

Cell-Processor Computing

Benjamin Krill (HFU)

Furtwangen, 15. Dezember 2005

Übersicht

- 1 Das Cell Projekt
- 2 Processor
 - Technische Daten
 - Aufbau BE
 - Funktionsablauf
- 3 Konzepte
 - Job Queue
 - Self Multi-Tasking
 - Spufs
- 4 Anwendung
 - Programmierung
 - Sample Code
- 5 Einsatz
 - For Whom?
 - Momentane Projekte
- 6 Links

Übersicht

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

1 Das Cell Projekt

2 Processor

- Technische Daten
- Aufbau BE
- Funktionsablauf

3 Konzepte

- Job Queue
- Self Multi-Tasking
- Spufs

4 Anwendung

- Programmierung
- Sample Code

5 Einsatz

- For Whom?
- Momentane Projekte

6 Links

Das Cell Projekt

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- "Firmenkonglomerat", bestehend aus **Sony**, **Toshiba** und **IBM**
- seit 2000 – über 400 Ingenieure der 3 Firmen entwerfen und entwickeln den neuen Prozessor im STIDC in Austin, Texas
- Sony: Playstation 3
- Toshiba: Fernseher, Development-boards
- IBM: "Cell Processor based Blades"

Das Cell Konzept

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- Kompatibel mit der 64 Bit Power Architektur
- Erhöhte Leistung durch:
 - zusammenhängendes Prozessorsystem / verteiltes Rechnen
 - hohe Frequenz und kleine Arbeitsspannung
 - DMA-Architektur
- Mehrfach OS Fähigkeit (RTOS/non-RTOS)
 - Kombination zwischen real-time und non-real time Welt

Übersicht

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

1 Das Cell Projekt

2 Processor

- Technische Daten
- Aufbau BE
- Funktionsablauf

3 Konzepte

- Job Queue
- Self Multi-Tasking
- Spufs

4 Anwendung

- Programmierung
- Sample Code

5 Einsatz

- For Whom?
- Momentane Projekte

6 Links

Bestandteile des Cell-Chip

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- 1 Power Processor Element (PPE)
- 8 Synergistic Processor Elements (SPEs)
- Element Interconnect Bus (EIB) zwischen den Prozessoren
- Direct Memory Access Controller (DMAC)
- 2 Rambus XDR Memory Controller
- Rambus FlexIO (Input / Output) interface

Cell Architektur

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

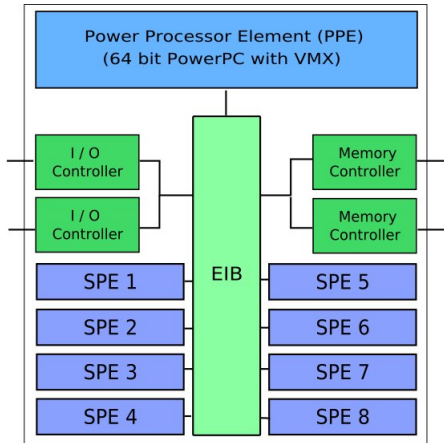
Sample Code

Einsatz

For Whom?

Momentane Projekte

Links



Cell Prozessor

- 9 bzw. 10 Prozessor-Kerne
- Größe: 221mm²
- Transistoren: 234 Millionen
- Fertigung: 90 Nanometer
- Takt: bis zu 4,6 GHz bei 1,3 Volt

Pentium 4 ExtremEdition

- 1 Core (HT)
- Transistoren: 125 Millionen
- Fertigung: 90 Nanometer
- Takt: 3,8 GHz

