FACHHOCHSCHULE **TRIER**
Hochschule für Technik, Wirtschaft und Gestaltung
University of Applied Sciences
Informatik

Spielekonsolenprogrammierung

Hardware, Speicher, Joypad

Übersicht

- Hardware der PSP
- Speicheraufbau und Verwendung der PSP
- Joypadabfrage auf der PSP

Hardwareaufbau

- Generelle Hardwarefeatures

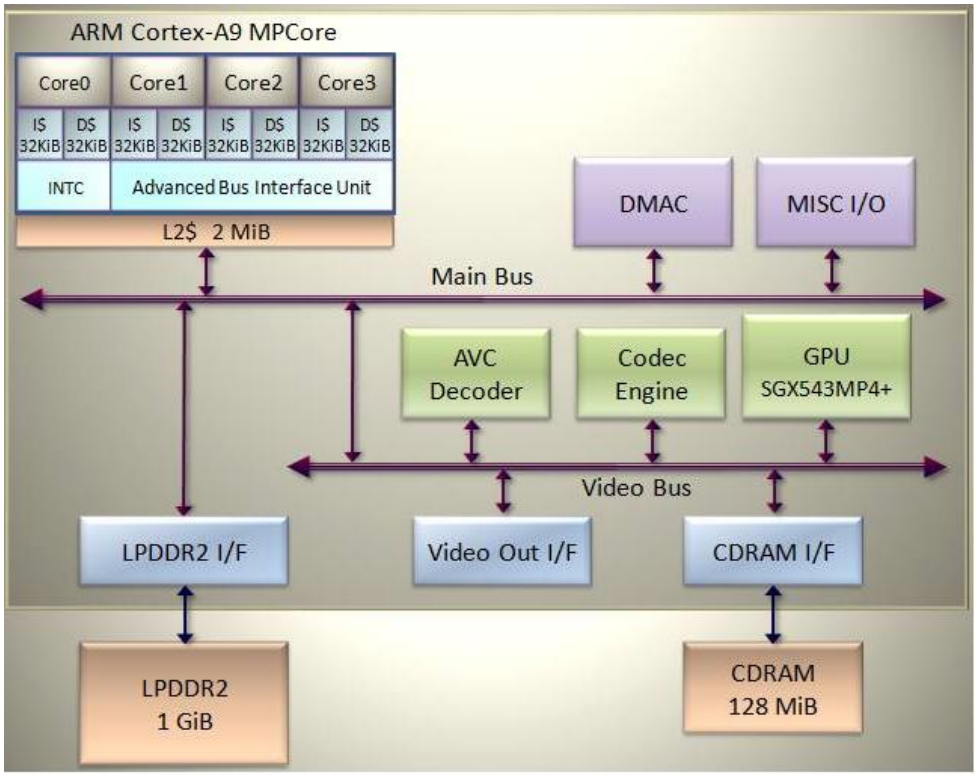
- System:
  - Quad Core ARM CPU (3 benutzbar)
  - Spezial GPU (Tile basiert, ungefähr Shader Model 3)
  - 128MiB Video Memory, 112MiB benutzbar
  - 1 GiB Hauptspeicher (benutzbar max 256 GiB + 26MiB Spezialspeicher)
  - Displayauflösung: 960x544

Hardwareaufbau

- Peripherie:
  - Konsolensteuerung mit 2 Sticks und diversen Buttons
  - 2 Touchpads mit je 6 Punkten, kraftsensitiv
  - 2 Kameras (vorne, hinten, VGA Auflösung)
  - Bewegungssensor (3 Achs Accelerometer, 3 Gyrokompasse, Magnetkompass)
  - Netzwerk: USB, WLAN, Bluetooth
  - Spezialchip für Videokompression / Dekompression

Hardwareaufbau

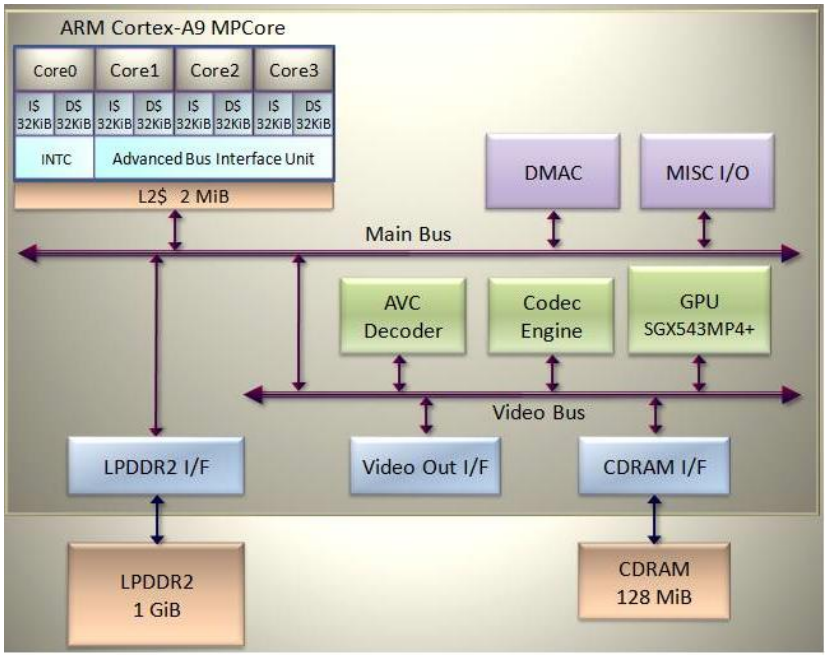
- Übersicht Blockschaltbild:



Hardwareaufbau

- Wesentliche Features:
  - L1,D1 Cache: je 32KiB
  - L2$ Cache: 2MiB

  - GPU Kann direkt auf LPDDR2 zugreifen:
    VORSICHT CPU CACHE!



