SoC Final Presentation

Spartanist - THE Gaming Console

Philipp Dunst, Birgit Haas, Michael Höfler, Michael Hraschan, Philipp Miedl, Oliver Söll, Thomas Unterluggauer, Mario Werner, Paul Wolfger, Adnan Kuleta, Markus Schuß, and Michael Hutter

Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology
Inffeldgasse 16a, A-8010 Graz, Austria

System on Chip 2012

- The course
  - Blocked course (from Oct. till Dec.)
  - 5 ECTS credits (3VU)
- Content
  - Embedded system design
  - Hardware IP design
  - HW/SW interaction
  - Linux, drivers, networks, peripherals, ...
  - Soft skills (english, presentations, discussions, ...)
- 11 participants

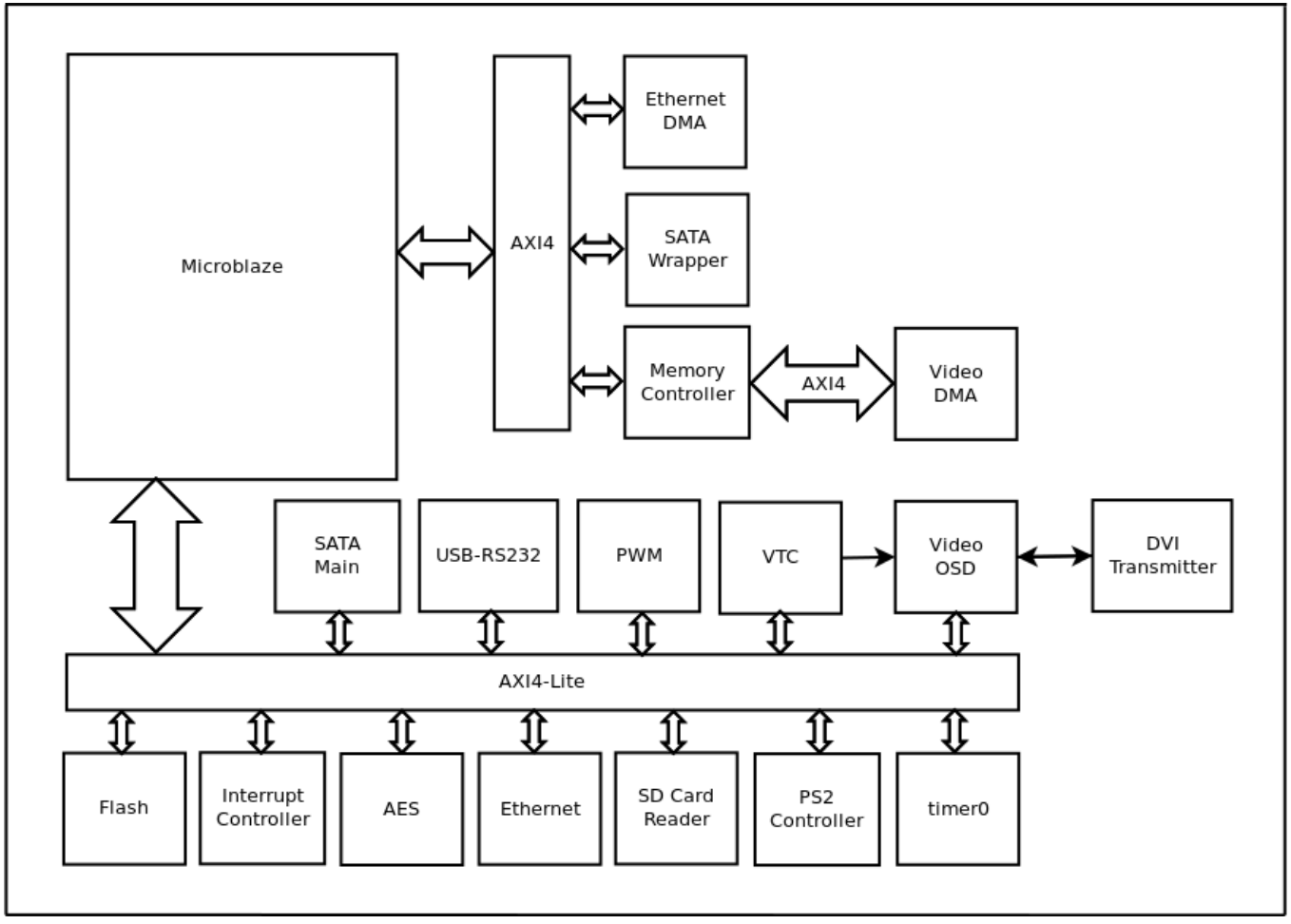
The Project 2012

- Design of an embedded game console
  - Game-console prototype on FPGA
  - Embedded processor/Linux as OS
  - DVI Video out/monitor
  - Audio out/sound
  - Network support (multiplayer)
  - Controllers/joysticks
  - Encrypted games (security) on SDcard
  - HDD support
- Goals
  - Fast time to market, IP reuse, low cost

The System Platform

- Spartan 6 LX150T board
- 128MB DDR3-SDRAM
- 32MB NOR Flash
- 32MB + 8MB platform configuration Flash
- 10/100/1000 Ethernet interface
- SD card interface

Overview


Microblaze
AXI4
Ethernet DMA
SATA Wrapper
Memory Controller
AXI4
Video DMA
SATA Main
USB-RS232
PWM
VTC
Video OSD
DVI Transmitter
AXI4-Lite
Flash
Interrupt Controller
AES
Ethernet
SD Card Reader
PS2 Controller
timer0

DVI Basics (1)

Digital Video Interface

- Lowest supported resolution: 640x480@60 Hz
- DDC (Display Data Channel) Support
  - Extended display identification data
- Hot Pluggable
- Several different power states

DVI Basics (2)

T.M.D.S.

- Transition Minimized Differential Signaling
- Advanced Encoding Algorithm
- Converts 8 bits into 10-bit transition minimized, DC balanced character
- Up to two TMDS links per DVI system
- On boot up just one is active

Design Choices

- DVI output
- 640x480@60 Hz resolution
- RGB color format
  - Each color 8 bit
- Read data from framebuffer