Cell-Chip

Cell-Processor Computing

Benjamin Krill

Das Cell Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

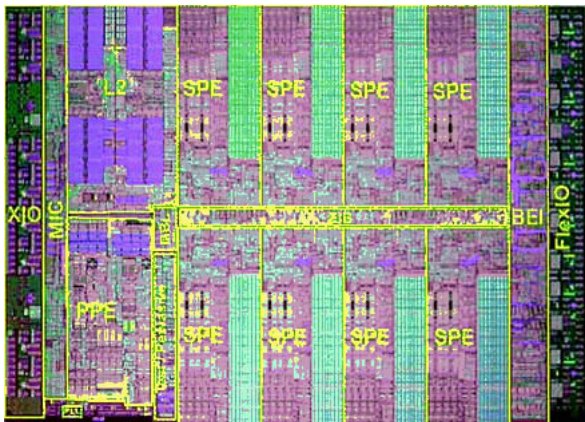
Sample Code

Einsatz

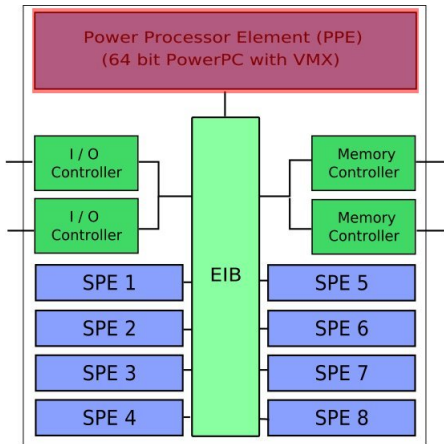
For Whom?

Momentane Projekte

Links



Power Processor Element (PPE)



Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

Power Processor Element (PPE)

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

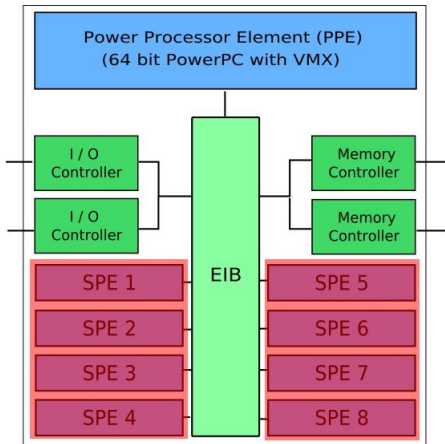
For Whom?

Momentane Projekte

Links

- herkömmlicher Dual-Threaded Mikroprozessorkern
- erweiterter PowerPC Befehlssatz
- Ausführung des Operation System
- besitzt VMX Vektor Befehle (bekannt als "AltiVec")
- kann die SPEs im Polling- oder Interrupt-Mode betrieben
- Verwaltung und Steuerung der SPEs (Threads)

Synergistic Processor Elements (SPEs)



Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

Synergistic Processor Elements (SPEs)

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

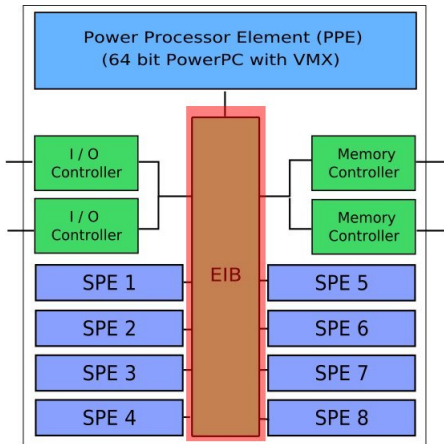
For Whom?

Momentane Projekte

Links

- 8 SPEs
- eine SPE ist ein unabhängiger und selbständiger Vektorprozessor (SIMD)
- kann im Pipeline Mode und/oder im Parallel Mode betrieben werden
- besitzt 128x 128Bit Register und 4 Gleitkomma Register
- besitzt 256 kByte lokalen Speicher für Code und Daten
- Kommunikation erfolgt über Mailbox, mit oder ohne Interrupt und DMA
- kein Paging auf dem lokalen Speicher

The Element Interconnect Bus (EIB)



Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

The Element Interconnect Bus (EIB)

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- 4x 128 Bit grosser Ringbus
- Optimiert für 1024 Bit Transferblocks
- 96 Byte pro Cycle
- SPE puffert und steuert den Transfer
(mehr SPEs erhöht die Latenzzeit)

Performance Übersicht

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

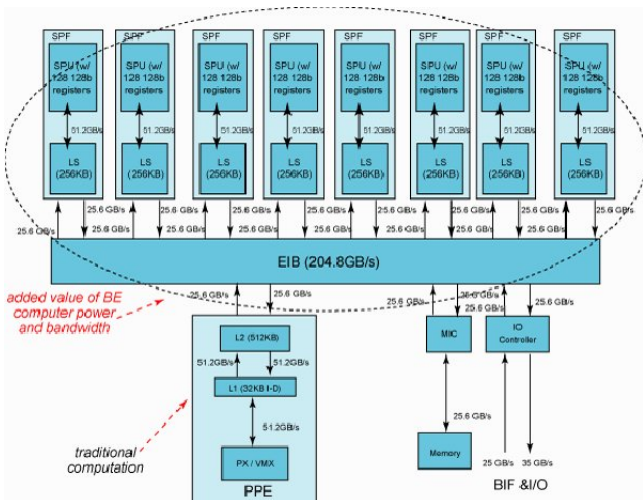
Sample Code

Einsatz

For Whom?

