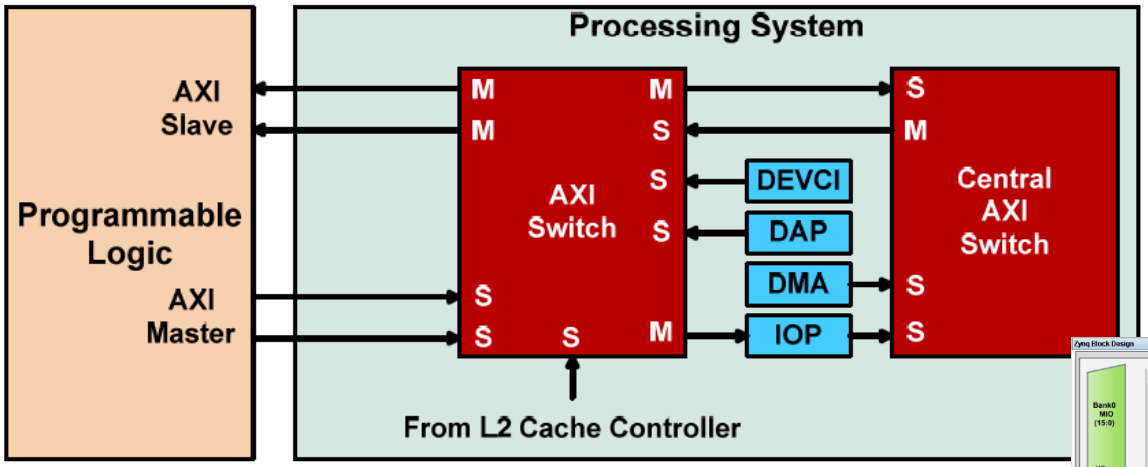
Follow @avnetxfest | Tweet this event: #avtxfest

AVNET

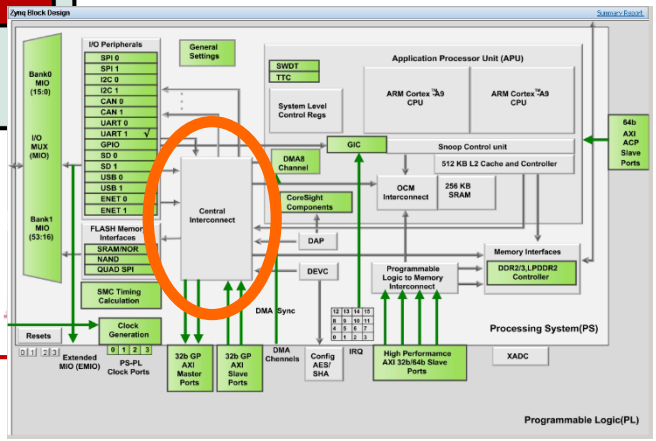
PS-to-PL AXI Ports

PS to PL General-Purpose AXI Ports

- **General-purpose AXI ports**
  - 2 masters from PL to PS
  - 2 masters from PS to PL
  - All general-purpose AXI ports are 32 bits wide, 150MHz in a -1 device
  - Clock domain crossing between the PL and PS is managed in the PS



1) IOP – IO Peripherals
2) DEVCI – Device Configuration Interface



DMA

- DMA transfers use AXI to move data between the on-chip memory, DDR and slave peripherals in PL
- Typical DMA examples
  - Memory to memory
    - On-chip memory to DDR
  - Memory to/from PL peripheral
    - DDR to PL peripheral

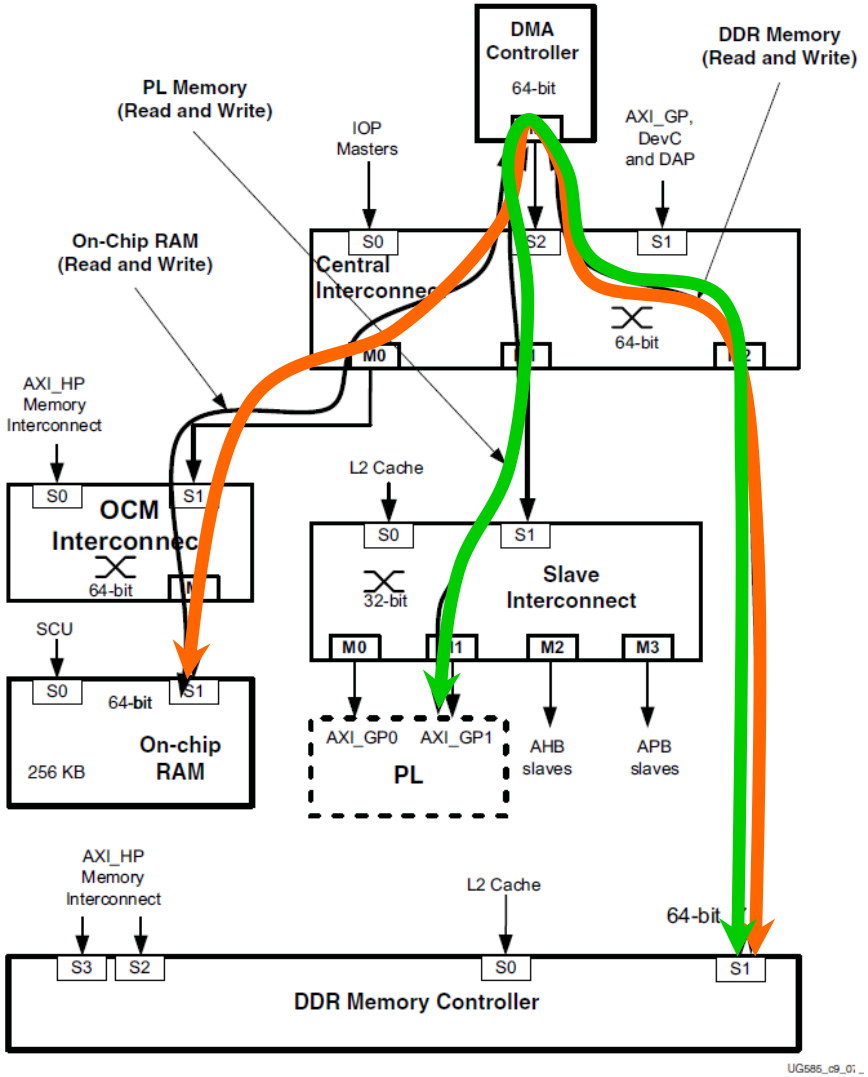
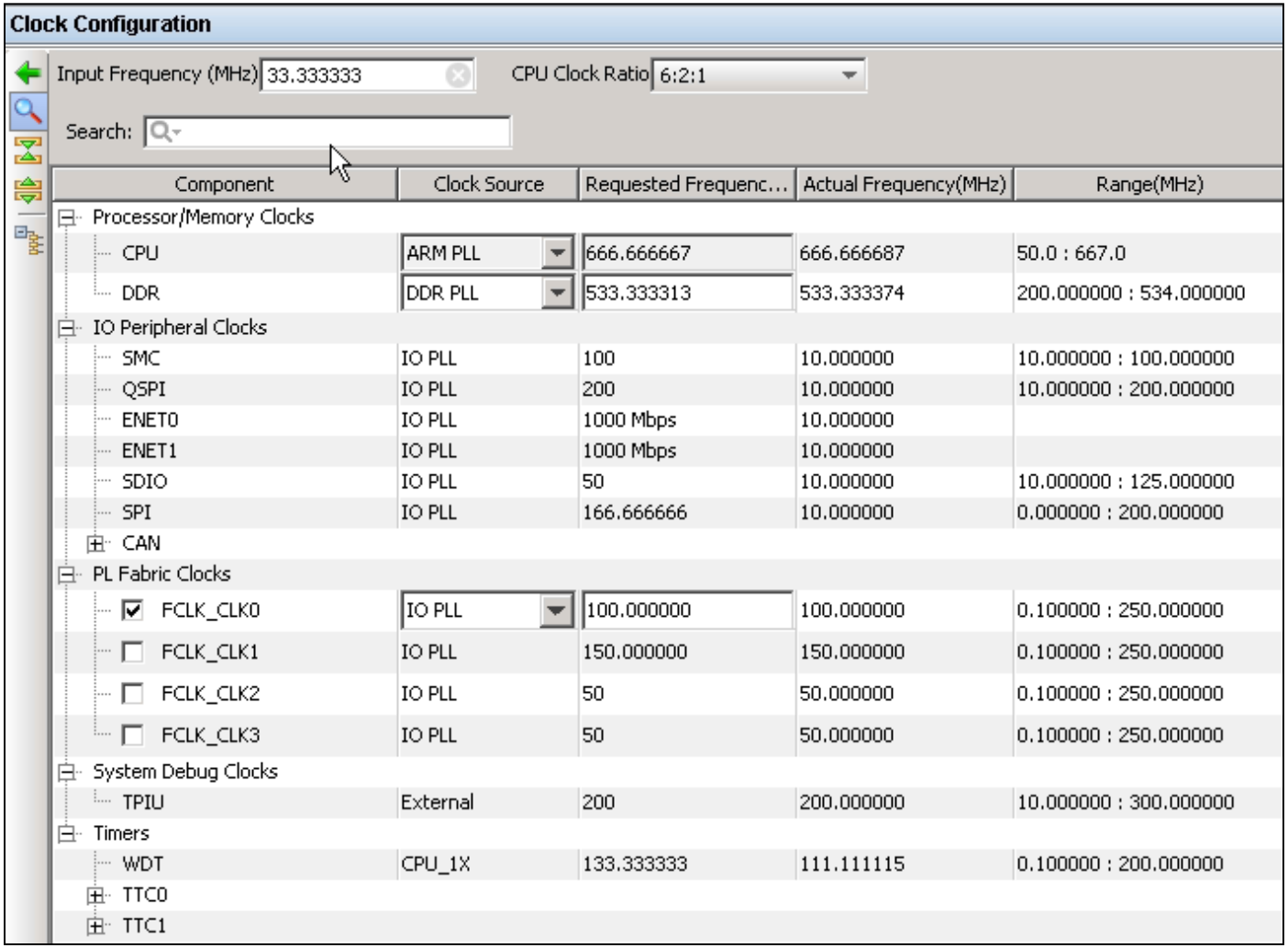