Memory Management

- Speicher kann in Blöcken angefordert werden
  - Maximal 2048 je Anwendung
  - Speicheranforderung hat Informationen zu
    - Größe
    - Alignment
    - Art
- Dokumentiert in Kernel / MemoryManager

FACHHOCHSCHULE TRIER
Hochschule für Technik, Wirtschaft und Gestaltung
University of Applied Sciences
Informatik

- Speicheranforderung

```
#include <kernel.h>
SceUID sceKernelAllocMemBlock(
    const char *name,
    SceKernelMemBlockType type,
    SceSize vsize,
    struct SceKernelAllocMemBlockOpt *pOpt
);
```

- Type:

SCE_KERNEL_MEMBLOCK_TYPE_USER_RW	(Size multiple of 4 KB)
SCE_KERNEL_MEMBLOCK_TYPE_USER_RW_UNCACHE	(Size multiple of 4 KB)
SCE_KERNEL_MEMBLOCK_TYPE_USER_MAIN_PHYCONT_RW	(Size multiple of 1 MB)
SCE_KERNEL_MEMBLOCK_TYPE_USER_MAIN_PHYCONT_NC_RW	(Size multiple of 1 MB)
SCE_KERNEL_MEMBLOCK_TYPE_USER_CDRAM_RW	(Size of 256KB always cache disabled)

- Umwandlung von BlockId in Pointer:

```
#include <kernel.h>
int sceKernelGetMemBlockBase(
    SceUID uid,
    void **ppBase
);
```

- Freigabe eines Blocks:

```
#include <kernel.h>
int sceKernelFreeMemBlock(
    SceUID uid
);
```

FACHHOCHSCHULE TRIER
Hochschule für Technik, Wirtschaft und Gestaltung
University of Applied Sciences
Informatik

- Möglichkeit des Alignments

```
#include <kernel.h>
SceUID sceKernelAllocMemBlock(
    const char *name,
    SceKernelMemBlockType type,
    SceSize vsize,
    struct SceKernelAllocMemBlockOpt *pOpt
);
```

```
typedef struct SceKernelAllocMemBlockOpt {
    SceSize size;
    SceUInt32 attr;
    SceSize alignment;
    SceUInt32 reserved;
    const char *strBaseBlockName;
} SceKernelAllocMemBlockOpt;
```

Attr: SCE_KERNEL_ALLOC_MEMBLOCK_ATTR_HAS_ALIGNMENT

Alignment muss 2er Potenz sein, max 16MB, min 4KB LPDDR2, 256KB CDRAM

AUFGABE 1

- Besorgen Sie sich je 5MB gecachten und 5MB ungecachten LPDDR2 Speicher (Hauptspeicher)
- Füllen Sie den Speicher bytewise mit 0xfe auf.
- Lesen Sie danach gleich den Speicher wieder aus bei dem Sie die Ergenisse bitweise zusammenodern.

- Messen Sie die ungefähre Laufzeit und erklären Sie den Unterschied.
- Benutzen Sie dazu die sceRtcGetCurrentTick aus der RTC library. (Diese muss dazu gelinked werden)
- Bilden Sie die Differenz mit den unteren 4 Bytes.

Memory

- Praxis Tipp:

- Benutzen Sie in erster Linie das Heap Symbol für normalen Speicher (sonst dürfen Sie sich eine eigene Speicherverwaltung schreiben)
- Benutzen Sie die vorgestellte Library später im Zusammenhang, die mit der Graphik verwendet werden müssen (uncached und CDRAM)

Joypad

- Wird mit Controller Service erledigt (zu finden unter I/O Devices)
- Benötigt lib „libSceCtrl_stub.a“ und header file „ctrl.h“
- Vorgehensweise:
  - Pad Einschalten:
    - SCE_CTRL_MODE_DIGITALONLY (default)
    - SCE_CTRL_MODE_DIGITALANALOG (alte PSP)
    - SCE_CTRL_MODE_DIGITALANALOG_WIDE
  - Pad auslesen

Joypad

- System liefert zu jedem VSYNC Daten
- Daten können in Batches abgeholt werden
- Wenn kein VSYNC passierte blockt Funktion.
- Idee: Mittle Joystickdaten in langsamen Spielen

- 2 Abfrage Möglichkeiten:
  - sceCtrlReadBufferPositive (1 : Button gedrückt)
  - sceCtrlReadBufferNegative (0: Button gedrückt)

Joypad

- Ergebnis steht in SceCtrlData
- ```
  #include <ctrl.h>
  typedef struct SceCtrlData {
      SceUInt64 timeStamp;
      SceUInt32 buttons;
      SceUInt8 lx;
      SceUInt8 ly;
      SceUInt8 rx;
      SceUInt8 ry;
      SceUInt8 rsrv[16];
  } SceCtrlData;
  ```
- Sticks gehen von 0x00-0xff
- Buttons sind durch Konstanten definiert (siehe Doku)
  ```

Aufgabe 2

- Schreiben Sie ein erstes Mini-Spiel (Kerninteraktionsschleife)
- Initialisieren Sie das Pad

- Jedes Mal, wenn Sie das Pad lesen konnten
  - Testen Sie, ob der X-Button gedrückt worden ist.
  - Lesen Sie bei frisch gedrücktem Button den linken Joystick aus und geben Sie die Werte im -1,1 Format wieder.

Zusammenfassung

- Hardware der PSP
- Speicheraufbau und Verwendung der PSP
- Joypadabfrage auf der PSP