Momentane Projekte

Links



Beispiel Ablauf

- PPE startet SPE-Thread
(inclusive "Hochladen" des Binaries in jeweiligen LS)
- SPE bearbeitet die "Aufgabe"
- PPE bekommt nach der Abarbeitung der Aufgabe ein Interrupt
- PPE wertet den Interrupt und die Daten aus
- SPE steht für neue Aufgaben bereit

Übersicht

Cell- Processor Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

1 Das Cell Projekt

2 Processor

■ Technische Daten

■ Aufbau BE

■ Funktionsablauf

3 Konzepte

■ Job Queue

■ Self Multi-Tasking

■ Spufs

4 Anwendung

■ Programmierung

■ Sample Code

5 Einsatz

■ For Whom?

■ Momentane Projekte

6 Links

Verschiedene Konzepte

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- Job Queue
- Self multi-tasking of SPEs
- **Spufs** (Lösung von IBM)
Virtuelles Filesystem

Job Queue

Cell- Processor Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- mehrere SPEs sind für die Verarbeitung freigeschaltet
- PPE kontrolliert die Job Queue
- SPE Kernel lädt Jobs von der Queue
- Synrchonisation zwischen PPE und SPE

Job Queue

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

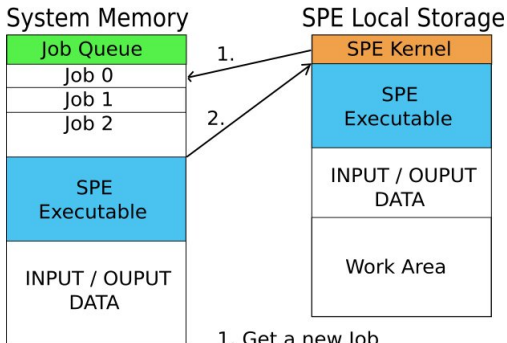
Sample Code

Einsatz

For Whom?

Momentane Projekte

Links



1. Get a new Job
2. Load the exectuable

Self Multi-Tasking of SPEs

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- SPE ist komplett selbständig
- Tasks, die geblockt sind, werden ausgetauscht (Taskswitch)
- gemeinsamer Speicher mit einer Queue im Hauptspeicher
- Synrchonisation zwischen den SPEs

Self Multi-Tasking

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

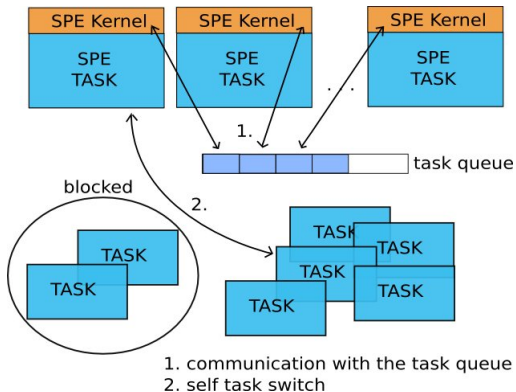
Sample Code

Einsatz

For Whom?

Momentane Projekte

Links



SPU - FileSystem

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- UNIX philosophy: "everything is a file"
- virtuelles FileSystem (ähnlich procfs, sysfs)
- realisiert durch Kernelmodul "spufs" (Patch für 2.6.13)
- per default in /spu gemountet (ähnlich /proc)

Bsp:

Cell-Processor Computing

Benjamin Krill

Das Cell Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

```
$ mkdir /spu/my-spu-12345
$ ls -lR /spu/
spu/:
total 0
drwxr-xr-x  2 root root 4096 Nov 11 22:25 my-spu-12345

spu/my-spu-12345:
total 0
-r--r-----  1 root root      0 Nov 11 22:25 ibox
-r--r-----  1 root root      0 Nov 11 22:25 mbox
-rw-rw----  1 root root 262144 Nov 11 22:25 mem
-rw-rw----  1 root root   2048 Nov 11 22:25 regs
-rw-rw----  1 root root      0 Nov 11 22:25 run
--w--w----  1 root root      0 Nov 11 22:25 wbox
```