Design Overview

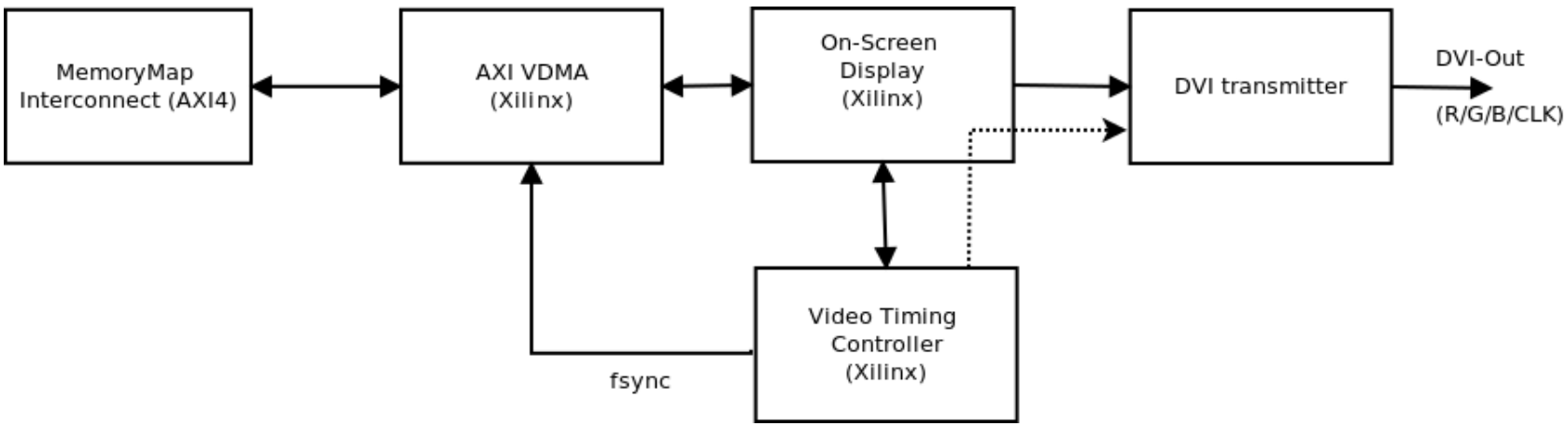
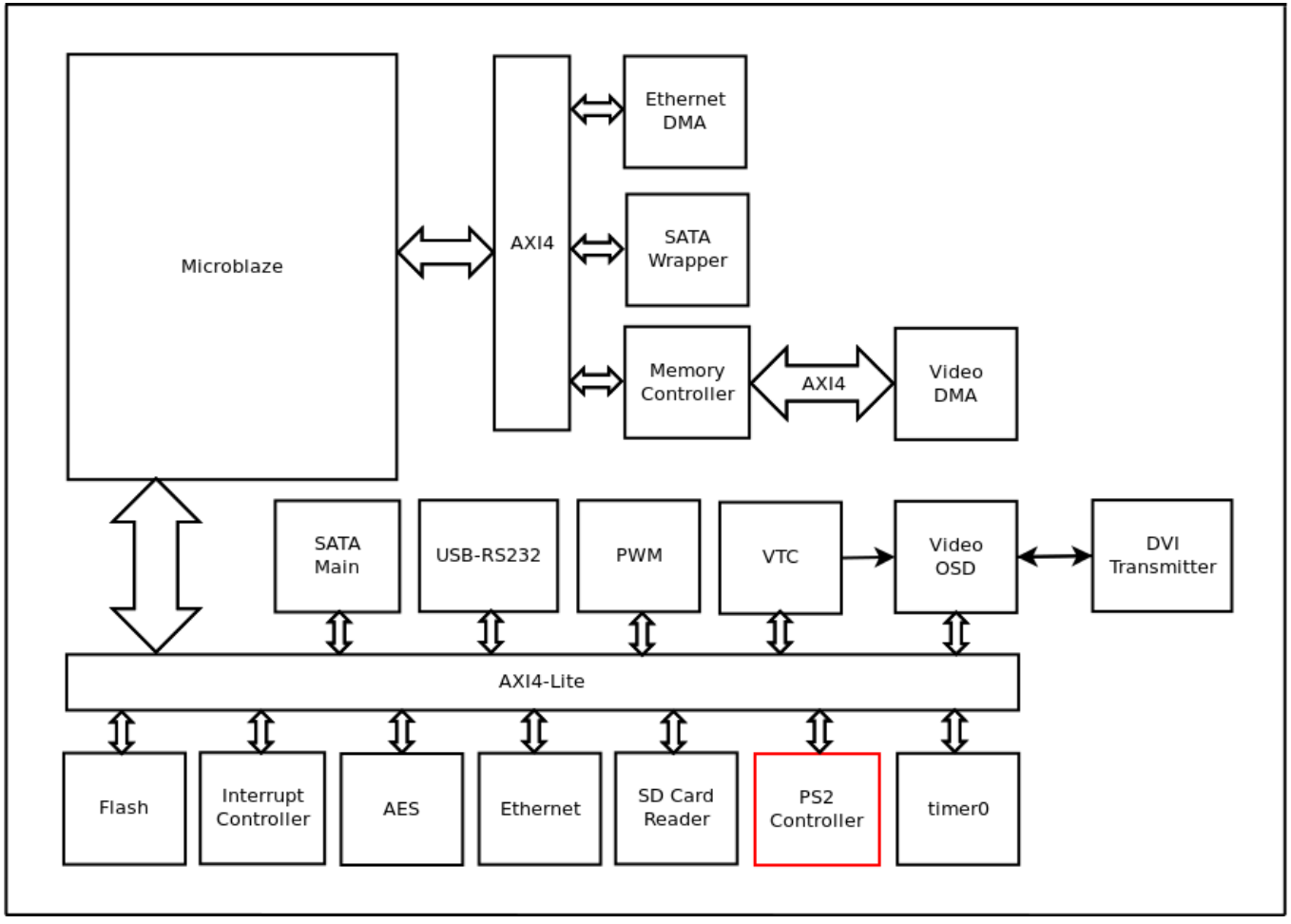


Figure : Video Design Overview

Overview

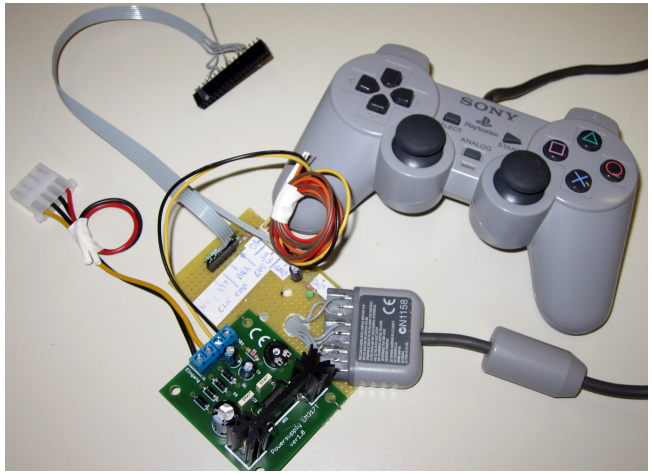

Microblaze
AXI4
Ethernet DMA
SATA Wrapper
Memory Controller
AXI4
Video DMA
SATA Main
USB-RS232
PWM
VTC
Video OSD
DVI Transmitter
AXI4-Lite
Flash
Interrupt Controller
AES
Ethernet
SD Card Reader
PS2 Controller
timer0

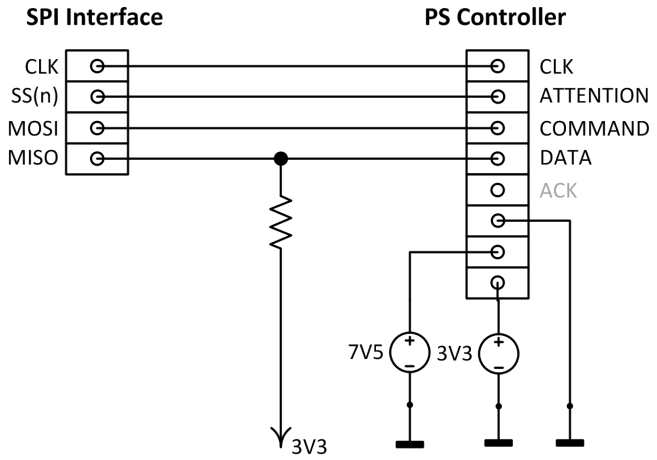
Playstation DualShock Controller

- Features:
  - Digital button values
  - Analog joysticks
  - Analog button-pressure values (DualShock2 only)
  - Rumble motors
- Serial peripheral interface (200 kHz clock frequency)
- 5 to 21 byte data packets
  - Configuration
  - Polling of button states
- Polling command:

Byte	1	2	3	4	5	6	7	8	9
Cmd	01	42	00	1)	2)	00	00	00	00
Data	FF	79	5A	FF	FF	7F	7F	7f	7f
	header			digital		analog joy			