Figure 9-3: **DMAC Reads/Writes DDR, On-chip RAM, and PL Peripheral**

PS Clock Configuration



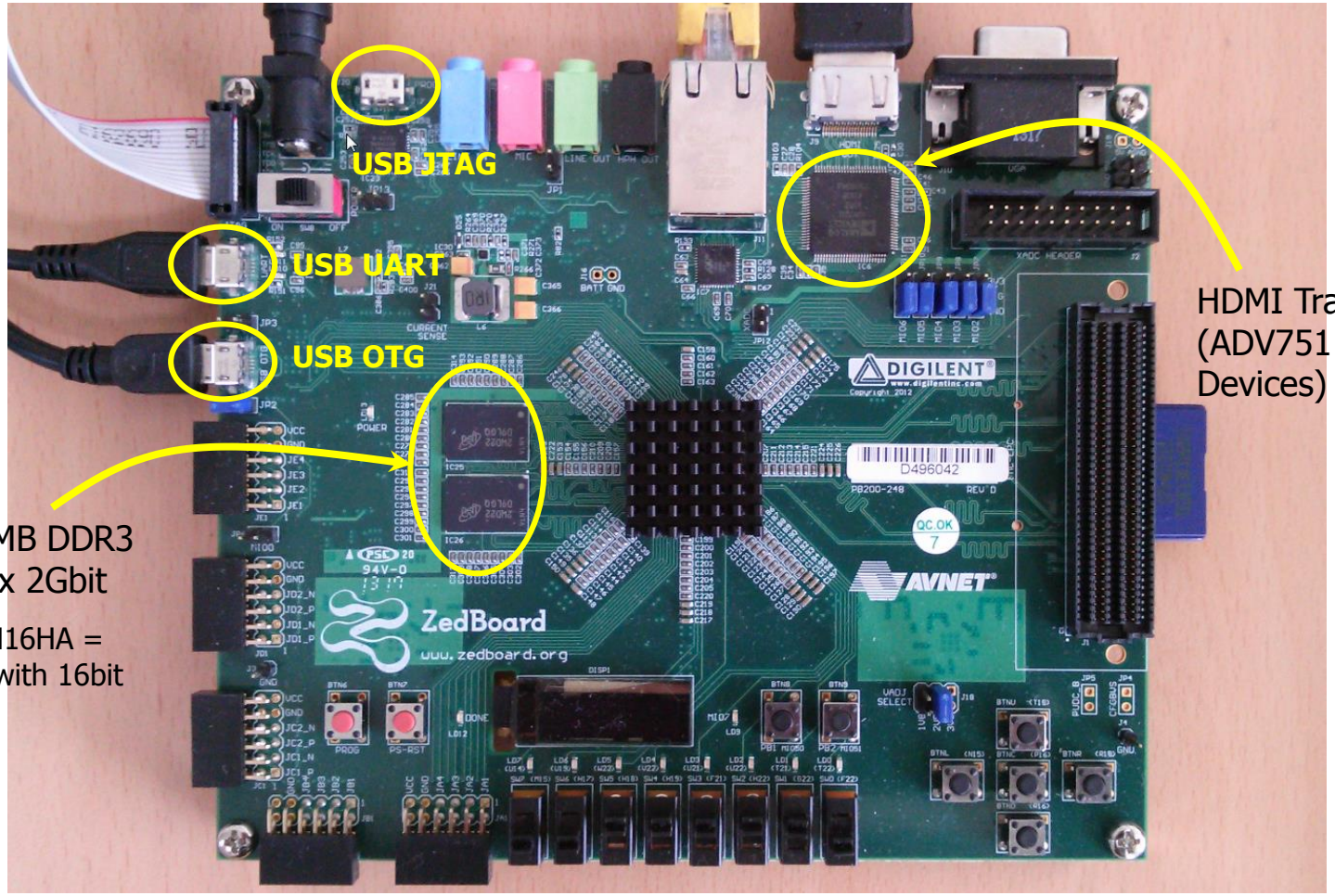
- The PS can generate up to **4 PLL (Phase-Locked Loop) based clocks** for the PL

Some Notes

- When designing an SoC system with custom peripherals, be aware that Zynq pins are already assigned on Zedboard
  - Refer to the xdc file on the Zedboard documentation page at http://www.zedboard.org/documentation/1521

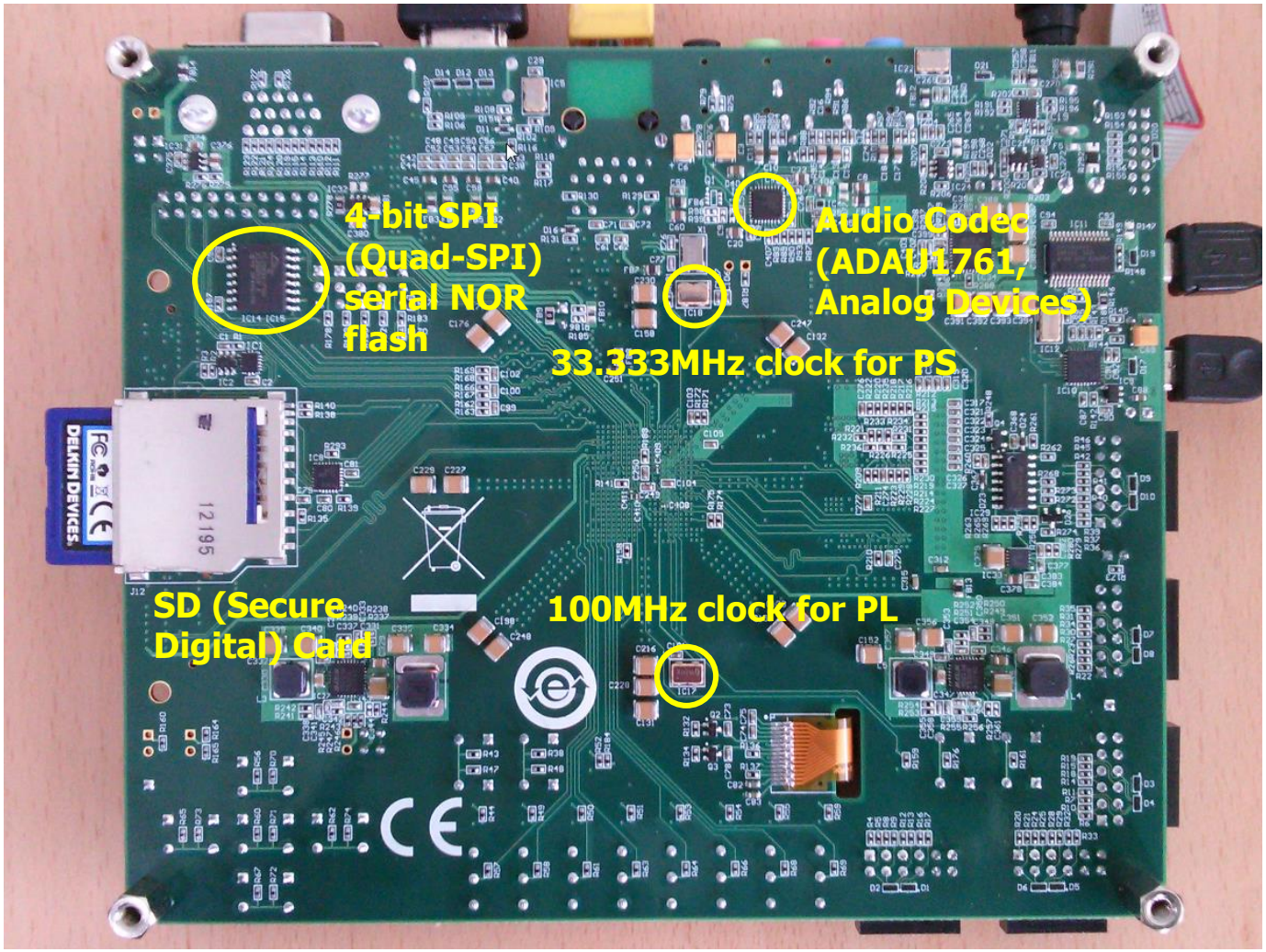
- One DRC (Design Rule Check) error case
  - Use PS clock source (33MHz) in the PL logic
  - For PL logic, you have to use either PL clock (100MHz) or PLL-generated clocks from PS (based on PS clock source)

Zedboard (Front)



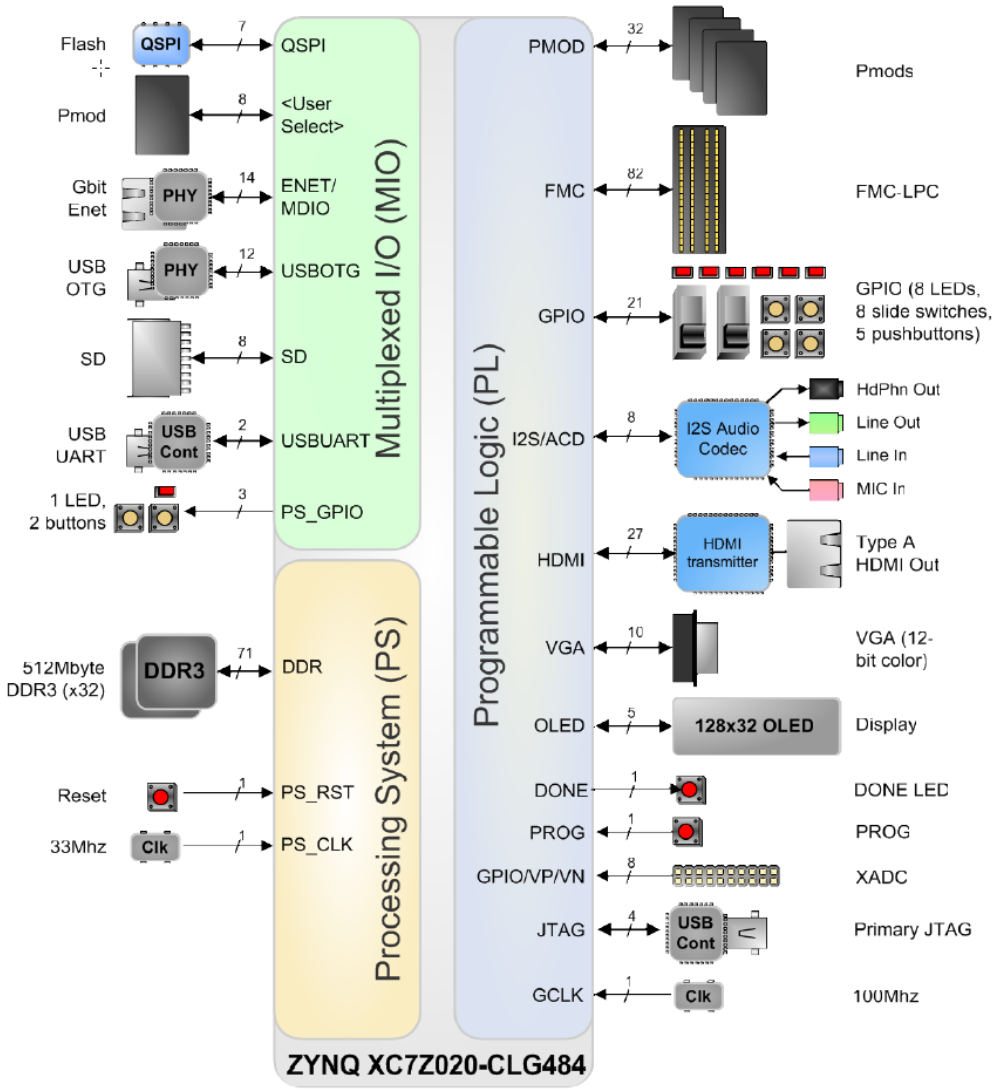
- USB OTG (On-The-Go) is a specification that allows USB devices (such as mobile phones) to act as a host, allowing other USB devices like a USB flash drive, mouse or keyboard to be attached to them
- JTAG (Joint Test Action Group) is the common name for the IEEE 1148.1 Standard Test Access Port and Boundary-Scan Architecture

