Managing Multiple Function Generators for FAIR

S. Rauch, M. Thieme, R. Bär, GSI Centre for Heavy Ion Research, Darmstadt

Abstract

In the FAIR control system, equipment which needs to be controlled with ramped nominal values (e.g. power converters) is controlled by a standard front-end controller called scalable control unit (SCU). An SCU combines a ComExpressBoard with Intel CPU and an FPGA baseboard and acts as bus-master on the SCU host-bus. Up to 12 function generators can be implemented in slave-board FPGAs and can be controlled from one SCU.

The real-time data supply for the generators demands a special software/hardware approach. Direct control of the generators with a FESA (front-end control software architecture) class, running on an Intel Atom CPU with Linux, does not meet the timing requirements. So an extra layer with an LM32 soft-core CPU is added to the FPGA. Communication between Linux and the LM32 is done via shared memory and a circular buffer data structure. The LM32 supplies the function generators with new parameter sets when it is triggered by interrupts. This two-step approach decouples the Linux CPU from the hard real-time requirements. For synchronous start and coherent clocking of all function generators, special pins on the SCU backplane are being used to avoid bus latencies.

SCU

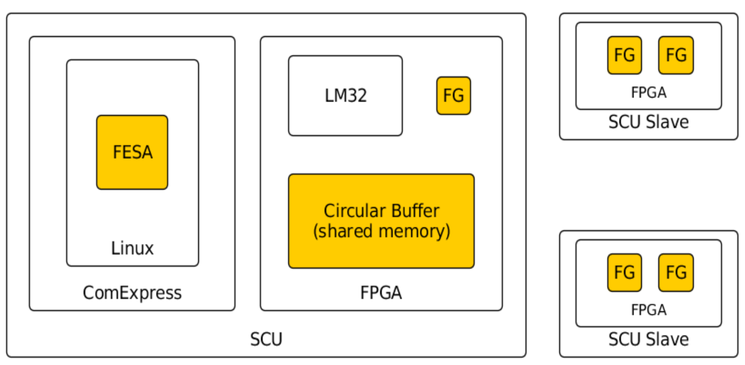
- standard frontend controller for FAIR
- FPGA board with ComExpress CPU attached
- Controls parallel bus called SCU bus
- SoC design in the FPGA on basis of Wishbone
- LM32 cluster connected via wishbone crossbar
- MSI wishbone crossbar with LM32 and PCIe bridge as targets

FG

- Function generator macro written in VHDL
- Can be used in SCU slaves cards or with wishbone interface
- Slave cards: DIOB (1 FG), ADDAC1/2 (2 FGs)
- Digital output with 32 Bit on DIOB card
- Analog output with 16 Bits on ADDAC card
- Is configured with a parameter set and interpolates for n steps

Data Supply with Real Time Boundaries

- Interpolation from 250 to 32000 steps
- Sample frequency 16 kHz up to 1 Mhz
- Worst case 250 steps with 1 Mhz sample rate: 250 µs
- Data rate too high for linux, to service 12 channels



Overview of SCU with slaves

Implementation with LM32 and MSI

- Add real time layer with LM32
- No operating sytem and software written in C
- Real time behaviour is easy to predict
- Real time processing done in IRQ handler
- Circular buffer in shared memory
- IRQ round trip 5 µs, SCU bus access 300 ns

- With a step width of 250, 13 FGs could be serviced
- Software Interrupt scheme from linux with MSI for messages

FG Operation

- LM32 software scans for FGs and enumerates them
- Up to 12 Fgs supported
- Presented as virtual devices to linux
- Circular buffer filled with parameters by linux, emptied by IRQ handler
- IRQ signalling is done via MSI

