THREE LETTER WORDS

FMC and its weird friends

Alessandro Rubini, Federico Vaga
Independent consultants in Pavia, Italy.
Working for CERN "hardware and timing" group
We created FMC. And we saw it was good.

- We support SPEC
- We'll soon support SVEC
- We offer a "fake-dev" device
  - Drivers: trivial, write-eeprom,
  - Drivers: chardev, wr-nic
  - Drivers: fine-delay, adc, tdc (Samuel)
We need to comply with IMPI-FRU standard
  - Bytes 0 to 200-400 of the FMC EEPROM are lost
  - The identifiers are (can be) string-based

fmc-bus uses such strings to match device and driver
  - We left an option to force the bus to always match

We need to stuff other data in the EEPROM
  - WR calibration data
  - fine-delay calibration data
  - ADC calibration data

So we chose to place a filesystem in the EEPROM
  - And we saw it was good.
SDB: self describing bus

Born to make sense of the FPGA internals
  • Idea by Wesley and me
  • First implementation by Manohar
  • Current implementation by Wesley
  • Now cast in stone

Recursive structure, with 64-byte records
  • A "bridge" routes to an sdb-described sub-bus
  • All addresses are relative
  • Extensible format (we have a "type" byte)
  • 64-bit vendor ID, 32-bit device ID, 64-bit address
  • 19-byte UTF-8 component name
SDBFS: Use SDB in the EEPROM/Flash

```
spusa$ fru-generator -v CERN -n FmcDelay1ns4cha -s proto-0 \
       -p EDA-02267-V3 > IPMI-FRU

spusa$ fru-dump /lib/firmware/fdelay-eeprom.bin
/lib/firmware/fdelay-eeprom.bin: manufacturer: CERN
/lib/firmware/fdelay-eeprom.bin: product-name: FmcDelay1ns4cha
/lib/firmware/fdelay-eeprom.bin: serial-number: proto-0
/lib/firmware/fdelay-eeprom.bin: part-number: EDA-02267-V3

spusa$ ls -l
-rw-rw-r-- 1 rubini staff 975 Nov 19 18:08 --SDB-CONFIG--
-rw-rw-r-- 1 rubini staff 216 Nov 19 18:13 IPMI-FRU
-rw-rw-r-- 1 rubini staff 11 Nov 19 18:04 fd-calib
-rw-rw-r-- 1 rubini staff 7 Nov 19 18:04 name

spusa$ sudo gensdbfs . /lib/firmware/fdelay-eeprom.bin

spusa$ sdb-read -l -e 0x100 /lib/firmware/fdelay-eeprom.bin
46696c6544617461:2e202020 @ 00000100-000018ff .
46696c6544617461:6e616d65 @ 00000200-0000206 name
46696c6544617461:66642d63 @ 00001800-000018ff fd-calib
46696c6544617461:49504d49 @ 00000000-000000d7 IPMI-FRU
```
SDBFS: Mount the FPGA and the EEPROM

spusa.root# insmod sdbfs/kernel/sdbfs.ko
spusa.root# insmod kernel/fmc-fine-delay.ko
spusa.root# mount -t sdbfs none /mnt

spusa.root# ls -l /mnt
-rwxr-xr-x 1 root root 8192 Jan 1 1970 fdelay-i2c/
-rwxr-xr-x 1 root root 1048576 Jan 1 1970 fdelay-iomem/

spusa.root# find /mnt

/mnt/fdelay-iomem
/mnt/fdelay-iomem/WB4-Crossbar-GSI
/mnt/fdelay-iomem/Fine-Delay-Core
/mnt/fdelay-iomem/WB4-Bridge-GSI
/mnt/fdelay-iomem/WB4-Bridge-GSI/WB4-Crossbar-GSI
/mnt/fdelay-iomem/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-BlockRAM
/mnt/fdelay-iomem/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Crossbar-GSI
/mnt/fdelay-iomem/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Crossbar-GSI/WB4-Endpoint
/mnt/fdelay-iomem/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Endpoint/WB4-Soft-PLL
/mnt/fdelay-iomem/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Endpoint/WB4-PPS-Generator
/mnt/fdelay-iomem/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Periph-Syscon
/mnt/fdelay-iomem/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Periph-UART
/mnt/fdelay-iomem/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Bridge-GSI/WB4-Periph-1Wire
spusa.root# devmem2 0xfd3e0134
0xcafebabe

spusa.root# /morganahome/rubini/wip/fmc-bus/tools/fmc-mem \
/mnt/fdelay-iomem/WB4-Bridge-GSI/WB4-Bridge-GSI/WR-Endpoint 34
cafebabe

spusa.root# /morganahome/rubini/wip/fmc-bus/tools/FRU-dump \
/mnt/fdelay-i2c/IPMI=FRU
/mnt/fdelay-i2c/IPMI=FRU: manufacturer: CERN
/mnt/fdelay-i2c/IPMI=FRU: product-name: FmcDelay1ns4cha
/mnt/fdelay-i2c/IPMI=FRU: serial-number: proto-0
/mnt/fdelay-i2c/IPMI=FRU: part-number: EDA-02267-V3
ZIO: the ultimate I/O framework