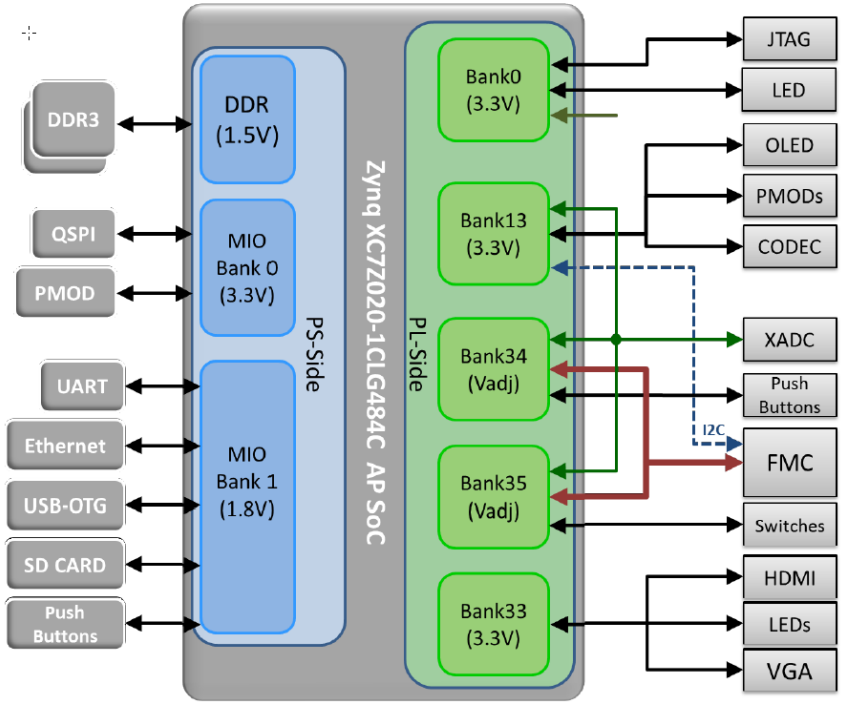
Zedboard (Back)



- The PS can generate up to 4 PLL based clocks for the PL
- SPI Flash can be used to initialize the PS subsystem and to configure the PL subsystem (bitstream)
- SPI: Serial Peripheral Interface named by Motorola

Zedboard Block Diagram





- Pmods (Peripheral Modules)
- Pmod is the trademark of Digilent Inc.
- Pmods are small I/O interface boards that offer an ideal way to extend the capabilities of FPGA/CPLD and embedded control boards

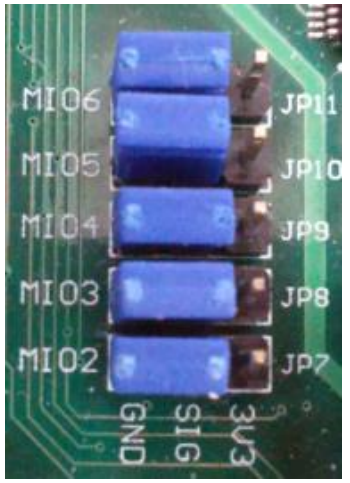
http://www.digilentinc.com/Pmods/licensing.cfm

FMC (FPGA Mezzanine Card)

Zedboard Configuration Modes

Table 18 – ZedBoard Configuration Modes

	MIO[6]	MIO[5]	MIO[4]	MIO[3]	MIO[2]
Xilinx TRM→	Boot_Mode[4]	Boot_Mode[0]	Boot_Mode[2]	Boot_Mode[1]	Boot_Mode[3]
JTAG Mode					
Cascaded JTAG					0
Independent JTAG					1
Boot Devices					
JTAG		0	0	0	
Quad-SPI		1	0	0	
SD Card		1	1	0	
PLL Mode					
PLL Used	0				
PLL Bypassed	1				
Bank Voltages					
MIO Bank 500			3.3V		
MIO Bank 501			1.8V		



Zedboard Jumper Settings

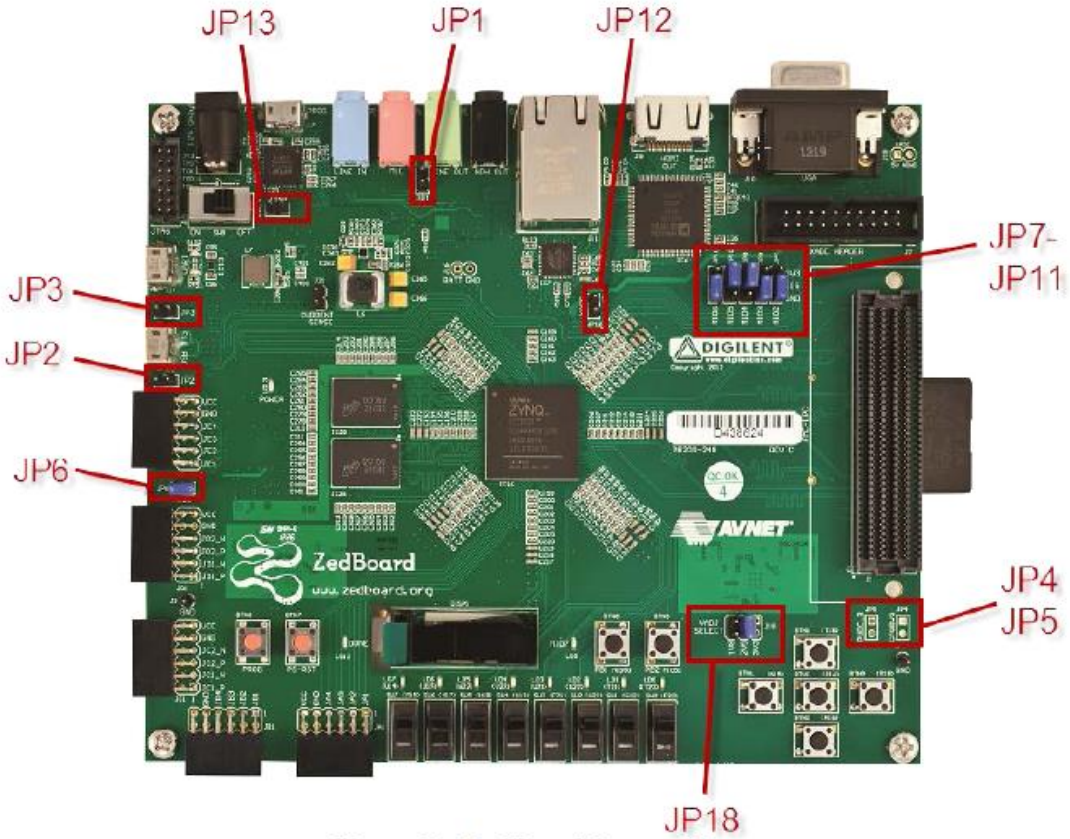


Figure 20 - ZedBoard Jumper Map

Table 22 - Jumper Settings

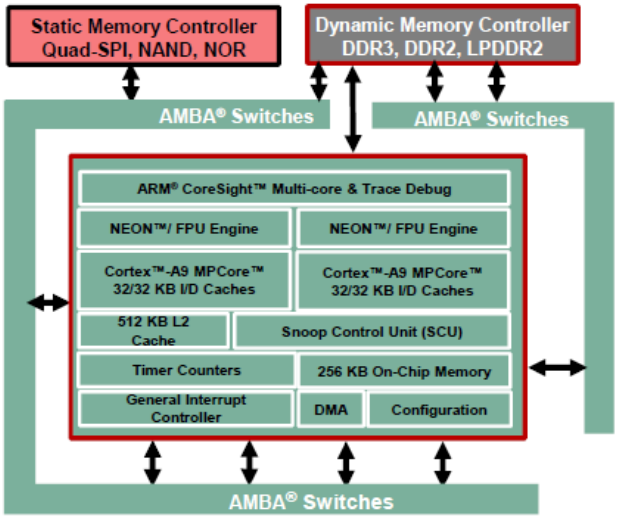
Ref Designator	Description	Default Setting	Function
JP1	Microphone Input Bias	Open – No Electret Microphone	Short to enable Bias Voltage for Electret Microphone. Right Channel only.
JP2	Vbus 5V Enable	Open – 5V Disconnected	Short to enable 5V output to USB OTG Connector, J13, for either Host or OTG modes.
JP3	USB Vbus Capacitor Setting	Open – Device Mode	Short for Host mode (>120uF). Open for Device or OTG modes (4.7uF).
JP4	CFGBVS Select	Not Populated	Pre-configuration I/O standard type for the dedicated configuration bank 0. Vcco_0 is 3.3V, Connected to 3.3V through a 10K resistor. This jumper connects to GND and should NOT be used.
JP5	PUDC Select	Not Populated	Active Low input enables internal pull-ups during configuration on all SelectIO pins. Connected to Vadj through 10K resistor.
JP6	PS_MIO0 Pull-Down	Short	Install for SD Card boot on CES silicon.
JP7	Boot_Mode[3]/MIO[2]	GND – Cascaded JTAG	JTAG Mode. GND cascades PS and PL JTAG chains. VCC makes JTAG chains independent.
JP8 JP9 JP10	Boot_Mode[0]/MIO[3] Boot_Mode[1]/MIO[4] Boot_Mode[2]/MIO[5]	110 – SD Card	Boot Device Select See Zynq Configuration Modes
JP11	Boot_Mode[4]/MIO[6]	GND – PLL Used	PLL Select. GND uses PS PLLs. VCC bypasses internal PS PLLs
JP12	XADC Ferrite Bead Disable	Open	Short bypasses XADC-GND ferrite bead connection to board GND.
JP13	JTAG PS-RST	Open	Short connects JTAG PROG-RST to PS Reset.
J18	Vadj Select	1.8V	Selects Vadj (1.8V, 2.5V, or 3.3V)

Zynq-7000 Configuration and Boot

2012 Xfest

Zynq-7000 EPP Configuration and Boot

- **CPU configures the PS and PL**
  - Standalone PL configuration (without PS configuration) is not supported
  - Configuration under external host control is also possible via JTAG
- **Two boot modes**
  - Secure boot
  - Non-secure boot
- **Four master boot methods (secure or non-secure boot)**
  - QSPI (16MB, 50MB/Sec)
  - NOR (64MB, 20MB/Sec)
  - NAND (tested up to 1GB, 10MB/Sec)
  - SD (Up to 32GB)
- **One slave boot method (non-secure)**
  - JTAG for debug and development



Follow @avnetxfest | Tweet this event: #avtxfest

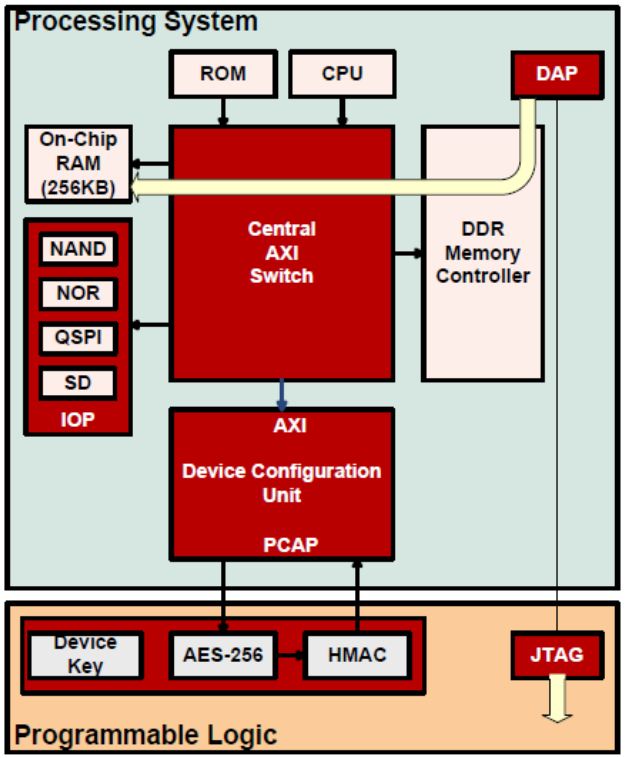
AVNET

Korea Univ

JTAG Non-Secure Boot

JTAG Non-Secure Boot (Debug/Development)

