Integration of FPGAs into SDR via Memory-Mapped I/O

Matt Carrick

SDR Forum ‘09

Outline

- ***Introduction***
- FPGA Interface
- Integration into Core Framework
- Profiling and Results
- Conclusion

Major Software Radio Components

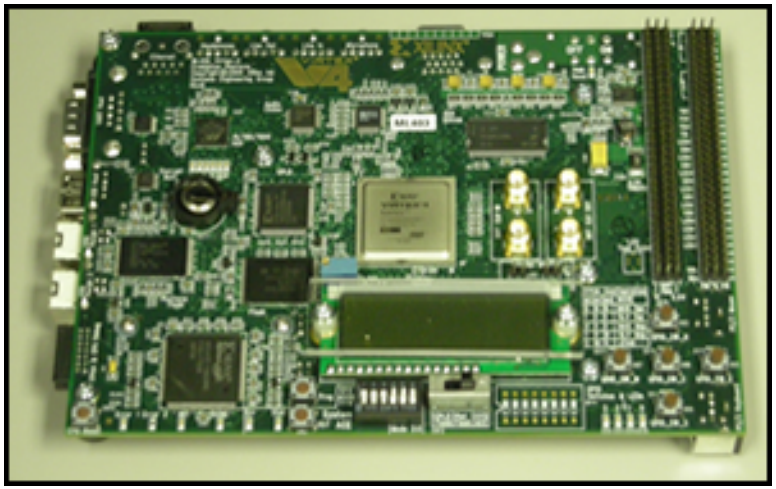
- Processing Blocks
  - GPP, DSP, FPGA, ASIC
  - (Software Architecture)
- Processors must:
  - Conform to software radio concepts
  - Trade between flexibility and performance
- FPGAs provide necessary throughput with some flexibility

FPGA Integration

- Integrate FPGA in both hardware and software
- Mechanisms for integration already exist
  - Memory Mapped I/O
- Additional layers of abstraction reduce throughput and effectiveness of FPGA

ML403 Platform

- PowerPC405D5 Processor
  - SCA CoreFramework (OSSIE)
- Virtex-4 FX12 FPGA
  - Signal Processing Circuit (FIR)



OSSIE

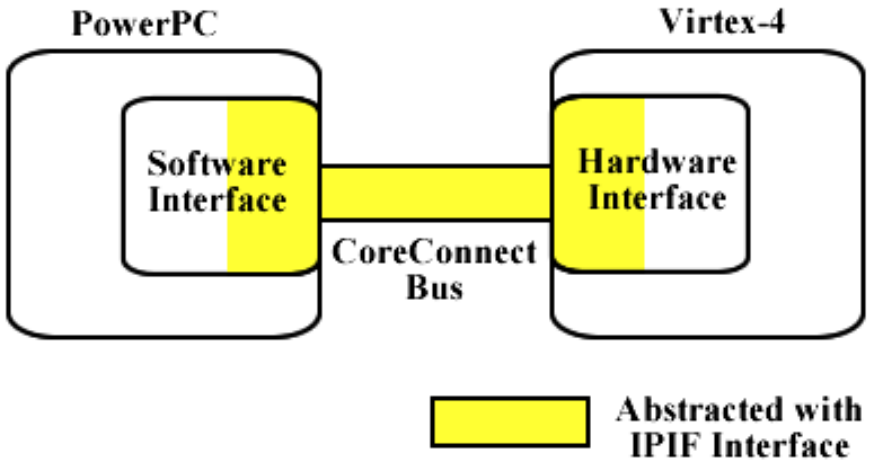
- Open-source implementation of the SCA
- Developed in the spirit of the SCA
- Includes Core Framework and Tool Suite

Outline

- Introduction
- ***FPGA Interface***
- Integration into Core Framework
- Profiling and Results
- Conclusion

FPGA Interface

- Software Portion
  - Memory Mapped I/O
  - *((unsigned long *) WRITE_ADDR) = value;



