

Netgauge auf Cell BE

Sven Eckelmann

TU Chemnitz

15. Oktober 2007



TECHNISCHE UNIVERSITÄT
CHEMNITZ

Inhaltsverzeichnis

- 1 Einführung
 - Aufgabe
 - Netgauge
 - Cell BE
- 2 Implementierung
 - SPU-Module
 - Übertragungsarten
 - SPE initialisieren
 - SPU-Programme
 - Häufige Fehler
- 3 Auswertung

Aufgabe

Netgauge

+

Cell

=

Messung Speichertransfers von Cell BE

Aufgabe

Netgauge

+

Cell

=

Messung Speichertransfers von Cell BE

Aufgabe

Netgauge

+

Cell

=

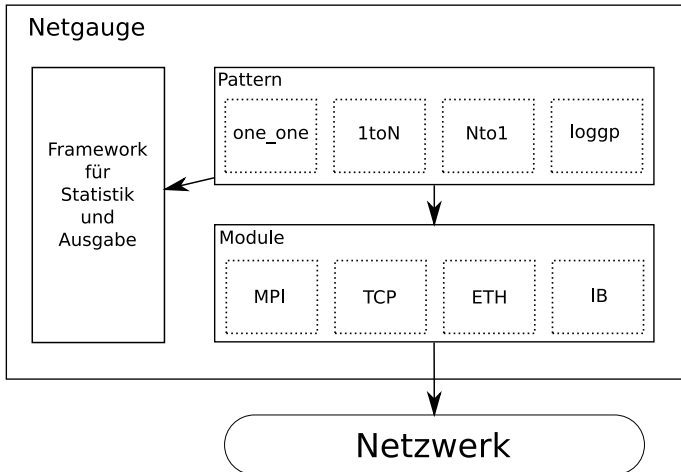
Messung Speichertransfers von Cell BE

Netgauge

- “Nachfolger” von Netpipe
 - hatte nur Ping-Pong-Test
 - schlechte Zeitmessung
 - viel Code innerhalb von Messungen
- Verschiedene Module für Transfer
- mehrere Tests
- genaue Zeitmessung

Netgauge

Aufbau

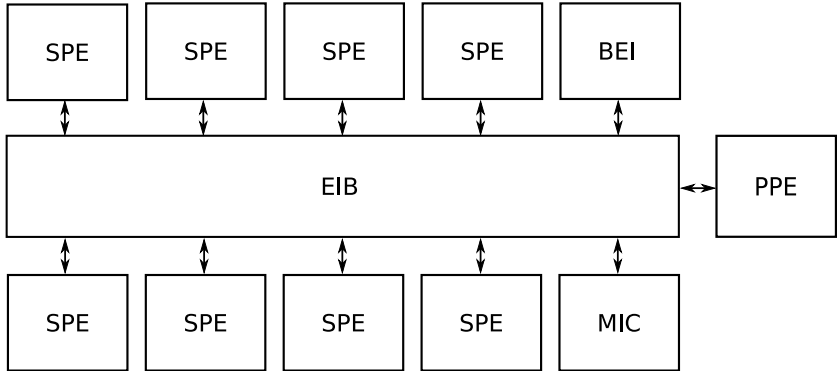


Cell BE

- Entwicklung von IBM, Toshiba, Sony
- Einsatz in Multimedia-Geräten (HDTV, Playstation 3)
- “Echte Hardware”: Steckkarten, Cell-Blades

Cell BE

Aufbau



Cell BE

SPE-Aufbau

- Memory Flow Controller
 - Local Store (256 KB Speicher)
 - Synergistic Processing Unit
 - SIMD-Befehle für 128-Bit-Daten
- Vortrag Simon Wunderlich

Cell BE

SPE-Aufbau

- Memory Flow Controller
- Local Store (256 KB Speicher)
- Synergistic Processing Unit
 - SIMD-Befehle für 128-Bit-Daten
 - Vortrag Simon Wunderlich

SPU-Module

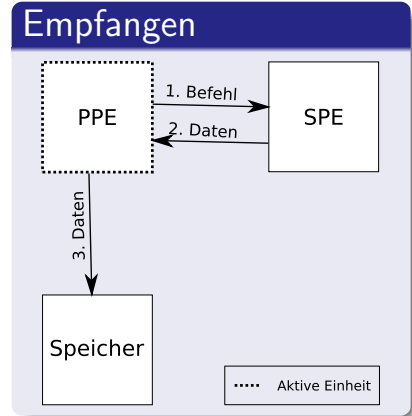
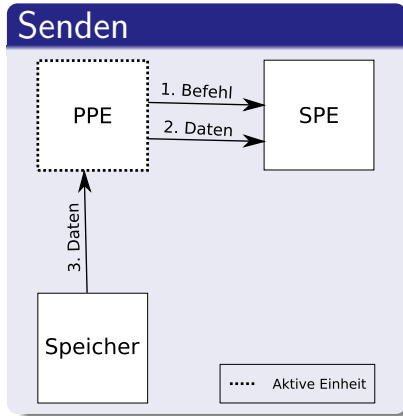
`cell` PPE übernimmt Speicherzugriff