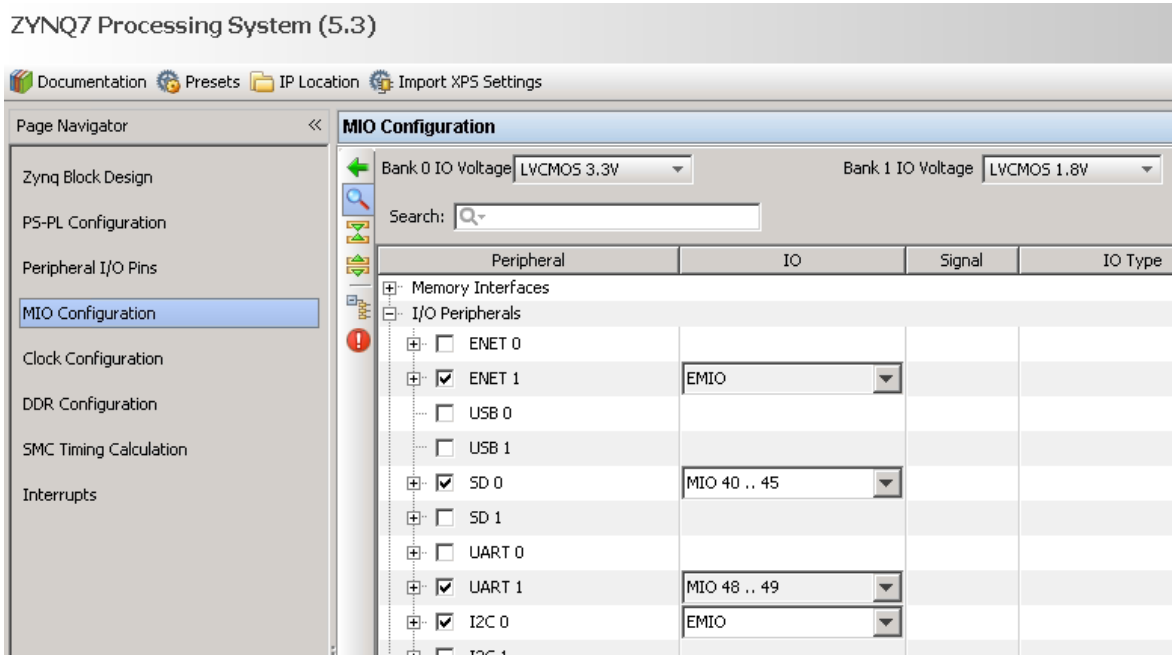
- **CPU starts executing code from ROM**
  1) Initializes the Cortex-A9 CPU 0
  2) Checks CRC on ROM code
  3) Reads the boot mode pins
  4) Disables all security features
  5) Enables DAP controller and the JTAG chain
  6) Boot ROM shuts down, releases CPU control to JTAG
  7) JTAG port is used to load the PS image into the OCM
  8) CPU starts executing code from OCM
  9) Optionally, JTAG or CPU is used to configure the PL



- DAP: Debug Access Port

GPIOs in Zynq

- 54 GPIOs routed to multiplexed IOs (MIOs)
  - Zynq PS has over 20 peripherals, which are directly routed to MIOs
- 64 GPIOs between PS and PL via Extended Multiplexed IOs (EMIOs)
  - Can be used to take input from and/or generate output to PL
  - IO peripheral signals can be routed to PL via EMIOs



MIO-EMIO Overview

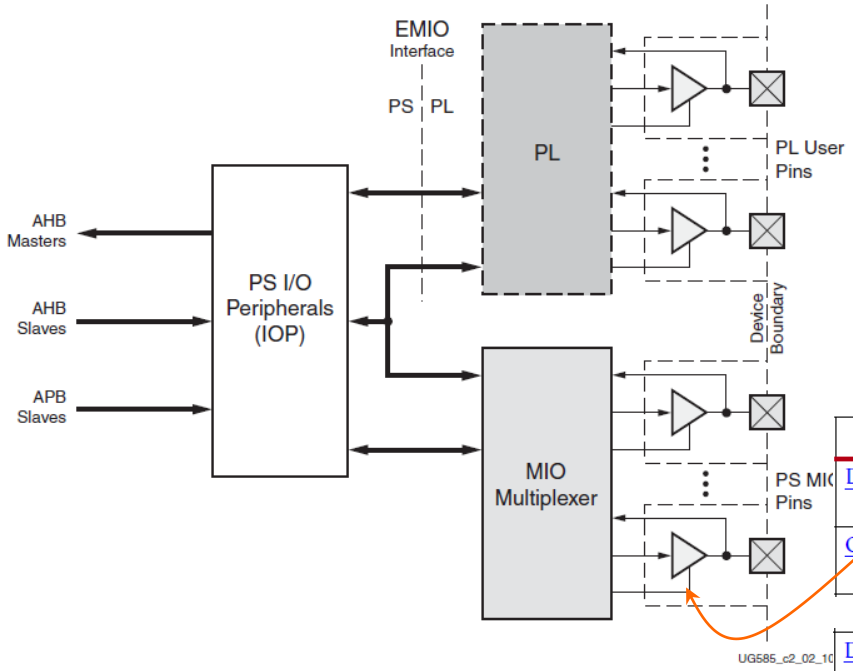
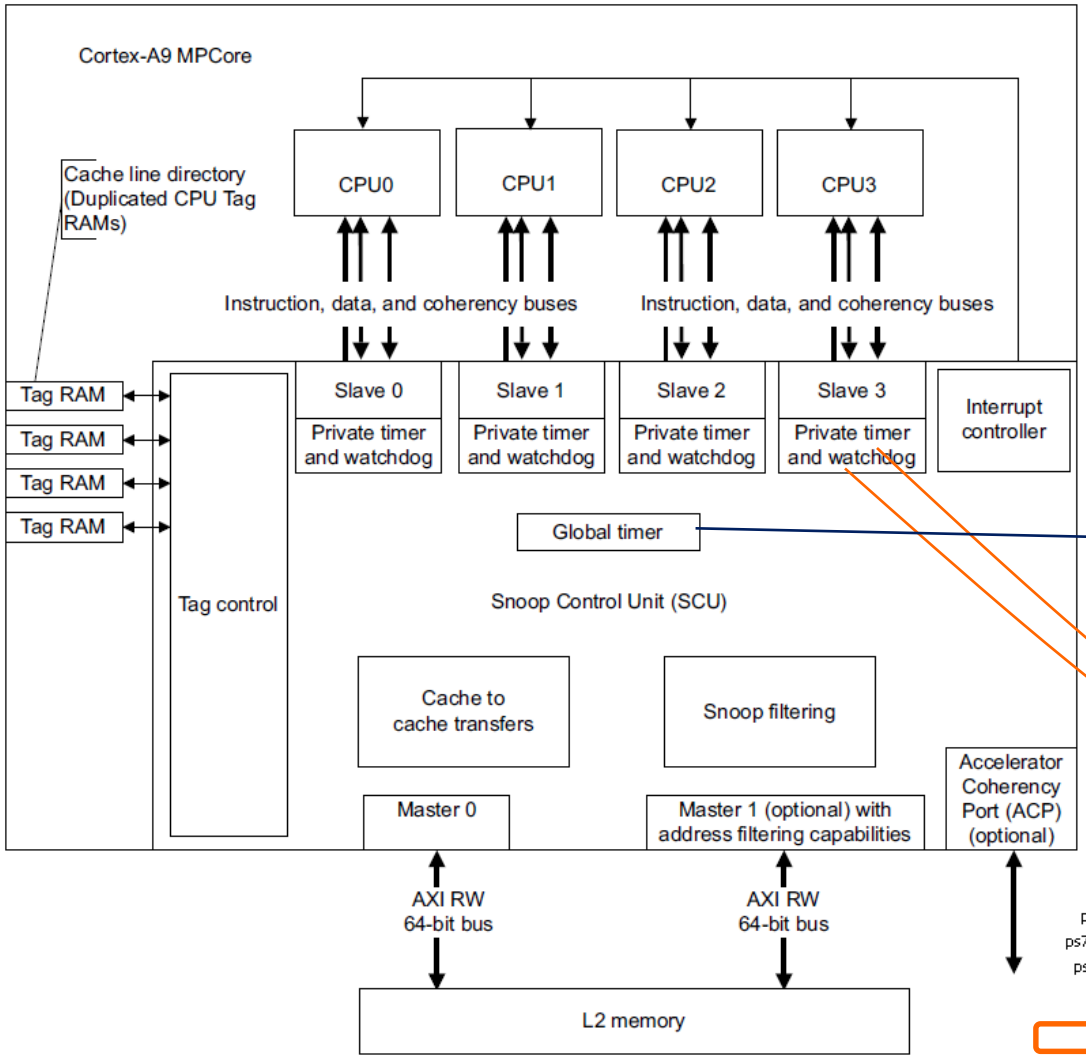


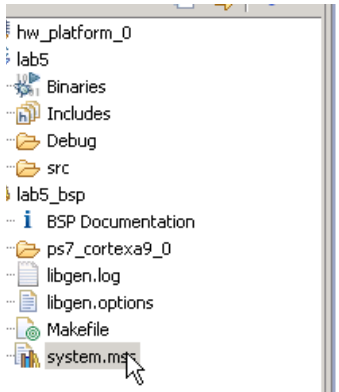
Figure 2-2: **MIO-EMIO Overview**

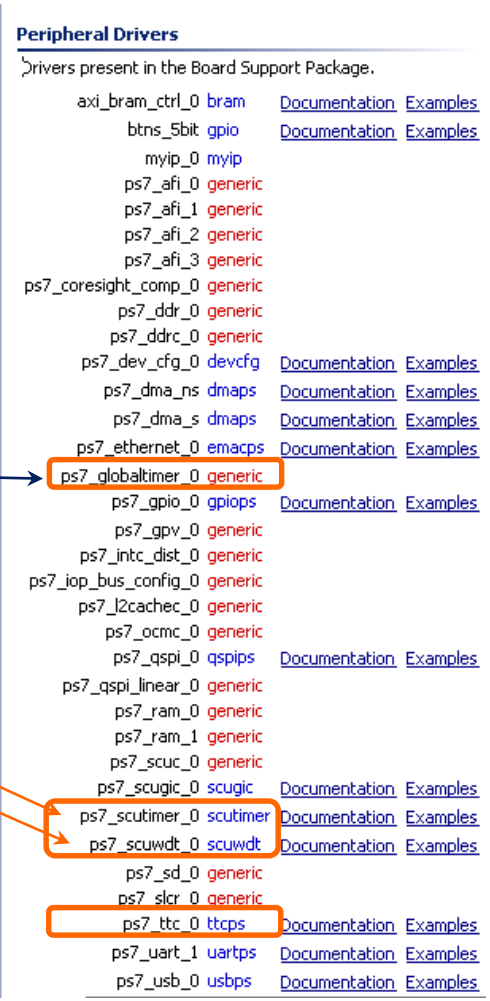
Base Address: 0xE000_A000

Register Name	Address	Width	Type	Reset Value	Description
DIRM_0	0x00000204	32	rw	0x00000000	Direction mode (GPIO Bank0, MIO)
OEN_0	0x00000208	32	rw	0x00000000	Output enable (GPIO Bank0, MIO)
DIRM_1	0x00000244	22	rw	0x00000000	Direction mode (GPIO Bank1, MIO)
OEN_1	0x00000248	22	rw	0x00000000	Output enable (GPIO Bank1, MIO)
DIRM_2	0x00000284	32	rw	0x00000000	Direction mode (GPIO Bank2, EMIO)
OEN_2	0x00000288	32	rw	0x00000000	Output enable (GPIO Bank2, EMIO)
DIRM_3	0x000002C4	32	rw	0x00000000	Direction mode (GPIO Bank3, EMIO)
OEN_3	0x000002C8	32	rw	0x00000000	Output enable (GPIO Bank3, EMIO)