/mem

- der **256KB** grosse, "software-gemapped" LokalStore
- Zugriff, wie auf reguläres Shared-Memory
- beinhaltet Code und Daten (max 256KB!)
- zur Zeit implementiert: read(), write(), mmap()

/run

- stump, der ioctl() ermöglicht
- einzige ioctl-Methode: spu_run()

/mbox

- erste SPE \Rightarrow PPE Mailbox
- readonly, 4B Einheiten, nur "non-blocking" Mode

/ibox

- zweite SPE \Rightarrow PPE Mailbox
- readonly, 4B, read im I/O-blocking Mode (pollen)

/wbox

- PPE \Rightarrow SPE Mailbox
- writeonly, 4B Einheiten
- blockiert, wenn voll

SPU - Beispielablauf

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- `mkdir /spu/myspe`
- `cat binary > /spu/myspu/mem`
- `dd if=/spu/myspu/run of=status bs=8 count=1`
- `tail -c 1234 /spu/myspu/mem > output`
- `rmdir /spu/myspu`

Übersicht

Cell- Processor Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

1 Das Cell Projekt

2 Processor

- Technische Daten
- Aufbau BE
- Funktionsablauf

3 Konzepte

- Job Queue
- Self Multi-Tasking
- Spufs

4 Anwendung

- Programmierung
- Sample Code

5 Einsatz

- For Whom?
- Momentane Projekte

6 Links

Hack den Cell!

Cell- Processor Computing

Benjamin Krill

Das Cell Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- Kernel mit Unterstützung für "spufs" (patches für 2.6.13 erhältlich)
- Compiler mit speziellem Backend
 - Toolchain für SPU erhältlich
 - Toolchain für PowerPC 64-Bit erhältlich
- Libraries (C, C++) — newlib für SPU erhältlich
- Emulator erhältlich (Mambo)

Cell Mambo

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten
Aufbau BE
Funktionsablauf

Konzepte

Job Queue
Self Multi-Tasking
Spufs

Anwendung

Programmierung
Sample Code

Einsatz

For Whom?
Momentane Projekte

Links

The screenshot displays the Cell Mambo software interface. The top window shows system information and a memory dump. The system information includes:

```
*****
CPBM Starting. Check Boot ROM...OK. FW is [Aug 9 2005 10:10:11]

SYSTEM INFORMATION
Processor = BE DD2.0
Timebase = 24000 khz (internal)
Config = SIMULATOR
Config = SMP disabled
SMP Size = 1 (2 threads)
Memory = 512MB (BE0: 512MB)
```

The memory dump window shows a table of memory addresses and their contents:

PC	00000000111047C	1d6	r31_Rev29	328	0000000000000000
LB	000000001112000	078	0000000011120470	328	0000000000000000
SPU 0	000000001114200	0	0000000000000000	16	000000001390000
1	000000001114200	1	0000000000000000	17	0000000000000000
2	000000001114200	2	FFFFFFFFFFFFFFFF	18	0000000000000000
3	000000001114200	3	FFFFFFFFFFFFFFFF	19	0000000000000000
4	000000001114200	4	FFFFFFFFFFFFFFFF	20	0000000000000000
5	000000001114200	5	FFFFFFFFFFFFFFFF	21	0000000000000000
6	000000001114200	6	FFFFFFFFFFFFFFFF	22	0000000000000000
7	000000001114200	7	FFFFFFFFFFFFFFFF	23	0000000000000000
8	000000001114200	8	FFFFFFFFFFFFFFFF	24	0000000000000000
9	000000001114200	9	FFFFFFFFFFFFFFFF	25	0000000000000000
10	000000001114200	10	FFFFFFFFFFFFFFFF	26	0000000000000000
11	000000001114200	11	FFFFFFFFFFFFFFFF	27	0000000000000000
12	000000001114200	12	FFFFFFFFFFFFFFFF	28	0000000000000000
13	000000001114200	13	FFFFFFFFFFFFFFFF	29	0000000000000000
14	000000001114200	14	FFFFFFFFFFFFFFFF	30	0000000000000000
15	000000001114200	15	FFFFFFFFFFFFFFFF	31	0000000000000000

The bottom window shows the Mambo GUI with a file tree on the left and a control panel on the right. The file tree includes:

- System
- FFCD0
 - PCTrack
 - FFCDBase
 - GFFlags
 - FFFlags
 - PCAddressing
- FFCD1
 - SFU0
 - SFU1
 - SFU2
 - SFU3
 - SFU4
 - SFU5
 - SFU6
 - SFU7
- Load-Elf-App
- Load-Elf-Kernel
- MemoryMap
- SystemMemory

The control panel shows CPU Cycles: 23,899,999 and various buttons for simulation control.