High data rate
Hardware timestamps (better than 1ns precision)
Big data blocks (stripes of many samples)
Off-line creation/gathering of data blocks
Easy monitoring of a diverse I/O environment
Support for several (many) boards of the same type
## ZIO: The Actors

<table>
<thead>
<tr>
<th>Block</th>
<th>Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Yellow</td>
<td>Luser</td>
</tr>
<tr>
<td>D</td>
<td>Yellow</td>
<td>Lemon</td>
</tr>
<tr>
<td>bl</td>
<td>Green</td>
<td>Fops</td>
</tr>
<tr>
<td></td>
<td>Green</td>
<td>Forest</td>
</tr>
<tr>
<td>S</td>
<td>Pink</td>
<td>Socket</td>
</tr>
<tr>
<td></td>
<td>Pink</td>
<td>Salmon</td>
</tr>
<tr>
<td>B</td>
<td>Brown</td>
<td>Buffer</td>
</tr>
<tr>
<td></td>
<td>Brown</td>
<td>Brown</td>
</tr>
<tr>
<td>T</td>
<td>Red</td>
<td>Trigger</td>
</tr>
<tr>
<td></td>
<td>Red</td>
<td>Tomato</td>
</tr>
<tr>
<td>P</td>
<td>Purple</td>
<td>Periph.</td>
</tr>
<tr>
<td></td>
<td>Purple</td>
<td>Purple</td>
</tr>
<tr>
<td>N</td>
<td>Neutral</td>
<td>Network</td>
</tr>
</tbody>
</table>

- **The block is overall blue**: C
- **Control**: C
- **Data**: D
### ZIO: The New Control

<table>
<thead>
<tr>
<th>V</th>
<th>v</th>
<th>A</th>
<th>a</th>
<th>sequence</th>
<th>nsamples</th>
<th>ssize</th>
<th>nbits</th>
</tr>
</thead>
<tbody>
<tr>
<td>fam</td>
<td>type</td>
<td>host-identification</td>
<td>device-id</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cset</td>
<td>chan</td>
<td>device name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>0x30</strong></td>
<td>tstamp: secs</td>
<td>tstamp: ticks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>0x40</strong></td>
<td>tstamp: bins</td>
<td>mem-addr</td>
<td>reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>0x50</strong></td>
<td>flags</td>
<td>trigger name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>0x60</strong></td>
<td>TLV record for optional extra information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This area hosts attributes for the device and for the currently active trigger.

Device and trigger are each characterized by 16 "standard" attrs and 32 "extended" attrs. A bit-mask states which attrs are active.

Each attribute is a 32-bit word
ZIO: The Usual pipeline

write → alloc_block → push_block → raw_io

store_block → retr_block → data_done

free_block

read → retr_block → pull_block → raw_io

free_block

alloc_block

store_block
Mapping Socket Types to ZIO

We map the three standard socket types to ZIO blocks:

- The code is implemented as a ZIO buffer
- Triggers and Peripheral drivers are unaffected
The ZIO pipeline, with zio-buf-sock active
Communication Paths Within a Host
device: zio-zero (input and output)
device: zio-loop (for stress-testing and diagnostics)
device: line discipline (input: UART or pty for stress-test)
device: GPIO (input and output)
device: AD7888/AD7887 (SPI ADC)
device: fmc-based TDC/DTC
device: fmc-fine-delay (input and output: 10ps resolution)
device: fmc-based 100MS ADC
    trigger: kernel timer
    trigger: high-resolution timer
    trigger: transparent trigger (user/device driven)
    trigger: external interrupt or external GPIO
buffer: "kmalloc"
buffer: "data" (SOCK_STREAM alike, coalescing blocks)
buffer: "vmalloc" (mmap-capable)
    sockets: SOCK_DGRAM and SOCK_RAW (sock STREAM almost working)
tools: zio-dump (control and data)
tools: zio-cat-file (demonstrating mmap for input channels)
tools: pfzio-send and pfzio-receive (like netcat)
Thank you for your attention