Timers in Cortex-A9 MPCore







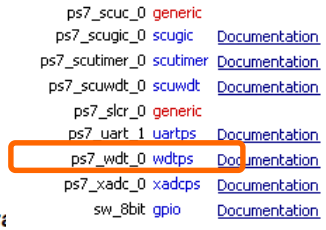
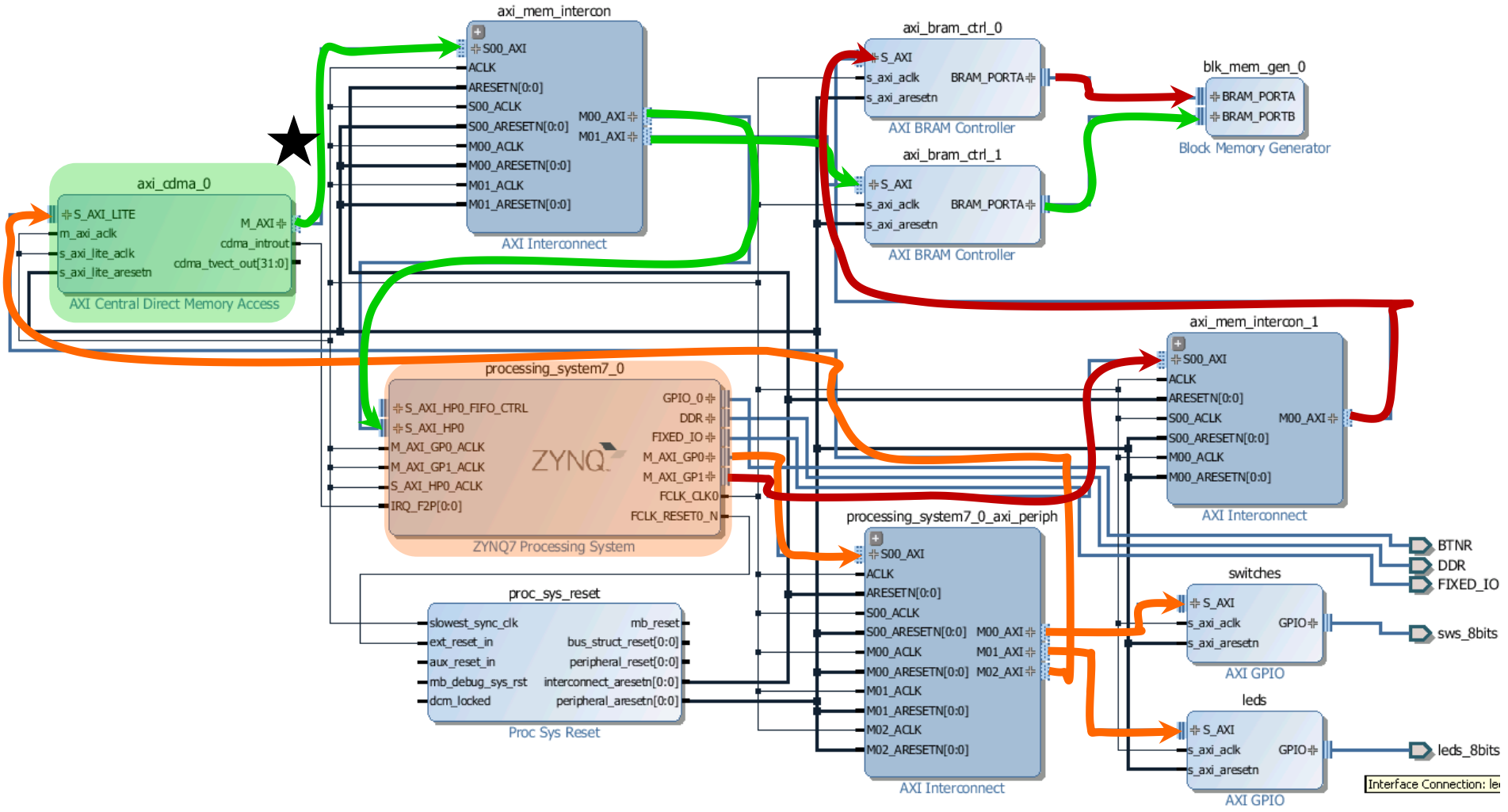
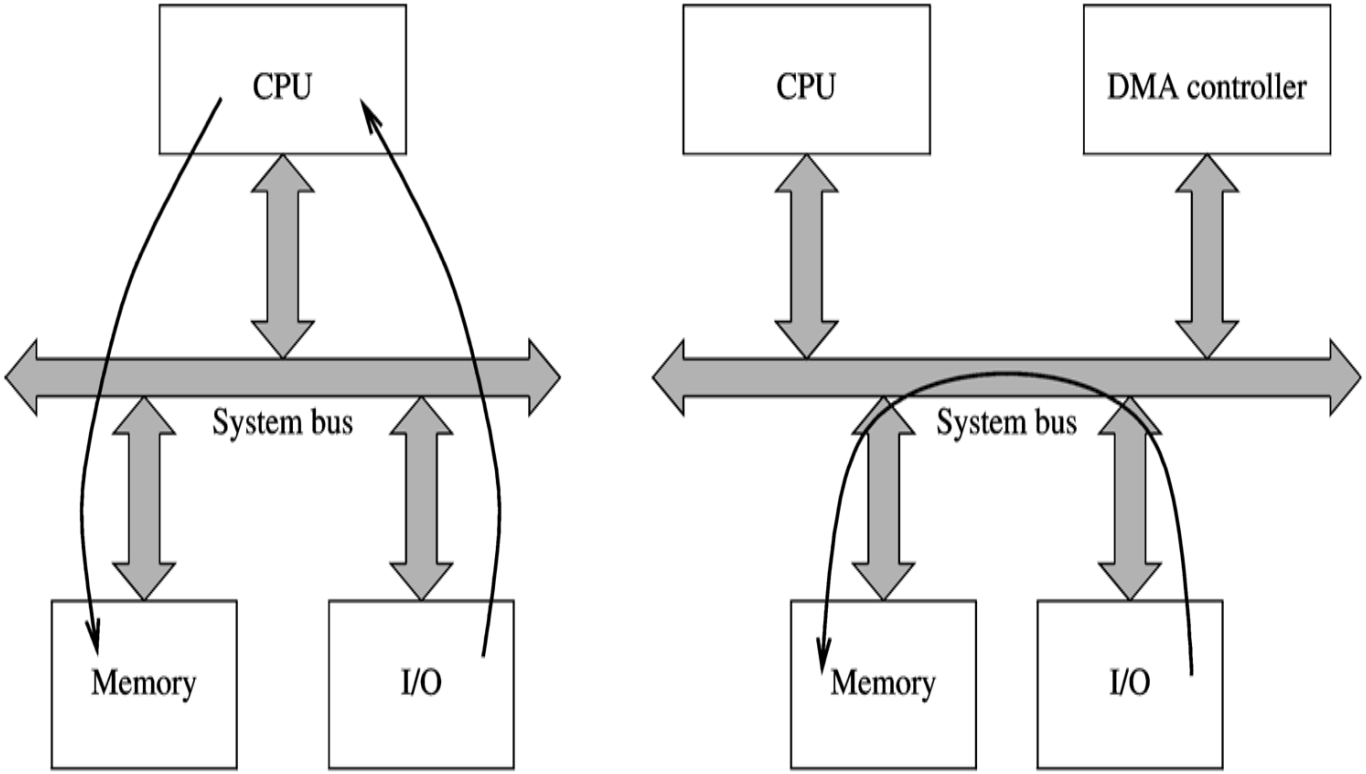


Figure 1-1 Example multiprocessor configura

Lab 6. DMA


axi_mem_intercon
S00_AXI
ACLK
ARESETN[0:0]
S00_ACLK
S00_ARESETN[0:0]
M00_ACLK
M00_ARESETN[0:0]
M01_ACLK
M01_ARESETN[0:0]
M00_AXI
M01_AXI
AXI Interconnect
axi_bram_ctrl_0
S_AXI
s_axi_aclk
s_axi_aresetn
BRAM_PORTA
AXI BRAM Controller
blk_mem_gen_0
BRAM_PORTA
BRAM_PORTB
Block Memory Generator
axi_bram_ctrl_1
S_AXI
s_axi_aclk
s_axi_aresetn
BRAM_PORTA
AXI BRAM Controller
axi_cdma_0
S_AXI_LITE
m_axi_aclk
s_axi_lite_aclk
s_axi_lite_aresetn
M_AXI
cdma_introut
cdma_tvect_out[31:0]
AXI Central Direct Memory Access
axi_mem_intercon_1
S00_AXI
ACLK
ARESETN[0:0]
S00_ACLK
S00_ARESETN[0:0]
M00_ACLK
M00_ARESETN[0:0]
M00_AXI
AXI Interconnect
processing_system7_0
S_AXI_HP0_FIFO_CTRL
S_AXI_HP0
M_AXI_GP0_ACLK
M_AXI_GP1_ACLK
S_AXI_HP0_ACLK
IRQ_F2P[0:0]
ZYNQ
GPIO_0
DDR
FIXED_IO
M_AXI_GP0
M_AXI_GP1
FCLK_CLK0
FCLK_RESET0_N
ZYNQ7 Processing System
processing_system7_0_axi_periph
S00_AXI
ACLK
ARESETN[0:0]
S00_ACLK
S00_ARESETN[0:0]
M00_ACLK
M00_ARESETN[0:0]
M01_ACLK
M01_ARESETN[0:0]
M02_ACLK
M02_ARESETN[0:0]
M00_AXI
M01_AXI
M02_AXI
AXI Interconnect
proc_sys_reset
slowest_sync_clk
ext_reset_in
aux_reset_in
mb_debug_sys_rst
dcm_locked
mb_reset
bus_struct_reset[0:0]
peripheral_reset[0:0]
interconnect_aresetn[0:0]
peripheral_aresetn[0:0]
Proc Sys Reset
BTNR
DDR
FIXED_IO
switches
S_AXI
s_axi_aclk
s_axi_aresetn
GPIO
AXI GPIO
sws_8bits
leds
S_AXI
s_axi_aclk
s_axi_aresetn
GPIO
AXI GPIO
leds_8bits
Interface Connection: le