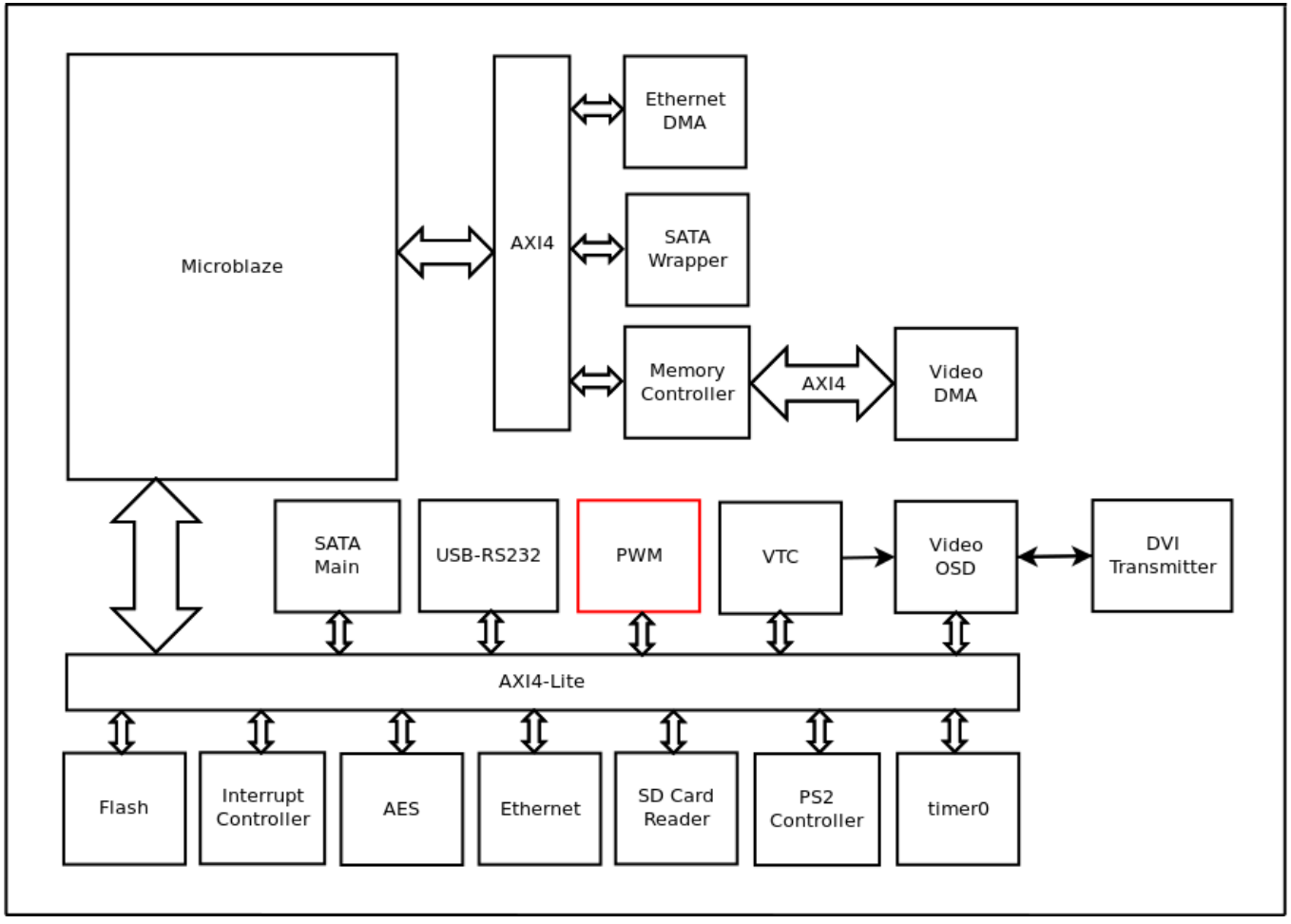
Hardware and Implementation





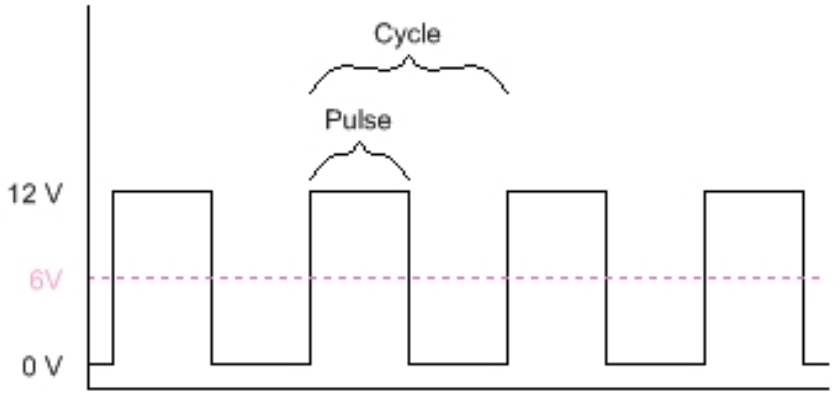
- Xilinx AXI-SPI core
- Slightly modified Xilinx SPI Linux driver
- Input device driver
  - Configuration and polling of the controller
  - Keyboard input events
  - Sound events for motor control

Overview


Microblaze
AXI4
Ethernet DMA
SATA Wrapper
Memory Controller
AXI4
Video DMA
SATA Main
USB-RS232
PWM
VTC
Video OSD
DVI Transmitter
AXI4-Lite
Flash
Interrupt Controller
AES
Ethernet
SD Card Reader
PS2 Controller
timer0

Audio Processing

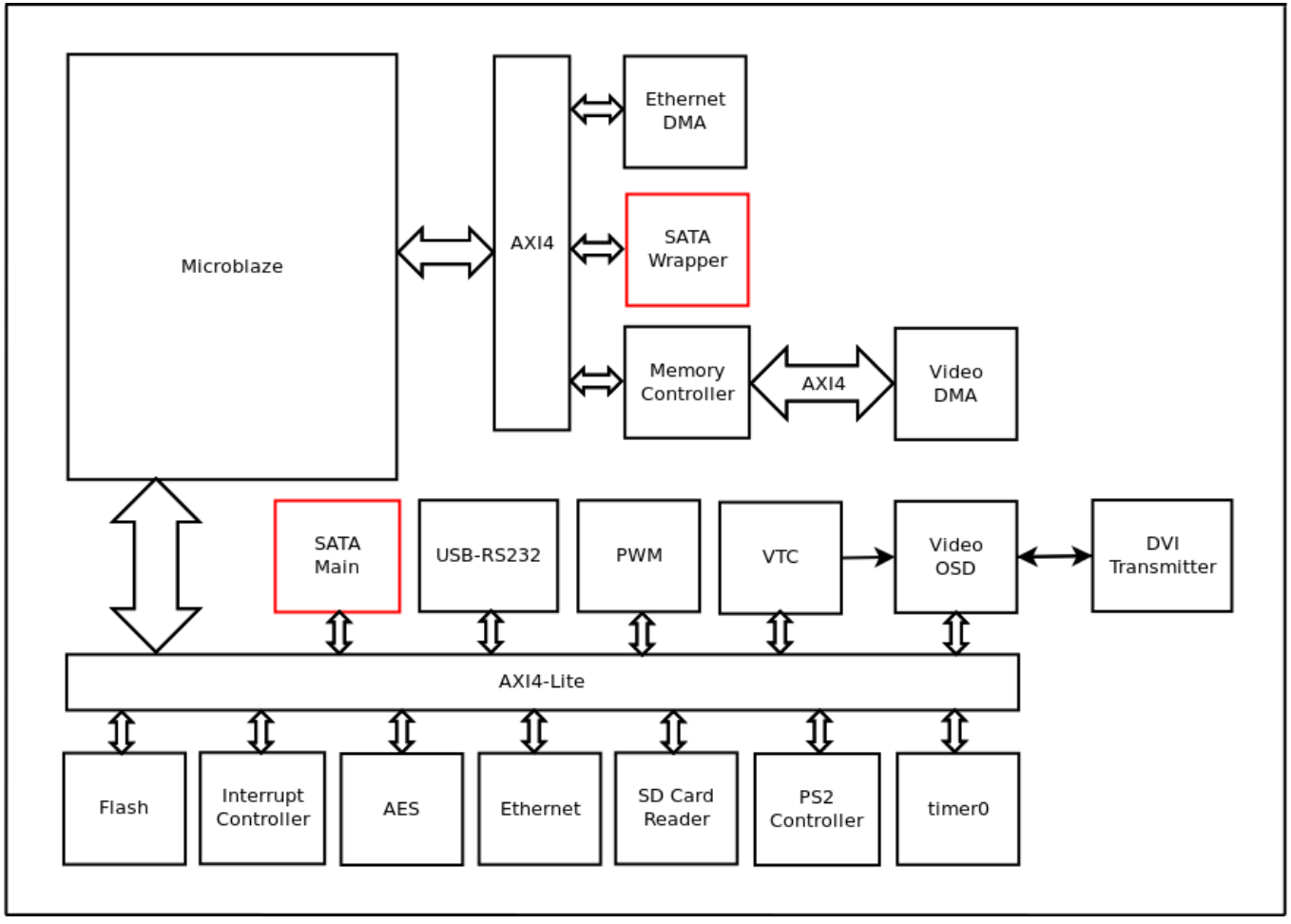
- LogiCORE IP AXI Timer
- Timer core with PWM Mode
- Generates sinus like signal
- External low pass filter and speaker amplifier
- Output via RST pin of ALI (Avnet LCD Interface) and GND