`cell_mail` Transfer über Mailbox

`cell_dma` SPE greift auf Speicher zu

`cell_dmalist` SPE greift über Listen auf Speicher zu

PPE kopiert Daten in Local Store



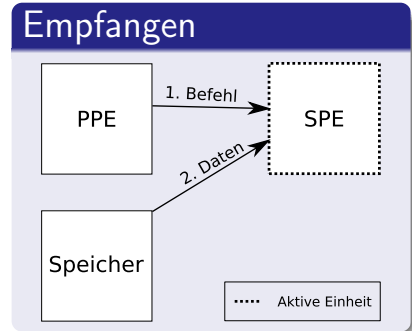
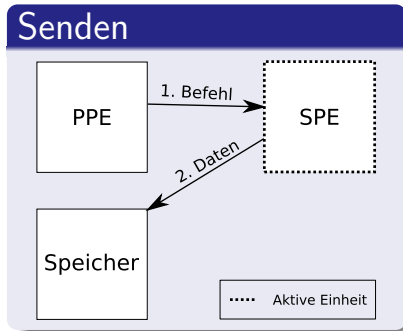
PPE kopiert Daten in Local Store

Beispiel Implementierung

Beispiel

```
void* local_store =  
    spe_ls_area_get(spu_context);  
....  
memcpy(local_store + addr, buffer,  
    size);
```

DMA-Operation durch SPE



DMA-Operation durch SPE

Beispiel Implementierung

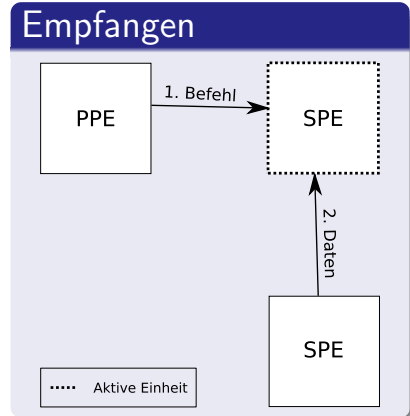
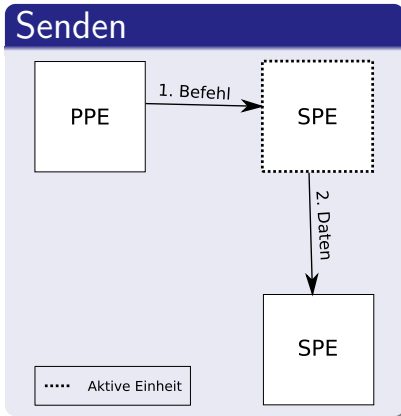
Beispiel

```
spu_mfcdma64((void*) &buffer ,  
             external_buffer_high ,  
             buffer_low , size ,  
             tag , MFC_GET_CMD);
```

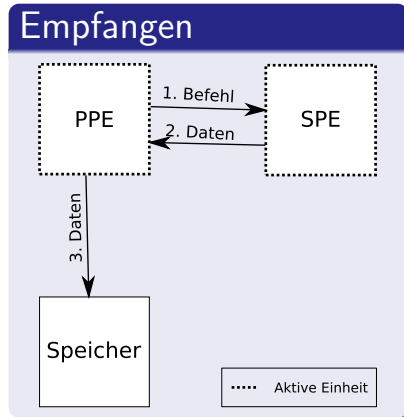
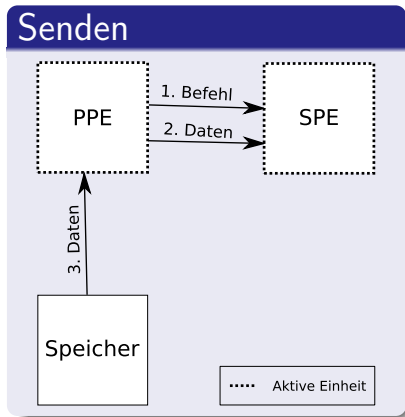
```
mfc_write_tag_mask(1 << tag );  
mfc_read_tag_status_all();
```