DMA



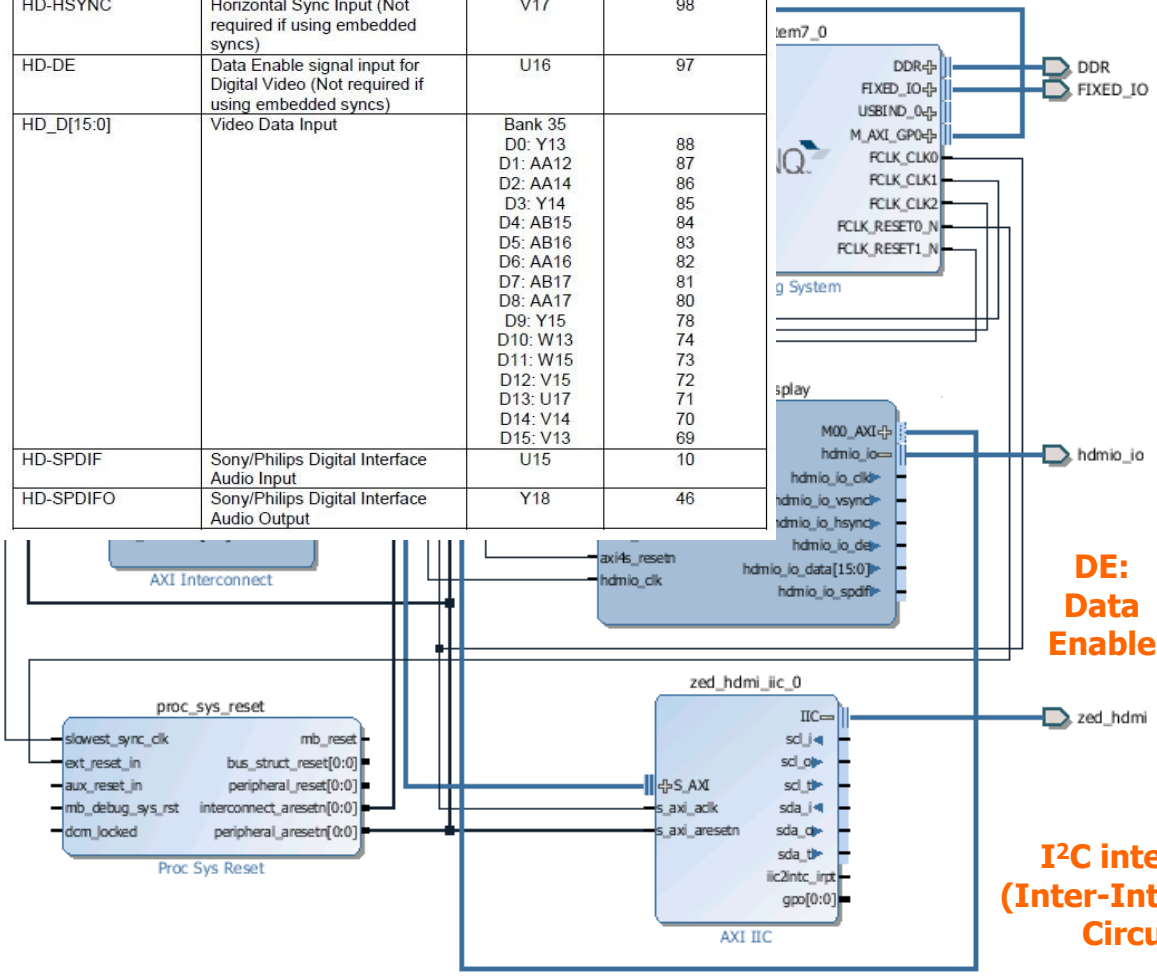
(a) Programmed I/O transfer

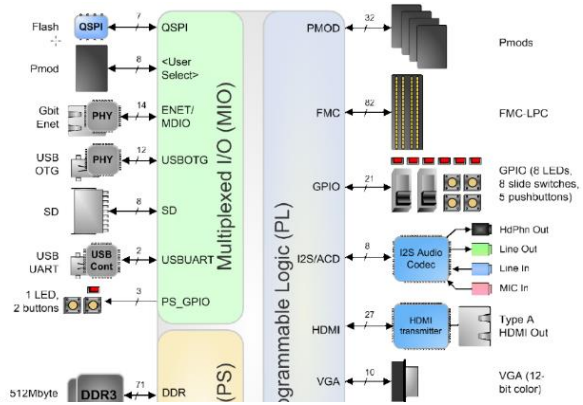
(b) DMA transfer

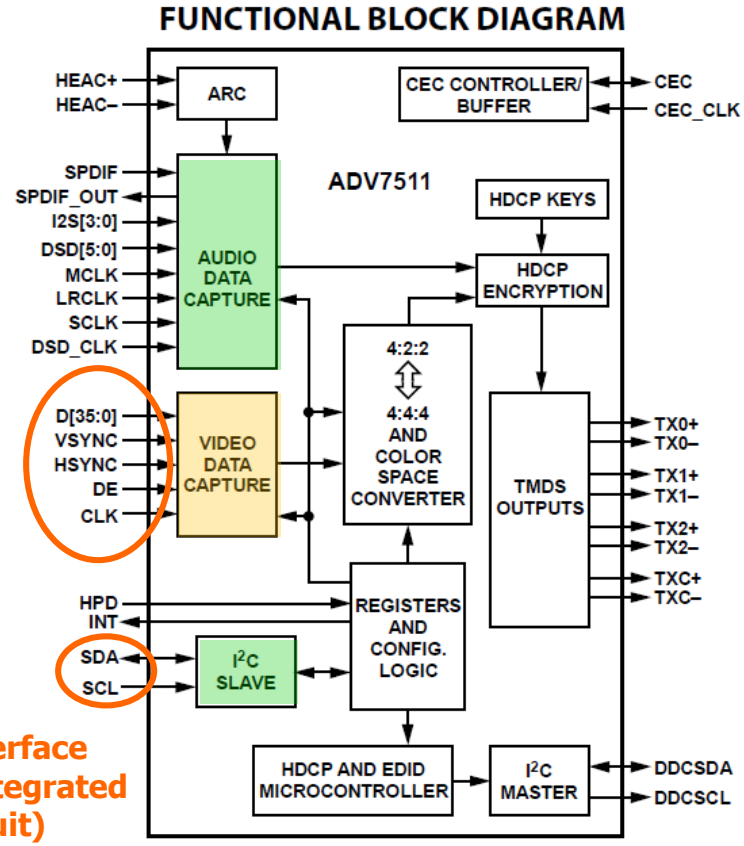
HDMI

Table 7 - HDMI Interface Connections

Signal Name	Description	Zynq pin	ADV7511 pin
HDP	Hot Plug Detect signal input	N/C	30
HD-INT	Interrupt signal output	W16	45
HD-SCL	I2C Interface. Supports CMOS logic levels from 1.8V to 3.3V	AA18	55
HD-SDA		Y16	56
HD-CLK	Video Clock Input. Supports typical CMOS logic levels from 1.8V up to 3.3V	W18	79
HD-VSYNC	Vertical Sync Input (Not required if using embedded syncs)	W17	2
HD-HSYNC	Horizontal Sync Input (Not required if using embedded syncs)	V17	98
HD-DE	Data Enable signal input for Digital Video (Not required if using embedded syncs)	U16	97
HD_D[15:0]	Video Data Input	Bank 35 D0: Y13 D1: AA12 D2: AA14 D3: Y14 D4: AB15 D5: AB16 D6: AA16 D7: AB17 D8: AA17 D9: Y15 D10: W13 D11: W15 D12: V15 D13: U17 D14: V14 D15: V13	 88 87 86 85 84 83 82 81 80 78 74 73 72 71 70 69
HD-SPDIF	Sony/Philips Digital Interface Audio Input	U15	10
HD-SPDIFO	Sony/Philips Digital Interface Audio Output	Y18	46







DE: Data Enable

I²C interface (Inter-Integrated Circuit)

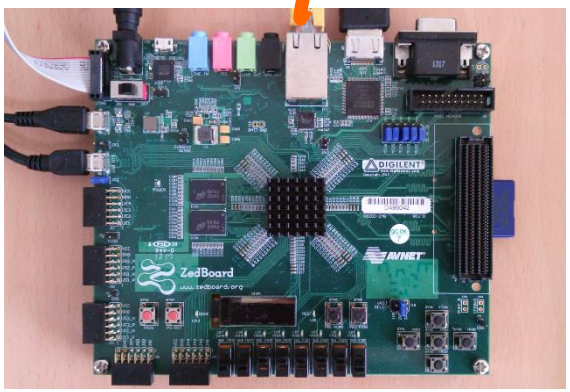
HDMI Transmitter (ADV7511, Analog Devices)

lwIP Lab

- Lighweight IP (lwIP) is an open source TCP/IP networking stack for embedded systems
- The objective of the lab is how to utilize the lwIP library to add networking capability to an embedded system
- The lab design includes the following software applications
  - Echo Server
  - Web server
  - tftp server
  - TCP TX/RX throughput test

lwIP Experiment Setup

Ethernet Cable



Web Server
(192.168.2.7)