Audio Integration

- Linux module (driver) providing interface to games
- Driver writes values to core registers
- Core generates pulses for given frequency
- Duration of sound implemented with sleep-and-stop thread

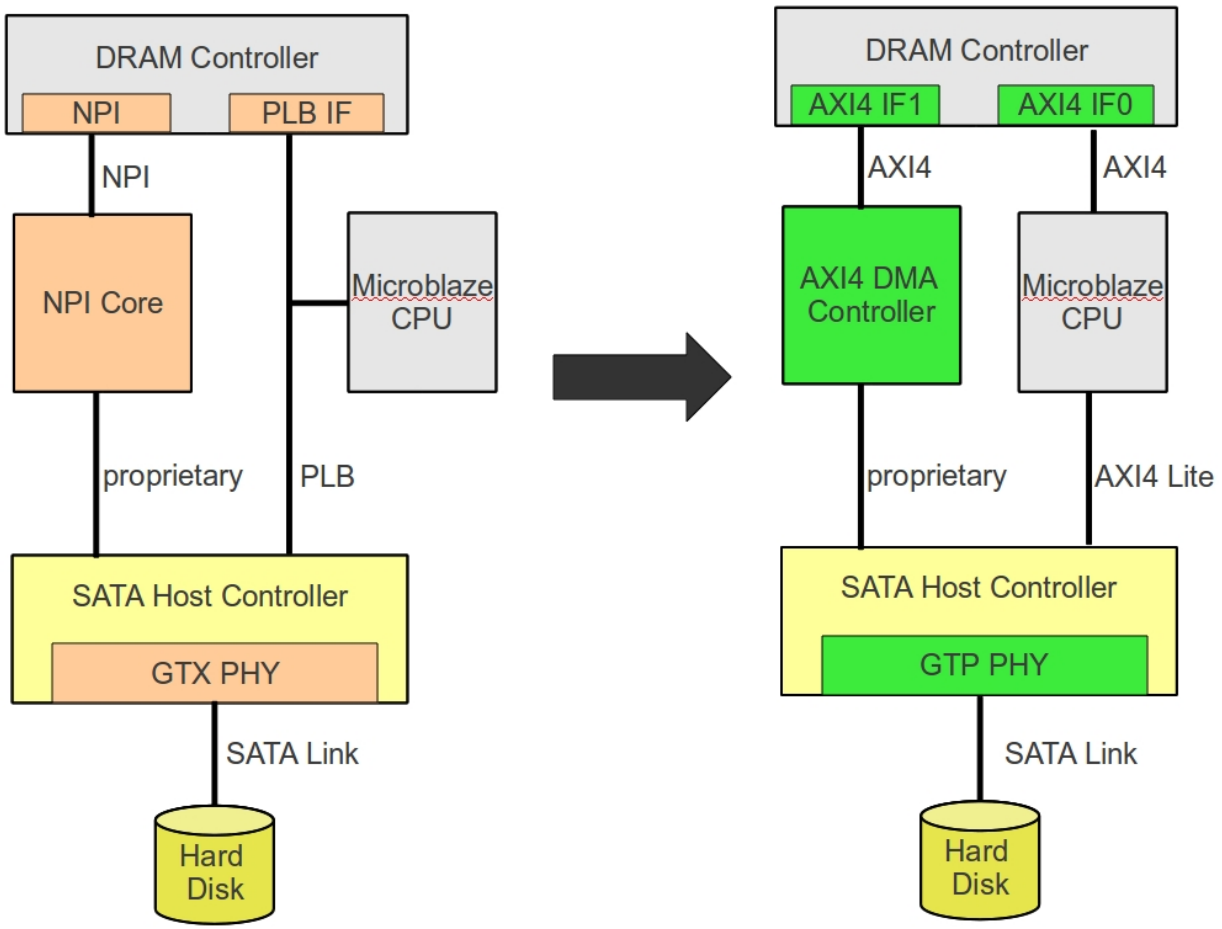
Overview


Microblaze
AXI4
Ethernet DMA
SATA Wrapper
Memory Controller
AXI4
Video DMA
SATA Main
USB-RS232
PWM
VTC
Video OSD
DVI Transmitter
AXI4-Lite
Flash
Interrupt Controller
AES
Ethernet
SD Card Reader
PS2 Controller
timer0