Communicating With the FPGA

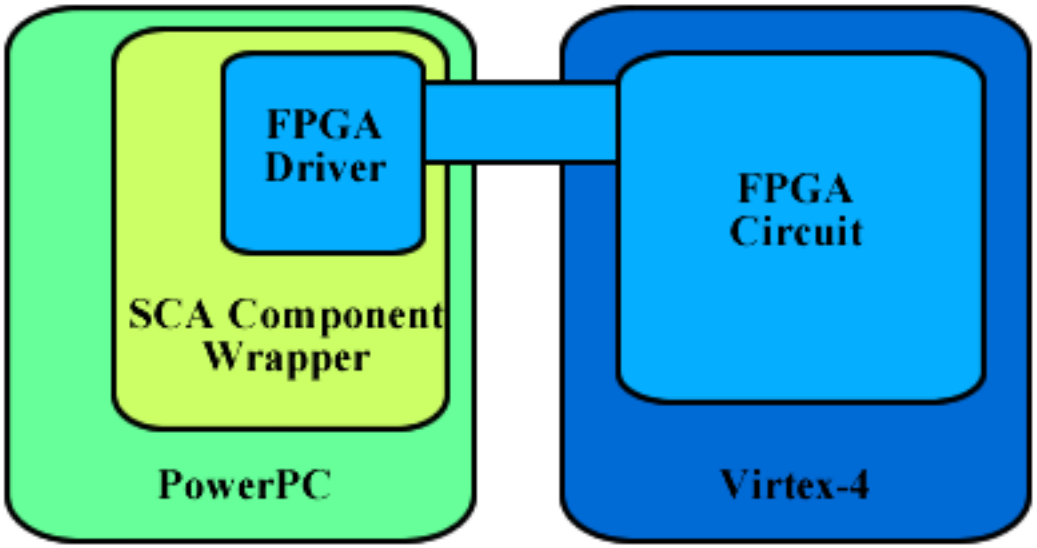
- FPGA IPIF Interface
  - Toggling Bus Lines (Read, Write, Data, etc.)
  - CoreConnect Bus
  - Xilinx IPIF Interface

Outline

- Introduction
- FPGA Interface
- ***Integration into Core Framework***
- Profiling and Results
- Conclusion

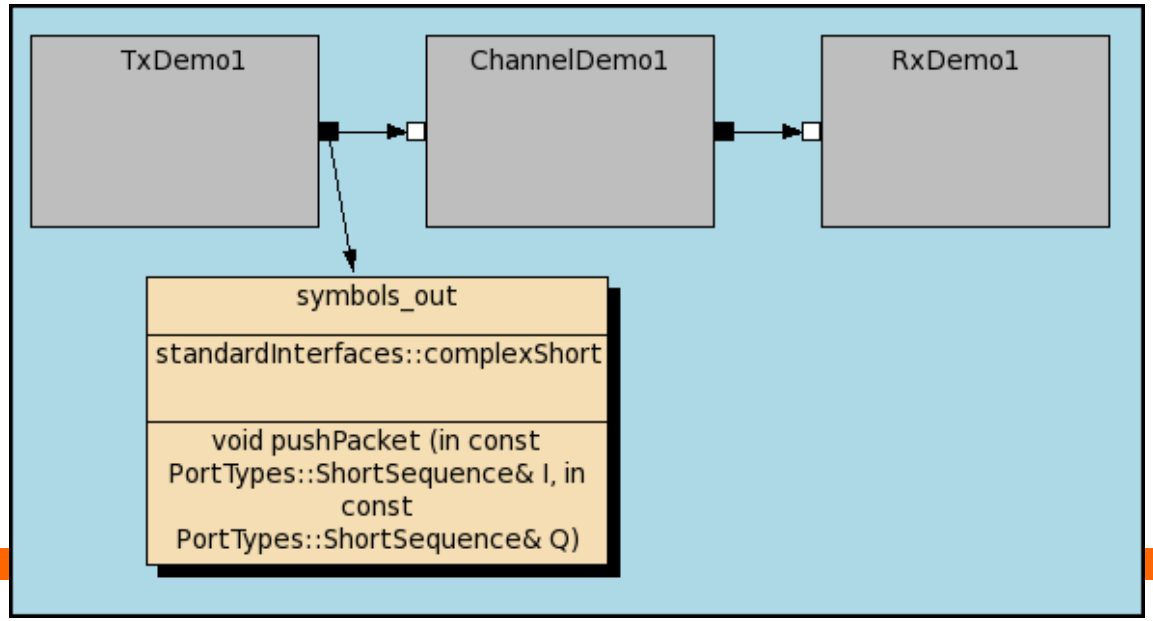
Integration into OSSIE

- Two Interfaces within component
  - Interface to Core Framework
  - Interface to FPGA



OSSIE Demonstration Waveform

- Waveform is ml403_ossie_demo, similar to ossie_demo waveform
- Components
  - TxDemo, ChannelDemo, RxDemo