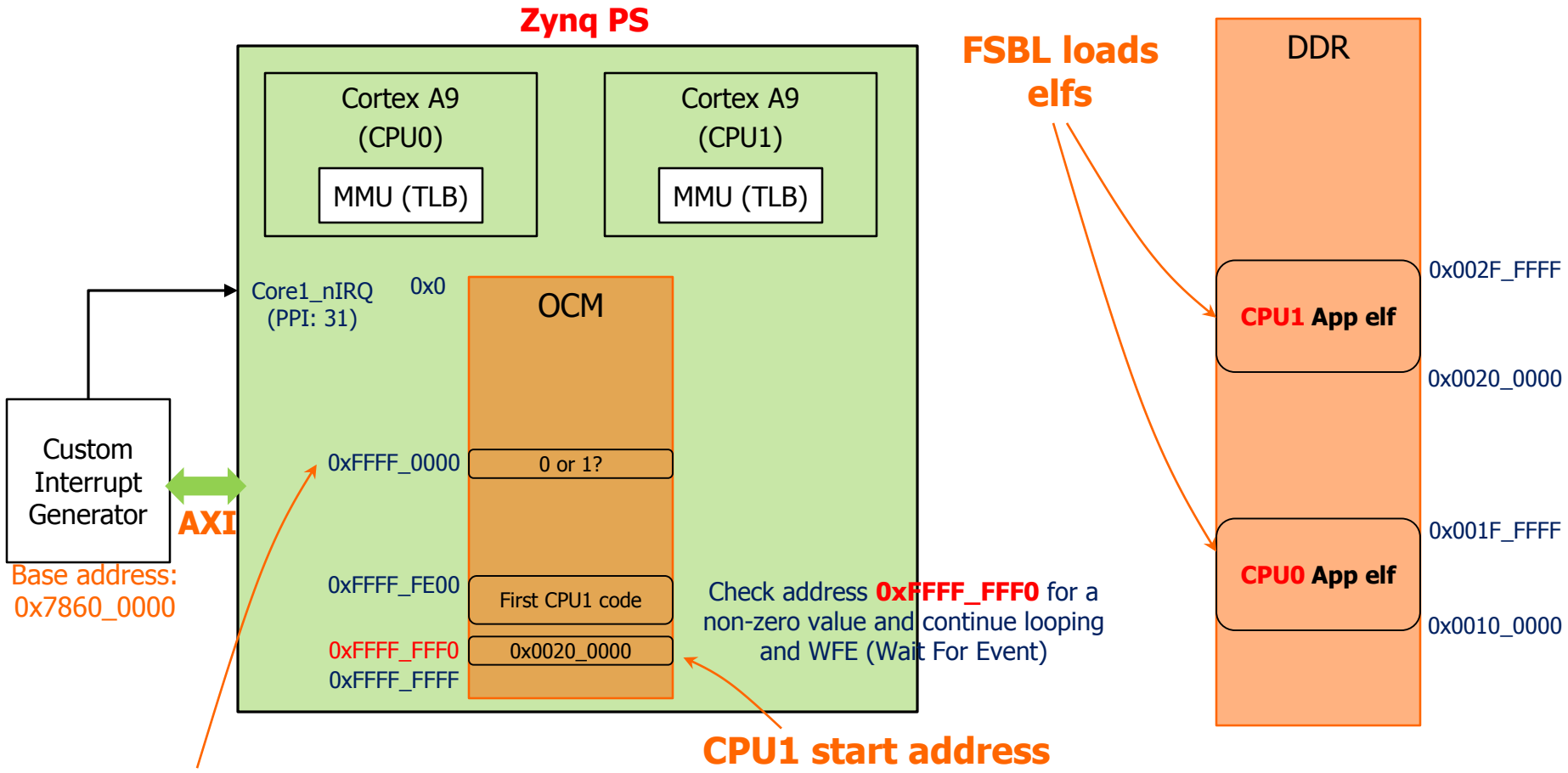
2 Flavors of Apps

- The lab design includes the following software applications
  - Echo Server
  - Web server
  - tftp server
  - TCP TX/RX throughput test

- 2 flavors of apps
  - Socket API-based
    - Reference apps are written on top of FreeRTOS
  - Raw API-based
    - More complicated writing code as it requires knowledge of lwIP Internals

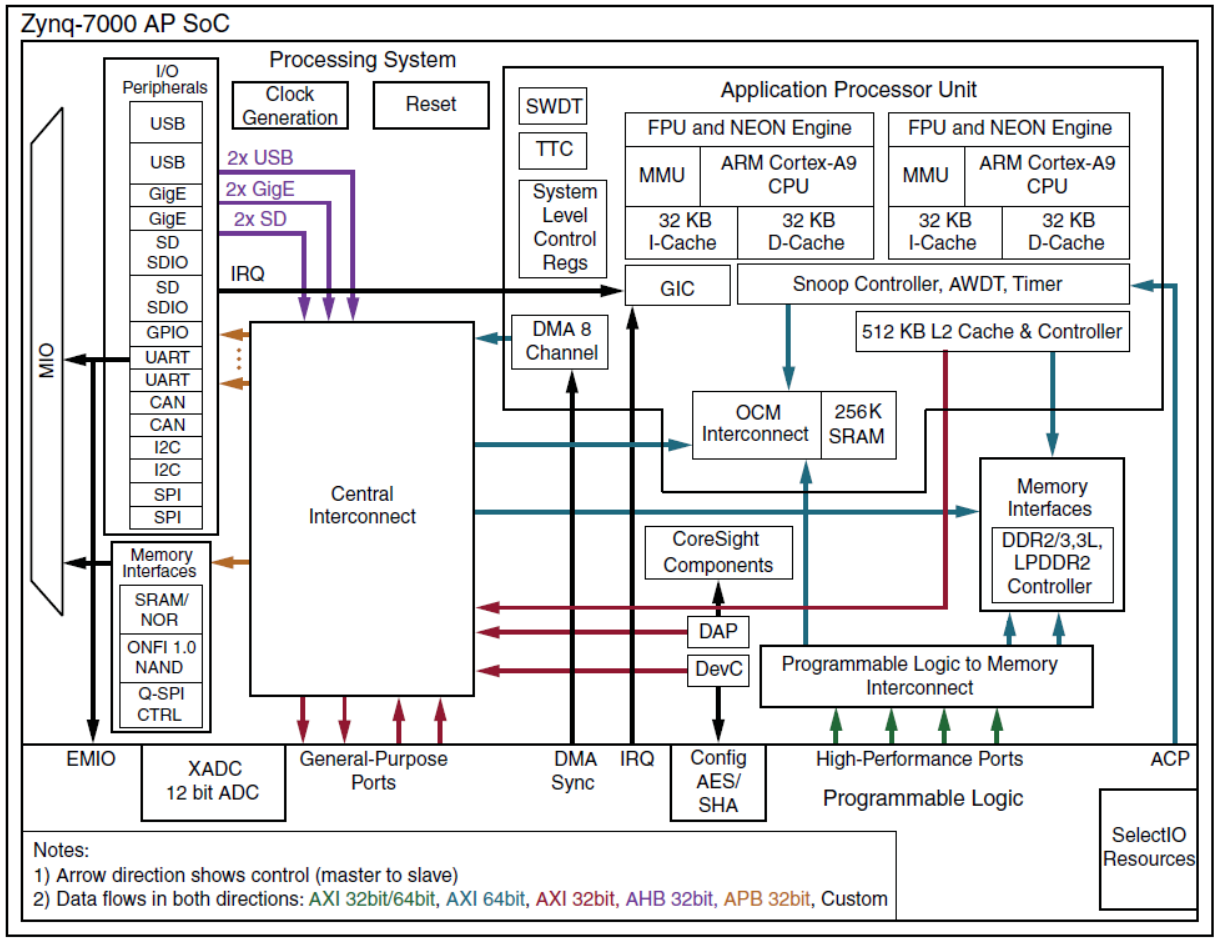
AMP Lab HW & SW Config.

- Asymmetric Multiprocessing (AMP) is a mechanism that allow each processor to run its own operating system or bare-metal application