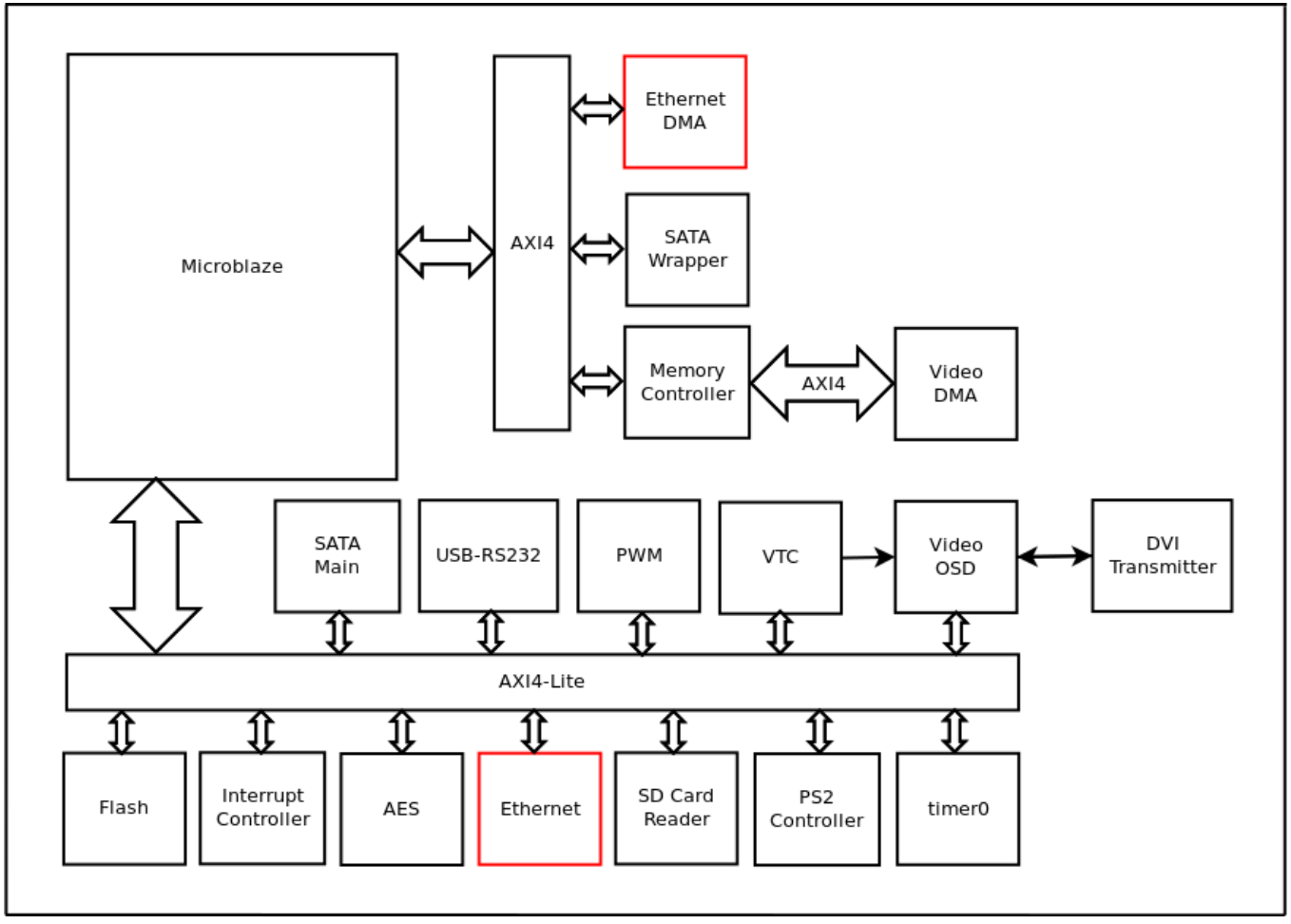
SATA Host Controller Core (1)

- Hard disk to store games
  - Old IDE standard
  - SATA I (1.5 Gbps), SATA II (3 Gbps), SATA III (6 Gbps)
- Plan: implement SATA I
- Free SATA cores are rare, commercial ones are expensive
  - Adapting free core from opencores to fit in our system
  - Include into Linux using block device driver
- Many mistakes to be made
  - Erroneous or missing documentation
  - Bug within the Gigabit Transceiver Wizard for Spartan 6
  - Many problems solved

SATA Host Controller Core (2)

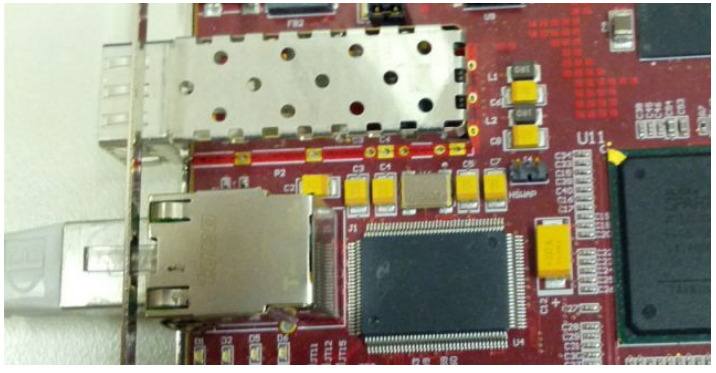


Overview

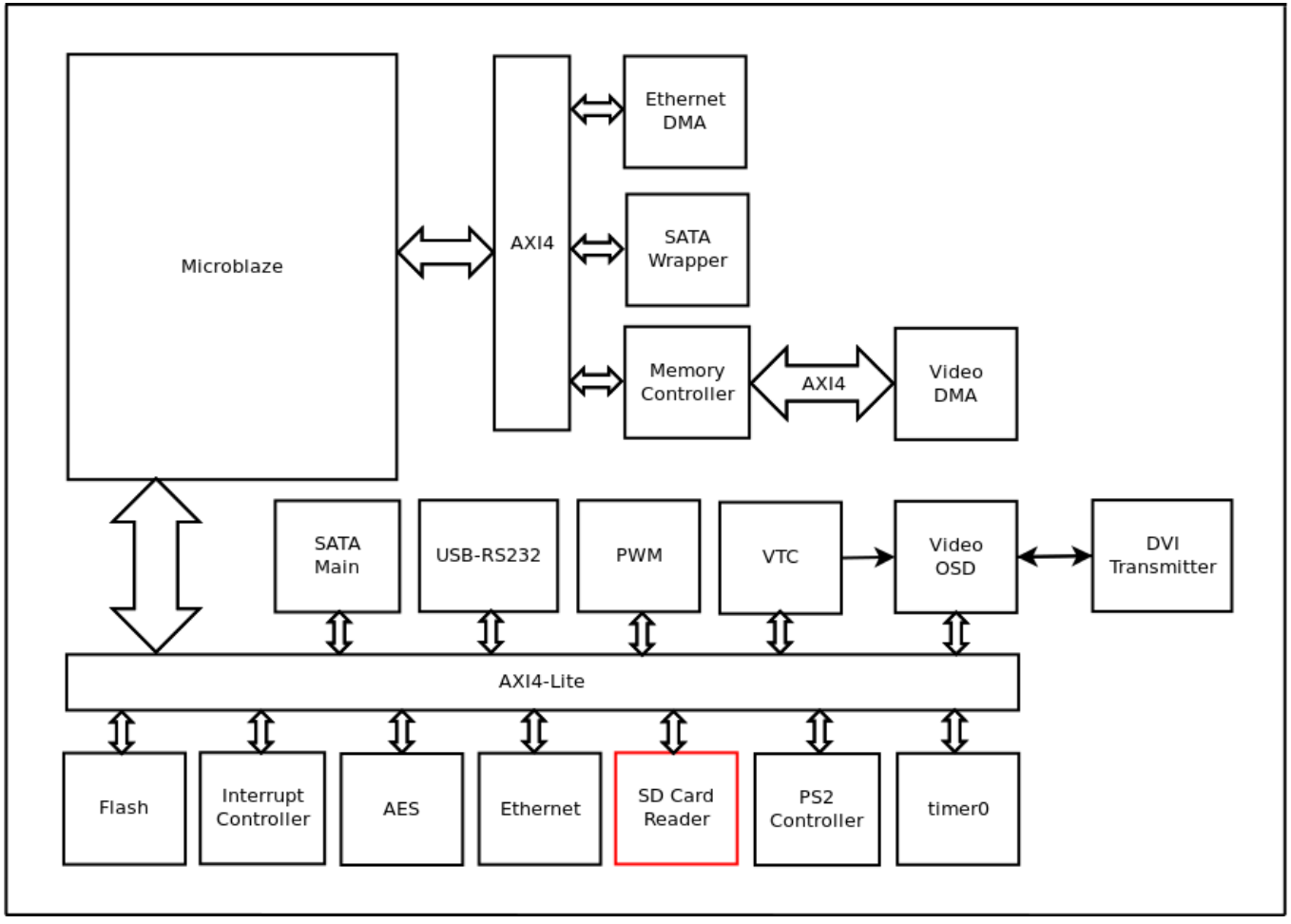

Microblaze
AXI4
Ethernet DMA
SATA Wrapper
Memory Controller
AXI4
Video DMA
SATA Main
USB-RS232
PWM
VTC
Video OSD
DVI Transmitter
AXI4-Lite
Flash
Interrupt Controller
AES
Ethernet
SD Card Reader
PS2 Controller
timer0