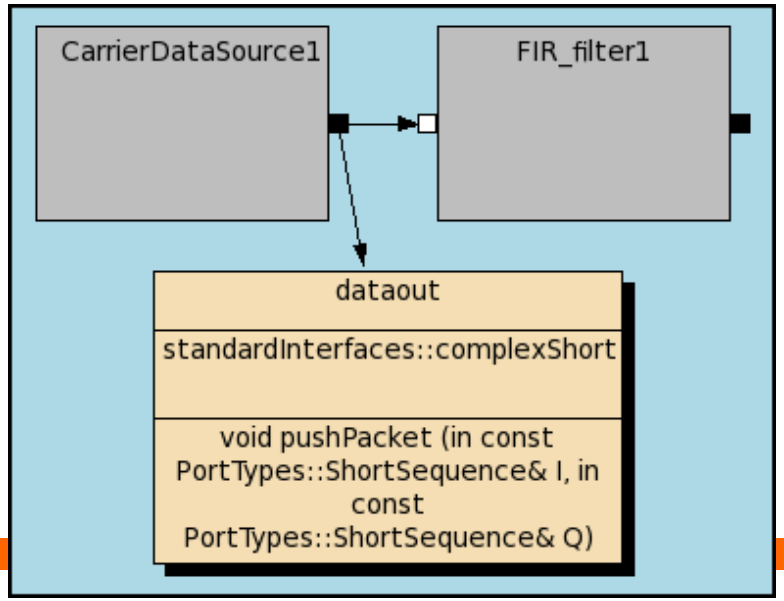


OSSIE Demonstration Waveform (con't)

- Symbol generation done on FPGA
- PowerPC components for ChannelDemo, RxDemo
- Good place to start if reviewing work or reproducing results

FIR Filter Waveform

- FIRFilterDemo waveform contains two components
  - CarrierDataSource
  - FIR_filter



FIR Filter Waveform (con’t)

- CarrierDataSource generates cosine samples
- FIR filter circuit is abstracted within FIR_filter

FIR Filter Component

- Cosine samples received by FIR_filter component
- Samples sent over bus to FPGA
- FIR Filter processes samples, stores in output FIFO
- Input samples written to FPGA one by one

Outline

- Introduction
- FPGA Interface
- Integration into Core Framework
- ***Profiling and Results***
- Conclusion

Method of Profiling

- Measure “wall clock” time
  - Calculate effective rate
- Measure processor usage
- Three Measurements
  - Interfacing with FPGA
  - Conversion to CORBA Type
  - Overall waveform measurement

FIRFilterDemo Waveform Results

- FIR_filter: Interfacing with FPGA
  - 10 Million iterations of 32 bit transfer
  - Operating Frequency: 10.1 MHz
  - Data Rate of 324 Mbps
  - Processor Utilization: 98.91%
  - System Utilization: 0.30%

FIRFilterDemo Waveform Results

- FIR_filter: CORBA type conversion
  - Interface with FPGA, convert to complexShort type
    - 16 x 2 bits, OSSIE Data Type
  - 60,000 complexShort packets, each of length 512
  - Operating Frequency: 8.5 MHz
  - Data Rate of 272 Mbps
  - Processor Utilization: 99.16%
  - System Utilization: 0.19%

FIRFilterDemo Waveform Results

- FIR_filter: Full waveform profiling
  - CarrierDataSource, FIR_filter components
  - 3000 complexShort packets, each of length 512
  - Operating Frequency: 117 kHz
  - Data Rate of 2.34 Mbps
  - Processor Utilization: 25%
  - System Utilization: 31%

Removal of Core Framework

- Removal of framework
  - Single ‘component’ used
  - Carrier samples read out of memory, not transferred through CORBA
  - Essentially same profiling method as interfacing with FPGA

Results with no Core Framework

- FIR Filter
  - Operating Frequency: 10.407 MHz
  - Very close to earlier result of 10.134 MHz

Table of Results

FIR_filter interfacing with FPGA directly

Frequency (MHz)	Processor Utilization	System Utilization
10.134	98.91%	0.30%

FIR_filter component interfacing with FPGA and conversion to CORBA

Frequency (MHz)	Processor Utilization	System Utilization
8.456	99.16%	0.19%

FIRFilterDemo Waveform

Frequency (MHz)	Processor Utilization	System Utilization
0.116	25.26%	30.59%

Outline

- Introduction
- FPGA Interface
- Integration into Core Framework
- Profiling and Results
- ***Conclusion***

Conclusion

- FPGAs allow for increase in throughput
- Integration must be done correctly to obtain the benefits
  - SCA does not define FPGA interface, use this as an advantage
  - Minimize overhead when possible
  - Abstract operation of FPGA into a software component

Conclusion (con’t)

- Alternate Methods other than Polling
  - Need deterministic data transfer
  - Schedule or prioritize
- All (open) source code can be found at:
  - http://ossie.wireless.vt.edu/

Questions?