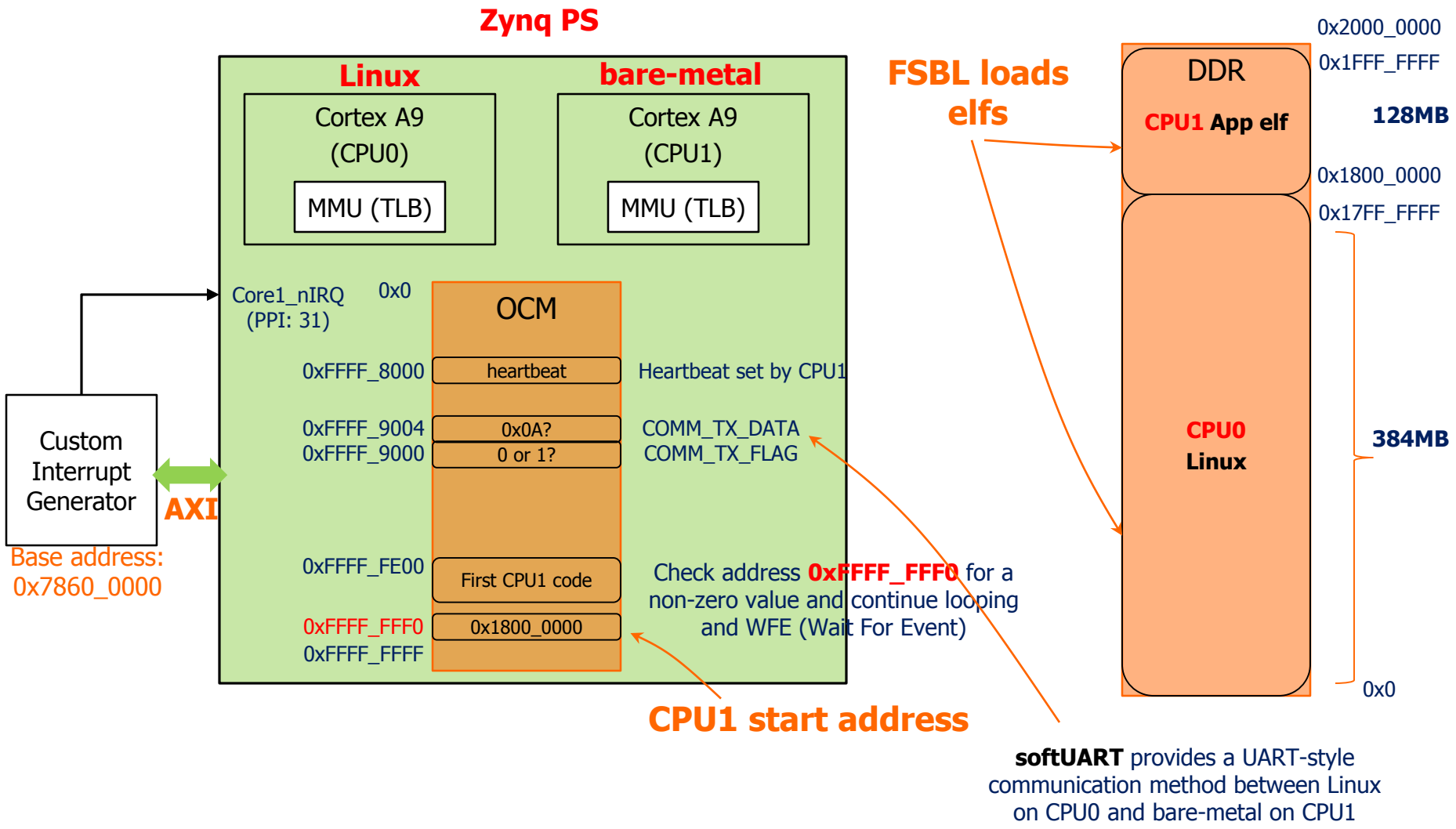


MMU, L1$s in Cortex A9

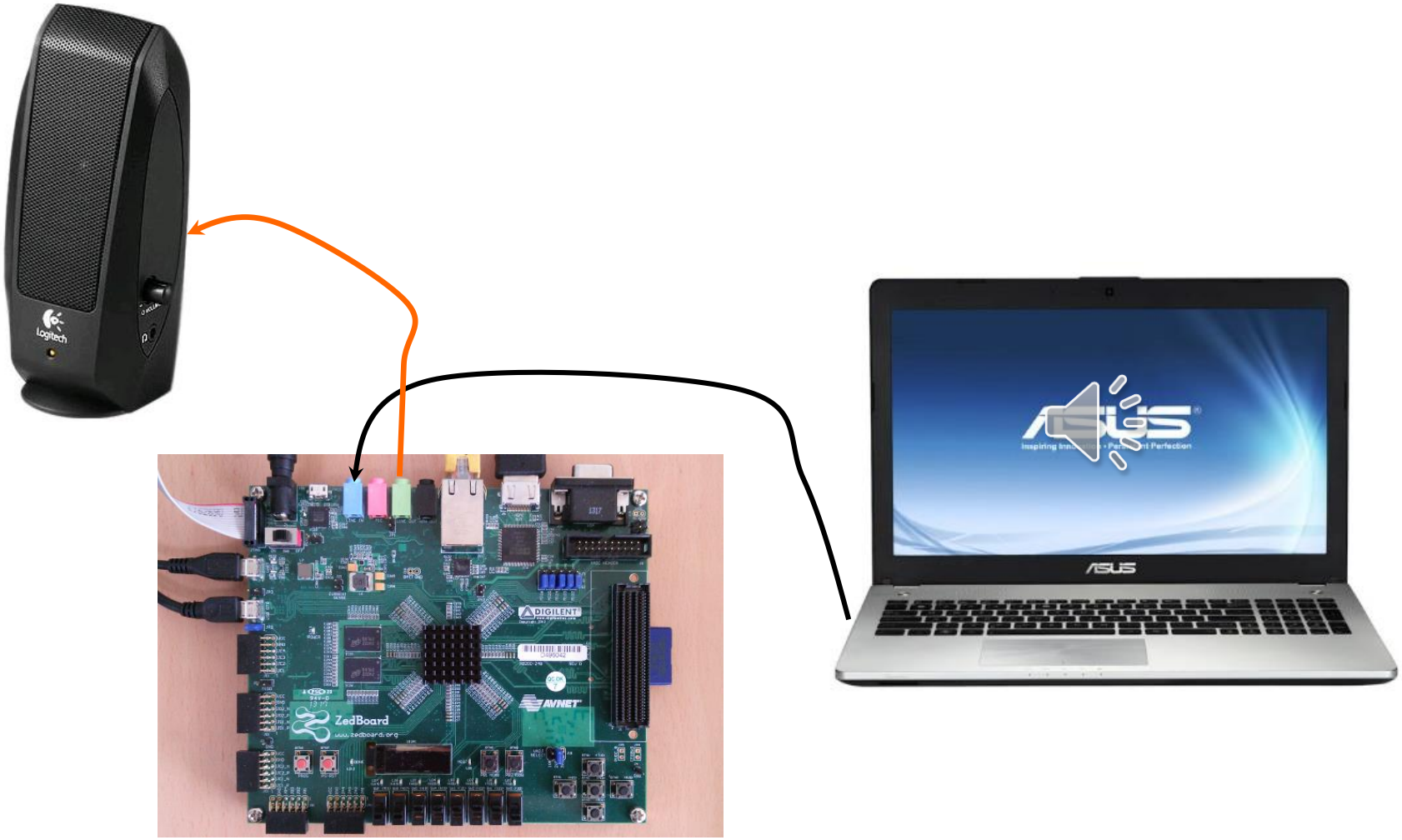
L1I$: Virtually-indexed, Physically tagged
L1D$: Physically-indexed, Physically tagged



AMP Linux Lab Config.



Planahead-based FIR Filter System



Planahead-based Design

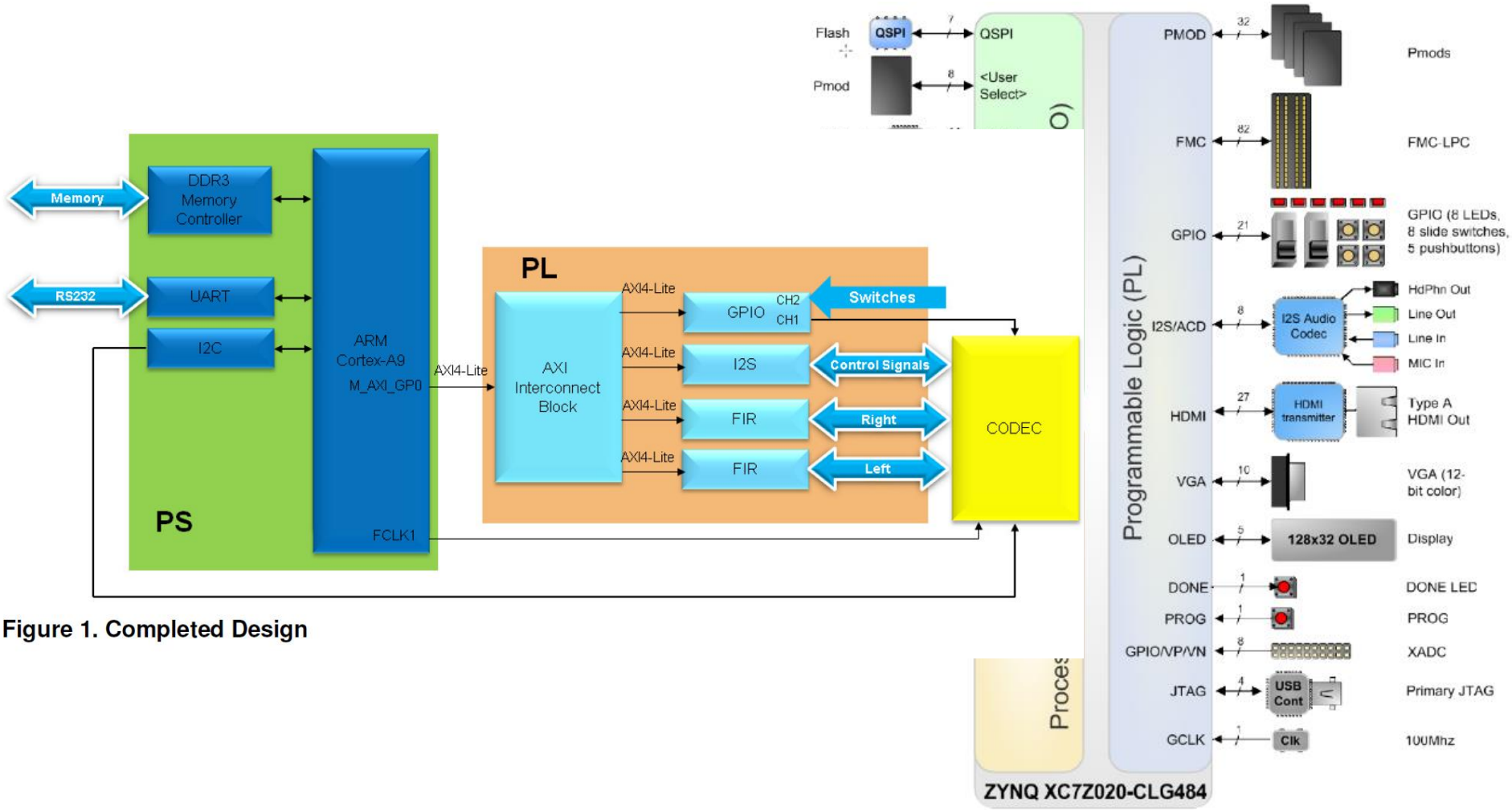
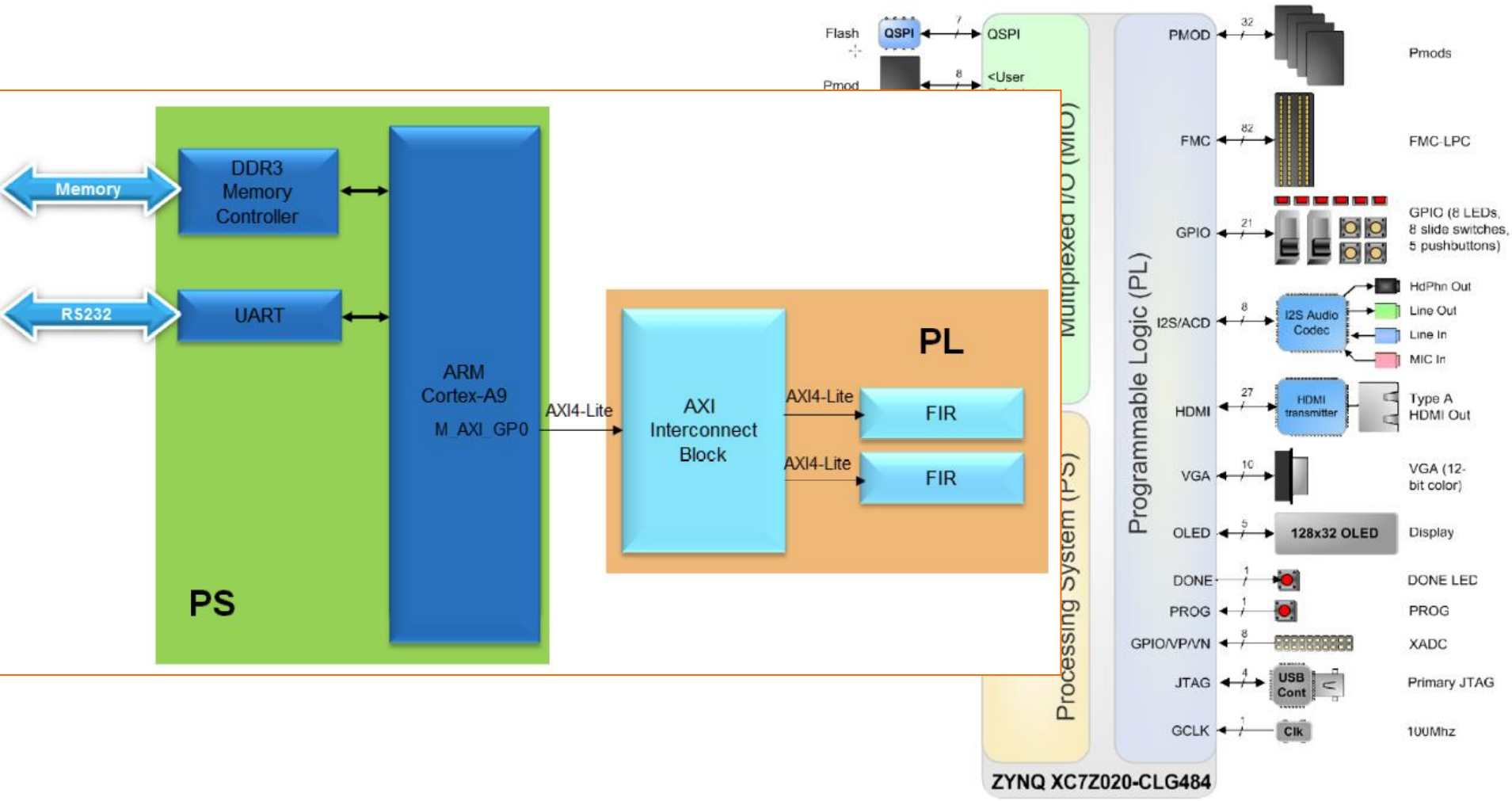


Figure 1. Completed Design

FIR Filter


Memory
DDR3 Memory Controller
RS232
UART
ARM Cortex-A9
M_AXI_GP0
PS
AXI4-Lite
AXI Interconnect Block
PL
AXI4-Lite
FIR
AXI4-Lite
FIR
Flash
QSPI
QSPI
Pmod
<User
Multiplexed I/O (MIO)
Processing System (PS)
Programmable Logic (PL)
PMOD
32
Pmods
FMC
82
FMC-LPC
GPIO
21
GPIO (8 LEDs, 8 slide switches, 5 pushbuttons)
I2S/ACD
8
I2S Audio Codec
HdPhn Out
Line Out
Line In
MIC In
HDMI
27
HDMI transmitter
Type A HDMI Out
VGA
10
VGA (12-bit color)
OLED
5
128x32 OLED
Display
DONE
1
DONE LED
PROG
1
PROG
GPIO/VP/VN
8
XADC
JTAG
4
USB Cont
Primary JTAG
GCLK
1
Clk
100Mhz
ZYNQ XC7Z020-CLG484

gprof

- Use GNU `gprof` to get the profile information
  - Compile and link your code with `-pg` option
  - Run your code
    - `gmon.out` is generated
  - Run `gprof` to interpret the information