Ethernet

- 10/100/1000GBit
- Data transfer via DMA
- IPCore + external PHY

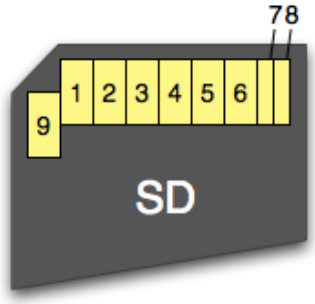


Overview

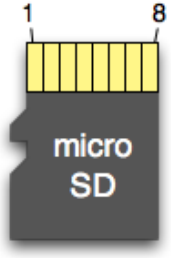

Microblaze
AXI4
Ethernet DMA
SATA Wrapper
Memory Controller
AXI4
Video DMA
SATA Main
USB-RS232
PWM
VTC
Video OSD
DVI Transmitter
AXI4-Lite
Flash
Interrupt Controller
AES
Ethernet
SD Card Reader
PS2 Controller
timer0

SD Card

- SPI legacy interface
- SD/SDHC compatible
- CS, MISO, MOSI, CLK@ 20MHz

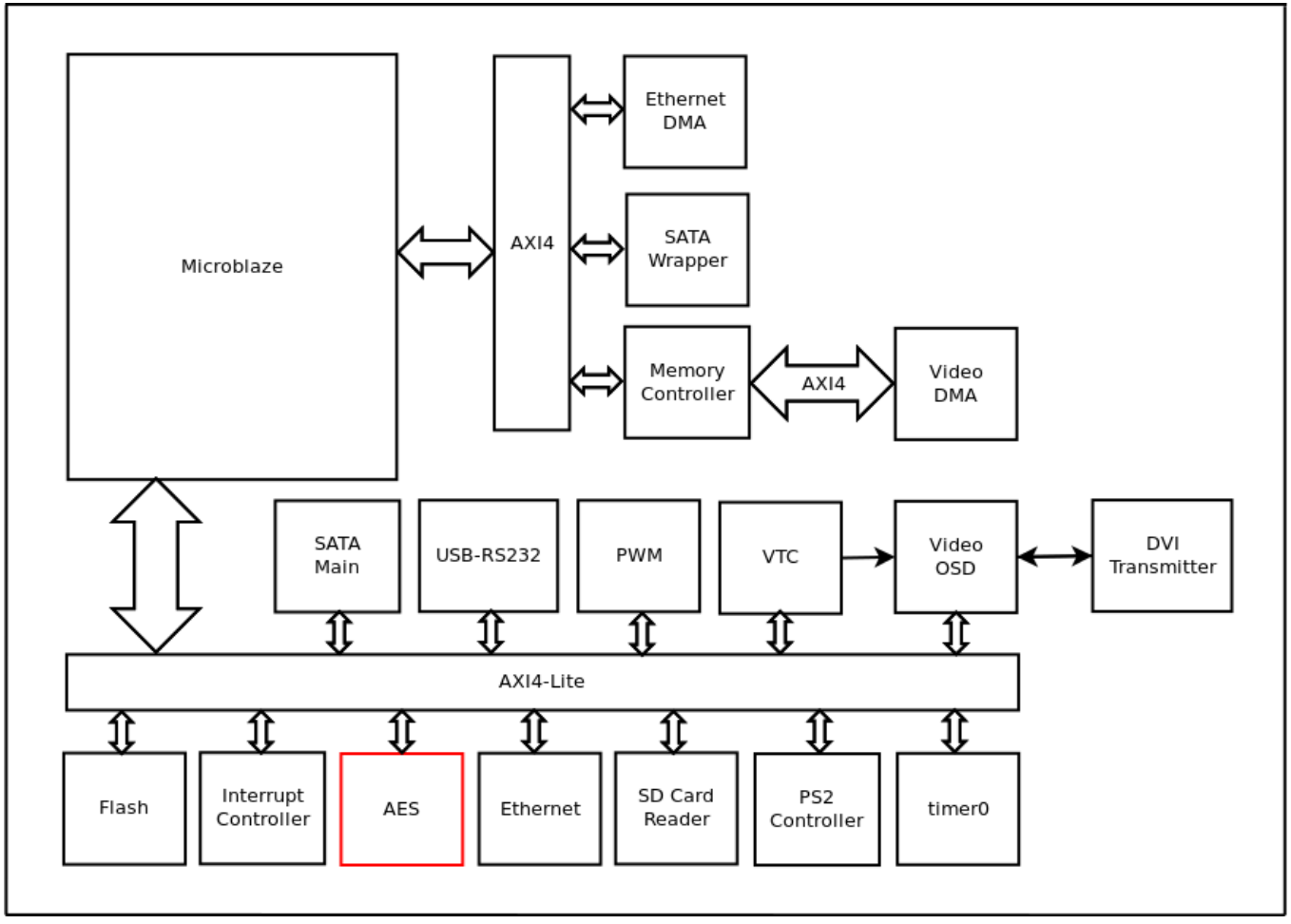


Pin	SD	SPI
1	CD/DAT3	CS
2	CMD	DI
3	VSS1	VSS1
4	VDD	VDD
5	CLK	SCLK
6	VSS2	VSS2
7	DAT0	DO
8	DAT1	X
9	DAT2	X



Pin	SD	SPI
1	DAT2	X
2	CD/DAT3	CS
3	CMD	DI
4	VDD	VDD
5	CLK	SCLK
6	VSS	VSS
7	DAT0	DO
8	DAT1	X

Overview


Microblaze
AXI4
Ethernet DMA
SATA Wrapper
Memory Controller
AXI4
Video DMA
SATA Main
USB-RS232
PWM
VTC
Video OSD
DVI Transmitter
AXI4-Lite
Flash
Interrupt Controller
AES
Ethernet
SD Card Reader
PS2 Controller
timer0

The Crypto Team

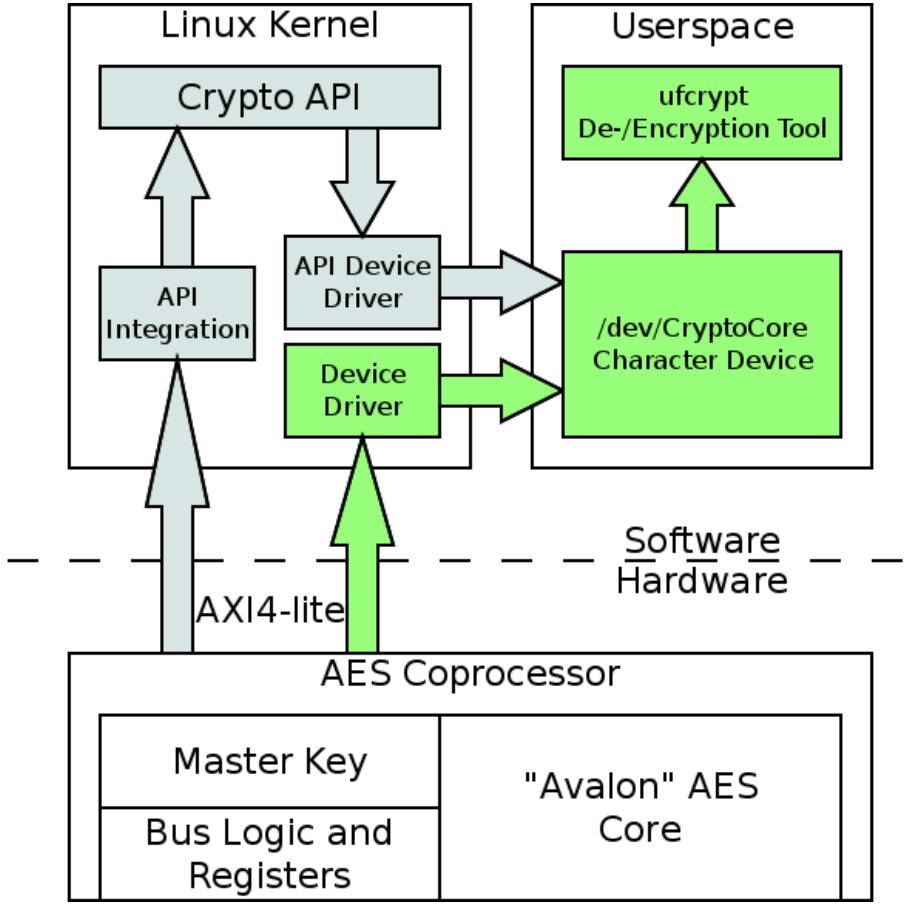
Objective:

- Integrate a crypto core for game de-/encryption.

