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
1 Overview
   - System on Chip 2012

2 Hardware Team
   - System
   - Video
   - Controller
   - Audio
   - SATA
   - Ethernet/SD card

3 Crypto Team

4 Linux Team
   - Objectives
   - Processor
   - Boot Process
   - GUI
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
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

Graz, 2012
SoC Design
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:**

<table>
<thead>
<tr>
<th>Byte</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cmd</td>
<td>01</td>
<td>42</td>
<td>00</td>
<td>1)</td>
<td>2)</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Data</td>
<td>FF</td>
<td>79</td>
<td>5A</td>
<td>FF</td>
<td>FF</td>
<td>7F</td>
<td>7F</td>
<td>7f</td>
<td>7f</td>
</tr>
</tbody>
</table>

- 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
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
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
Ethernet

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

Institute for Applied Information Processing and Communications (IAIK)

Graz, 2012

SoC Design
SD Card

- SPI legacy interface
- SD/SDHC compatible
- CS, MISO, MOSI, CLK@ 20MHz
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
- ufencrypt (CBC mode)
Crypto-Coprocessor Integration

- Linux Kernel
  - Crypto API
  - API Integration
  - API Device Driver
  - Device Driver

- Userspace
  - ufcrypt De-/Encryption Tool
  - /dev/CryptoCore Character Device

- AXI4-lite
- AES Coprocessor
  - Master Key
  - Bus Logic and Registers
  - "Avalon" AES Core

Software
Hardware

Graz, 2012
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
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
Boot Process

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

<table>
<thead>
<tr>
<th>Size</th>
<th>Name</th>
<th>Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>256KB</td>
<td>UBoot</td>
<td>0x42000000</td>
</tr>
<tr>
<td>256KB</td>
<td>UBoot env</td>
<td>0x42040000</td>
</tr>
<tr>
<td>5MB</td>
<td>Kernel</td>
<td>0x42080000</td>
</tr>
<tr>
<td>2M</td>
<td>SRec UBoot</td>
<td>0x42580000</td>
</tr>
<tr>
<td>-</td>
<td>root FS</td>
<td>0x42780000</td>
</tr>
</tbody>
</table>
INSTITUTE FOR APPLIED INFORMATION PROCESSING AND COMMUNICATIONS (IAIK)

Graz, 2012 SoC Design

/dev/ttyUSB0 - PuTTY

DTB: 0xc057c74
SDRAM:
   Icache:ON
   Dcache:ON
U-Boot Start:0xc0400000
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 20111108 (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 = videomemorsize;
```
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

```bash
#!/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