Project Status

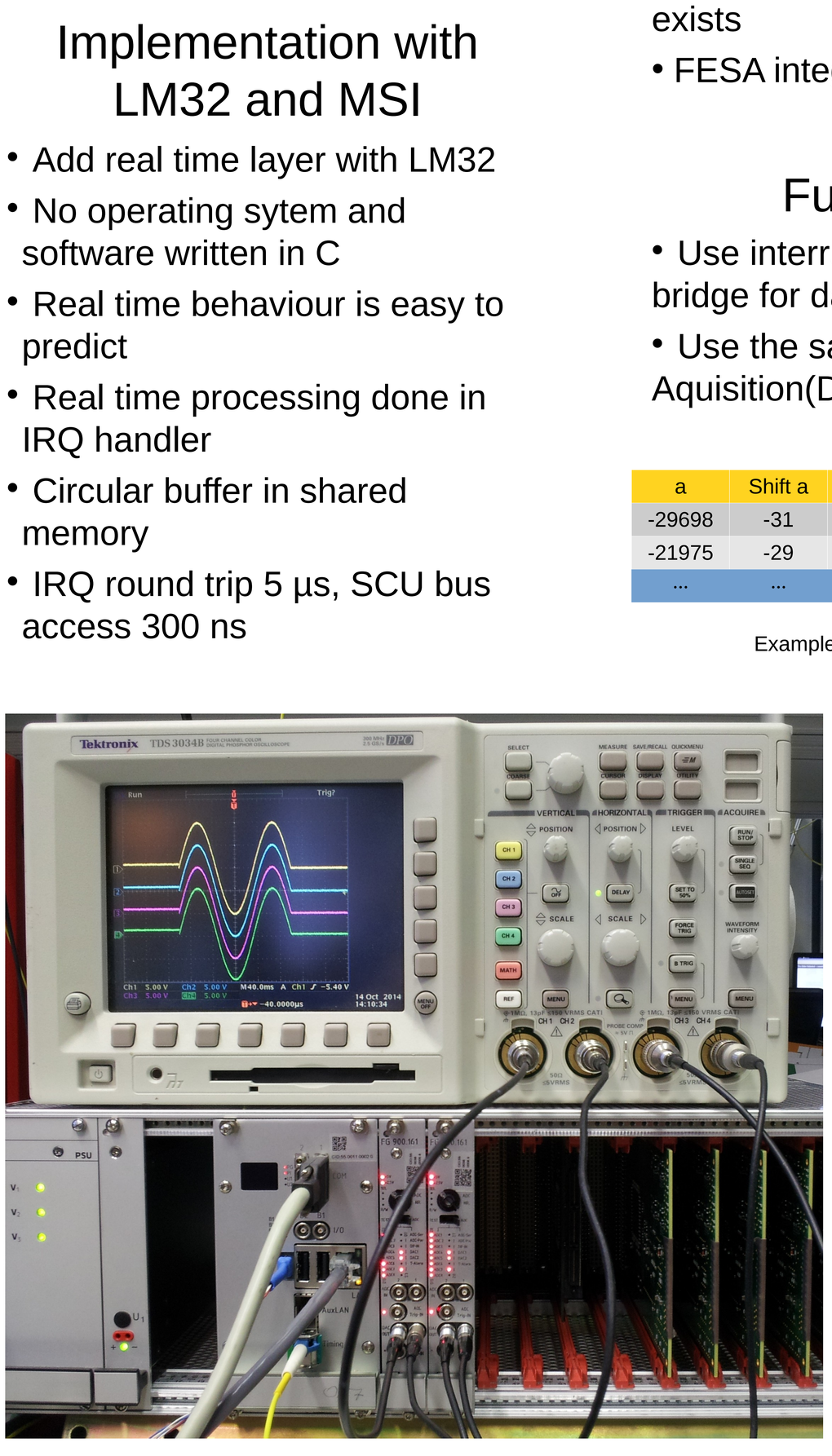
- Hardware is implemented
- Test program for the FG system exists
- FESA integration ongoing

Future Work

- Use interrupts to the PCIe bridge for data transfer
- Use the same system for Data Aquisition(DAQ)

a	Shift a	b	Shift b	c	Steps
-29698	-31	25318	-13	0	2
-21975	-29	25201	-13	3084	2
...	...	...	...	...	...

Example parameter set for a sine wave



Example with 4 channel sine wave generated by the FG.

Shown is an SCU with two ADDAC cards. The DAC outputs are connected to an oscilloscope.