Implementation:

- “Avalon” AES core from OpenCores
- AXI4-lite bus interface
- Master key within the coprocessor
- Derivation and normal mode possible
- Linux drivers
- ufcrypt (CBC mode)

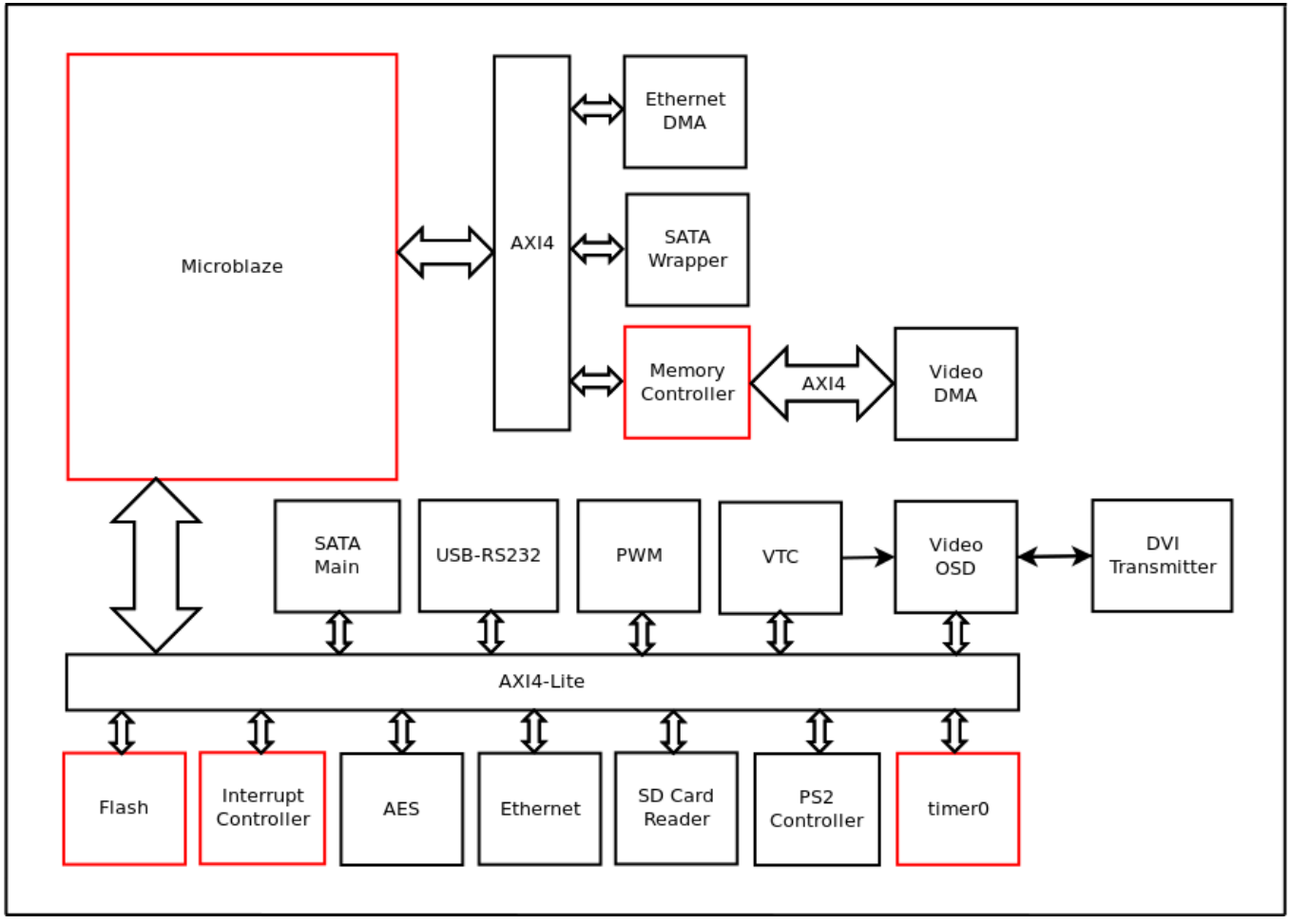
Crypto-Coprocessor Integration



Usage Example

```
$ echo "Plaintext0123456789" > plain.txt
$ strings -3 plain.txt
Plaintext0123456789

$ cat plain.txt | ufcrypt -e > crypt.bin

$ strings -3 crypt.bin
Y!n
(fx

$ cat crypt.bin | ufcrypt > plain_recovered.txt

$ sha1sum plain.txt crypt.bin plain_recovered.txt
e8663f1a35656a426c6da9e866e0aee4e630557d  plain.txt
6e68728ef8233ac7b2a39a4a198d9870d3f32a3c  crypt.bin
e8663f1a35656a426c6da9e866e0aee4e630557d  plain_recovered.txt
```

Overview


Microblaze
AXI4
Ethernet DMA
SATA Wrapper
Memory Controller
AXI4
Video DMA
SATA Main
USB-RS232
PWM
VTC
Video OSD
DVI Transmitter
AXI4-Lite
Flash
Interrupt Controller
AES
Ethernet
SD Card Reader
PS2 Controller
timer0

Linux-Team Objectives

- Select an appropriate processor
- Run Linux on this processor
- Start Linux at power up
- Deploy a root FS
- Ethernet support
- Support other teams in Linux issues
- Basic start-up environment

Our Processor

- Microblaze processor
- 32 bit
- RISC
- Little endian
- MMU support
- 90MHz
- Barrel shifter
- Pipeline depth of 5
- Linux kernel 3.5.0 from Xilinx

Linux - Userland

- Buildroot: ”package manager”
- Bash: debug shell, scripts
- SDL: game library
- nCurses: user interface