DMA-Operation durch SPE

Zwischen zwei SPEs



Mailbox zwischen SPE und PPE



Mailbox zwischen SPE und PPE

Beispiel Implementierung

Beispiel PPU

```
uint32_t data;  
spe_in_mbox_write(spu_context ,  
    (void*)&data ,  
    1, SPE_MBOX_ALL_BLOCKING);
```

Beispiel SPU

```
uint32_t data = spu_read_in_mbox();
```

SPE initialisieren

- 1 Erstelle Thread-Kontext
- 2 Erstelle SPU Kontext
- 3 Lade SPU-Programm
- 4 Starte SPU-Programm in Thread
- 5 Affinität festlegen
- 6 Informationen über Puffer versenden

SPE initialisieren

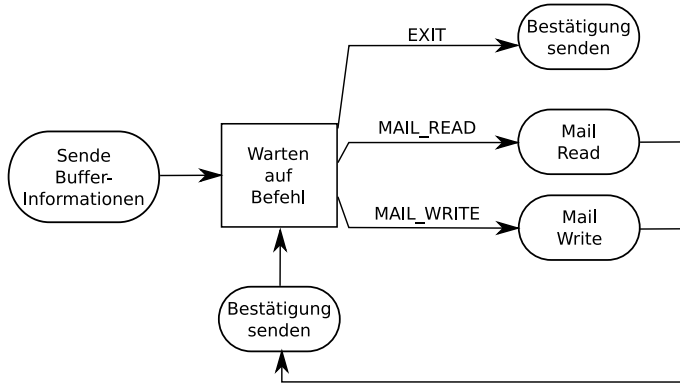
- 1 Erstelle Thread-Kontext
- 2 Erstelle SPU Kontext
- 3 Lade SPU-Programm
- 4 Starte SPU-Programm in Thread
- 5 Affinität festlegen
- 6 Informationen über Puffer versenden

SPE-Übertragung initialisieren

- 1 Erstelle Thread-Kontext
- 2 Erstelle SPU Kontext
- 3 Lade SPU-Programm
- 4 Starte SPU-Programm in Thread
- 5 Affinität festlegen
- 6 Informationen über Puffer versenden

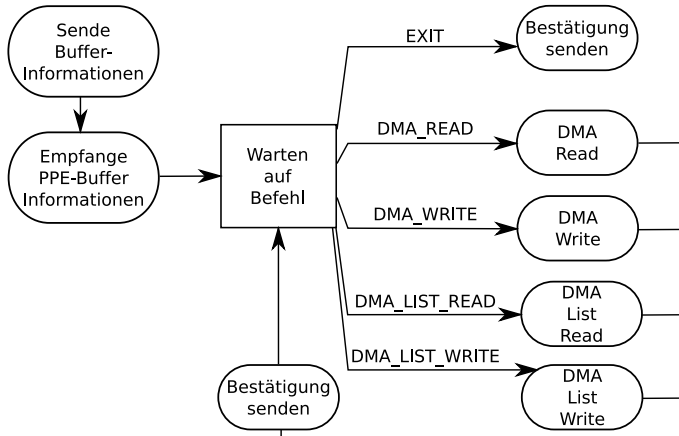
SPU-Programme

mod_cell_spu



SPU-Programme

mod_cell_dma_spu



Häufige Fehler

Bus Error is not your friend

- Unterschiede bei Datentypen
 - PPU-void* (32-/64-Bit), SPU-void* (32-Bit)
 - PPU-long (32-/64-Bit), SPU-long (32-Bit)
 - char unsigned!!!
- DMA-Transfers
 - Größe: 1, 2, 4, 8 Byte oder vielfaches von 16
 - Alignment: 16 Byte
- Kein impliziter Zugriff auf Hauptspeicher

Häufige Fehler

Bus Error is not your friend

- Unterschiede bei Datentypen
 - PPU-void* (32-/64-Bit), SPU-void* (32-Bit)
 - PPU-long (32-/64-Bit), SPU-long (32-Bit)
 - char unsigned!!!
- DMA-Transfers
 - Größe: 1, 2, 4, 8 Byte oder vielfaches von 16
 - Alignment: 16 Byte
- Kein impliziter Zugriff auf Hauptspeicher