Cell Mambo

Cell- Processor Computing

Benjamin Krill

Das Cell Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- Systemsimulator für Cell
- Zwei Modi
 - Cycle-genau
 - Funktionalitäts-genau
- Debug-Kontrollfunktionen (z.B. Breakpoints, Register)
- Zugriffe auf Console und Disk (Bogus)

PPE DMA Code

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <libspe.h>
4
5 int main(int argc, char* argv[])
6 {
7
8     spe_program_handle_t *binary;
9     speid_t threadnum;
10    int status;
11
12    static int test_result __attribute__((aligned (128)));
13
14    /* DMA - Write test. */
15    binary = spe_open_image("spe-dma-write");
16    if (!binary)
17        exit(2);
18
19    printf("Test result Addr: %p\n",&test_result);
20
21    threadnum = spe_create_thread(0, binary, &test_result, NULL, 0, 0);
22
23    spe_wait(threadnum,&status,0);
24
25    printf("Test result: %04x\n",test_result);
26
27    spe_close_image(binary);
28
29    return 0;
30
31 }
```

SPE DMA Code

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

```
1 #include "spu_intrinsics.h"
3 typedef union {
4     unsigned long long    ull;
5     unsigned int        vp[2];
6 } addr64;
7
8 int result __attribute__((aligned (128)));
9
10 int main ( int spuid, addr64 argp, addr64 envp)
11 {
12     unsigned long long argAddress = argp.ull;
13
14     /* Write to specified address */
15     result = 42;
16
17     /* dma write */
18     spu_mfcdma32(&result, argAddress, 4, 0,
19                 (((0) << 24) | ((0) << 16) | (0x20)));
20     /* set mfc tagmask */
21     __builtin_si_wrch((22), __builtin_si_from_uint(1));
22     /* wait for all outstanding mfc tags */
23     spu_mfcstat(0x2);
24
25     /* Done */
26     return 0;
27 }
```

Übersicht

Cell- Processor Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

1 Das Cell Projekt

2 Processor

■ Technische Daten

■ Aufbau BE

■ Funktionsablauf

3 Konzepte

■ Job Queue

■ Self Multi-Tasking

■ Spufs

4 Anwendung

■ Programmierung

■ Sample Code

5 Einsatz

■ For Whom?

■ Momentane Projekte

6 Links

Wem nutzt der Cell?

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- hoch parallelisierte, hoch vektorisierte Anwendungen, mit hunderten von Millionen von "kleinen cellularen" Berechnungen
- **game programming**, Video- und Audio- De- und Encoding, De- und Encryption und allgemeine Berechnungen, sowie deren Visualisierung
- Physik, Biologie (Genforschung), Chemie, Meteorologie, Metrologie

Momentane Projekte

Cell- Processor Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- Sony PS3 (erhältlich Ende 2006)
- IBM CPBW (Cell Prozessor Based Workstation)
- Toshiba Dev-Board (2006)
- ... abwarten was kommt

Übersicht

Cell-
Processor
Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

1 Das Cell Projekt

2 Processor

■ Technische Daten

■ Aufbau BE

■ Funktionsablauf

3 Konzepte

■ Job Queue

■ Self Multi-Tasking

■ Spufs

4 Anwendung

■ Programmierung

■ Sample Code

5 Einsatz

■ For Whom?

■ Momentane Projekte

6 Links

Links

Cell- Processor Computing

Benjamin Krill

Das Cell
Projekt

Processor

Technische Daten

Aufbau BE

Funktionsablauf

Konzepte

Job Queue

Self Multi-Tasking

Spufs

Anwendung

Programmierung

Sample Code

Einsatz

For Whom?

Momentane Projekte

Links

- <http://www-128.ibm.com/developerworks/power/cell/>
- <http://www-128.ibm.com/developerworks/library/pa-cell/>
- <http://www.bsc.es/projects/deepcomputing/linuxoncell/>
- <http://www.research.ibm.com/people/ashwini/cell%20blades.html>
- <http://www.research.scea.com/research/html/CellGDC05/>
- <http://www.the-cell-chip.de/>