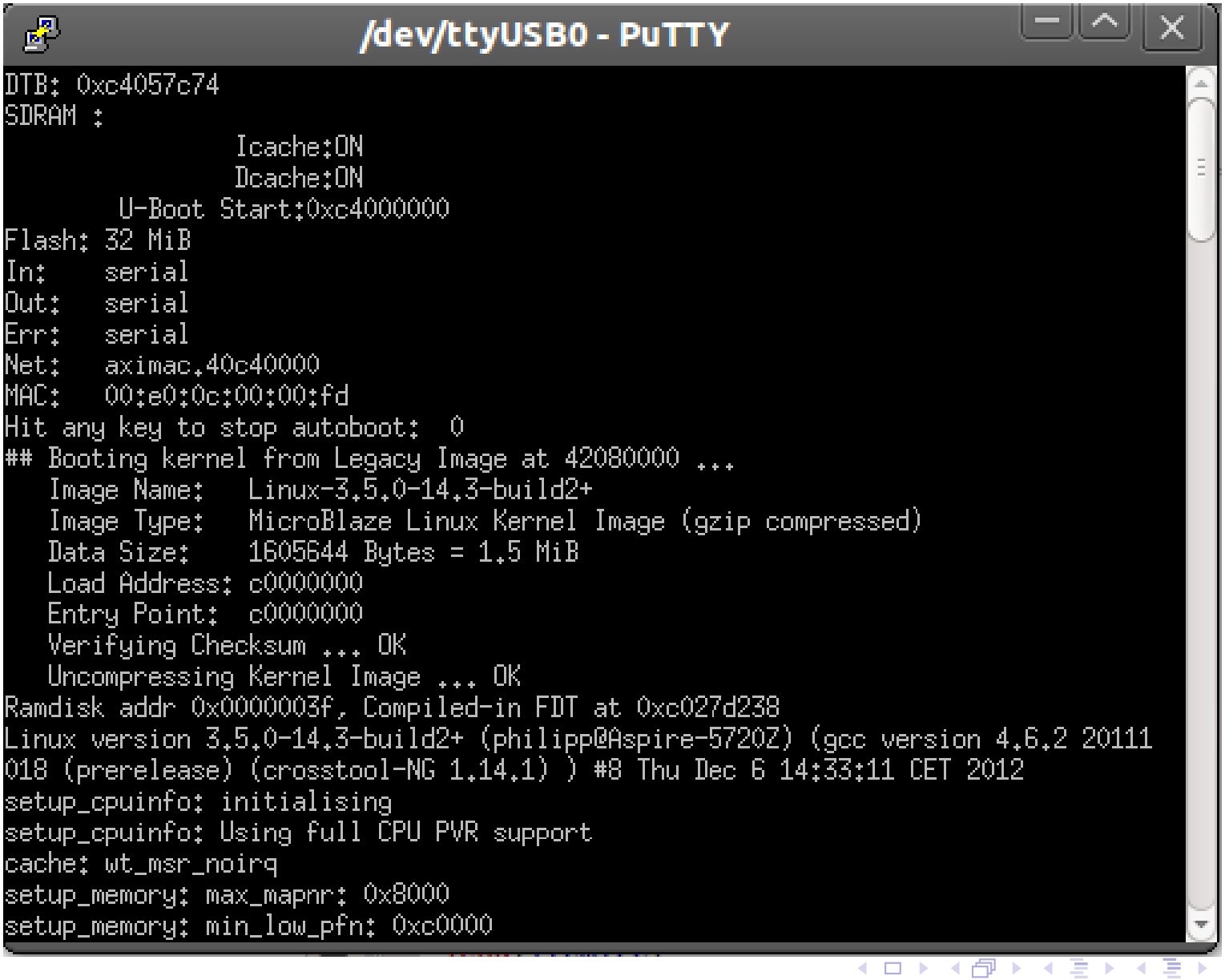
copyright http://buildroot.uclibc.org/

copyright http://www.libsdl.org/

Boot Process

- 2 stage boot process
- First stage
  - SRec
  - Configuration Flash
- Second stage
  - UBoot
  - Denx

Size	Name	Addresses
256KB	UBoot	0x42000000
256KB	UBoot env	0x42040000
5MB	Kernel	0x42080000
2M	SRec UBoot	0x42580000
-	root FS	0x42780000


/dev/ttyUSB0 - PuTTY
DTB: 0xc4057c74
SDRAM :
Icache:ON
Dcache:ON
U-Boot Start:0xc4000000
Flash: 32 MiB
In: serial
Out: serial
Err: serial
Net: aximac.40c40000
MAC: 00:e0:0c:00:00:fd
Hit any key to stop autoboot: 0
Booting kernel from Legacy Image at 42080000 ...
Image Name: Linux-3.5.0-14.3-build2+
Image Type: MicroBlaze Linux Kernel Image (gzip compressed)
Data Size: 1605644 Bytes = 1.5 MiB
Load Address: c0000000
Entry Point: c0000000
Verifying Checksum ... OK
Uncompressing Kernel Image ... OK
Ramdisk addr 0x0000003f, Compiled-in FDT at 0xc027d238
Linux version 3.5.0-14.3-build2+ (philipp@Aspire-5720Z) (gcc version 4.6.2 20111
018 (prerelease) (crosstool-NG 1.14.1)) #8 Thu Dec 6 14:33:11 CET 2012
setup_cpuinfo: initialising
setup_cpuinfo: Using full CPU PVR support
cache: wt_msr_noirq
setup_memory: max_mapnr: 0x8000
setup_memory: min_low_pfn: 0xc0000

Video Driver

- Initialize the three cores for DVI video output
- Adapted the existing virtual framebuffer

```
//dvi.c
videomemory = kmalloc(640*480*4, GFP_DMA);
phy_addr = virt_to_phys(videomemory);

//fb.c
vfb_fix.smem_start = (unsigned long) videomemory;
vfb_fix.smem_len = videomemorysize;
```

GUI

- nCurses
- Spawns automatically on framebuffer
- Used as basic environment
- Automatically lists all installed games
- Functionality
  - Install an encrypted game from an SD card
  - Uninstall all games
  - Start an installed game

Install and Launch Process

- Stored on the SD card
  - game.crypt
    - Encrypted
    - tar.gz
  - des.txt
- Decrypt and extract to a predefined location via Shell script
- Executed via Shell script

```
#!/bin/sh

HDD=/mnt/HDD
SD=/mnt/SD
CRYPT=/usr/bin/ufcrypt

cd $HDD
cat $SD/game.crypt | $CRYPT  | tar xzf -
```

GUI

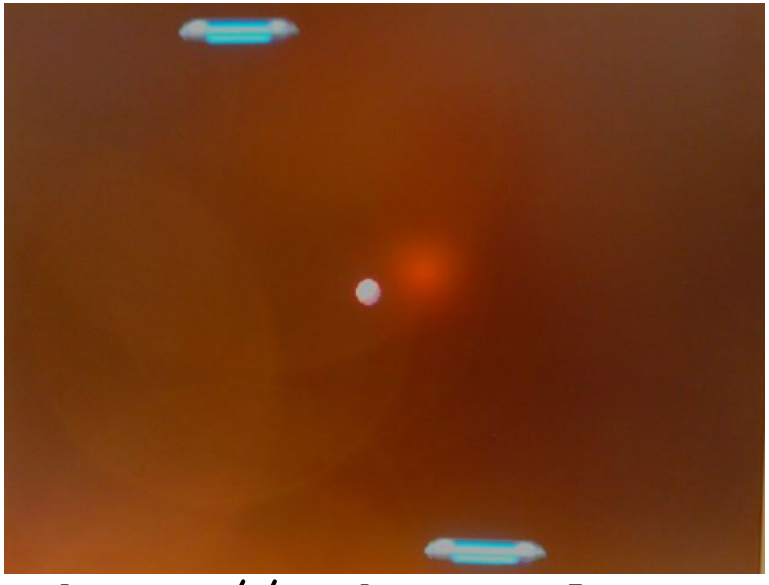

-------- Welcome to SPARTANIST --------
install snake
snake
openpong
uninstall games
-------- The SOC12 Project --------

Singleplayer Game

- Snake
  - Patching from source
    - It's built on SDL library
    - SDL initialization without doublebuffering
    - Resolution 640x480
    - *//snake_update_world*(...)

`https://bitbucket.org/grouzen/snake-sdl`

Multiplayer Game



- OpenPong v0.4.2
  - It's built on top of the SDL
  - Remove background
  - Resize canvas
  - Change the redraw

`http://code.google.com/p/openpong/`

Demo