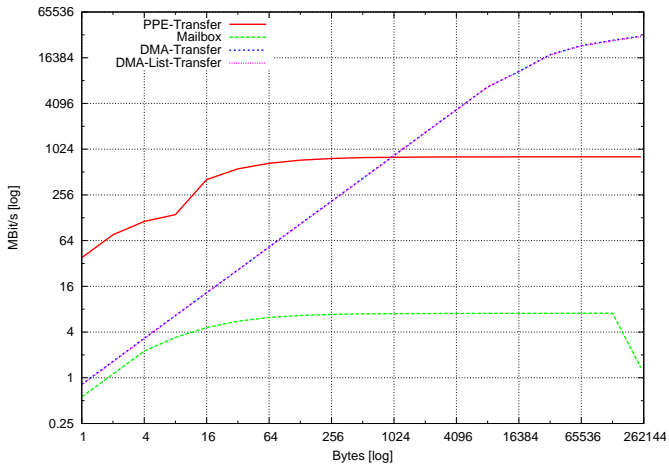
Häufige Fehler

Bus Error is not your friend

- Unterschiede bei Datentypen
 - PPU-void* (32-/64-Bit), SPU-void* (32-Bit)
 - PPU-long (32-/64-Bit), SPU-long (32-Bit)
 - char unsigned!!!
- DMA-Transfers
 - Größe: 1, 2, 4, 8 Byte oder vielfaches von 16
 - Alignment: 16 Byte
- Kein impliziter Zugriff auf Hauptspeicher

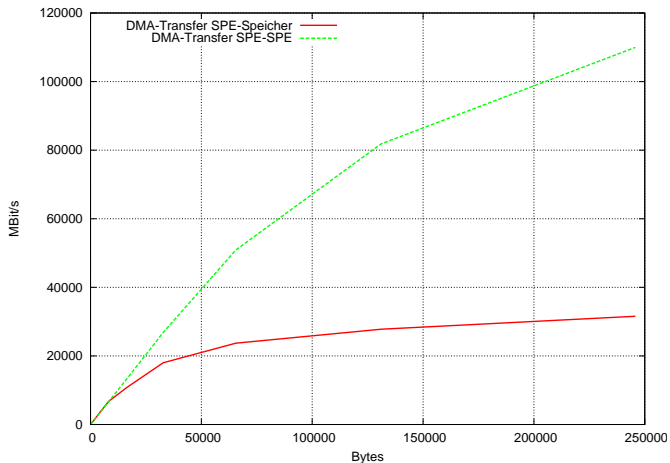
Auswertung

Transferraten zwischen Speicher und SPE



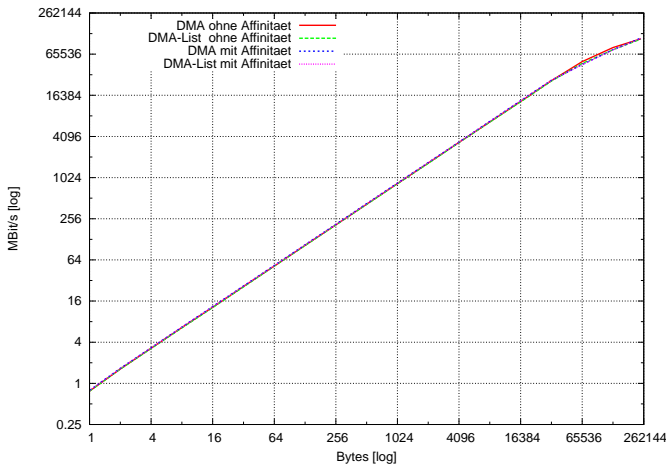
Auswertung

Transferraten abhängig vom Gegenstelle



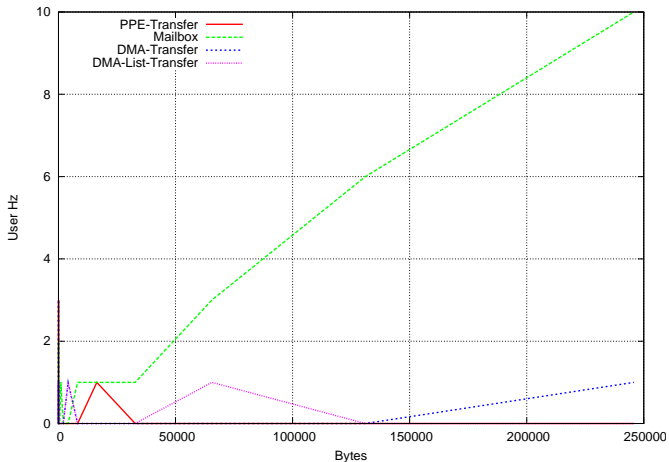
Auswertung

Transferraten zwischen zwei SPEs mit und ohne Affinitäten



Auswertung

Verbrachte Zeit in Hardware-/Software-Interrupts



Literatur

- <http://www.ibm.com/developerworks/power/cell/>
- <http://www.unixer.de/research/netgauge/>
- <http://blog.perlplexity.org/>

Fragen?