Using Digital Signal Processors in GNU Radio

Justin Ford
jrford@sandia.gov
505-284-5567

Key Team Members (SNL): Adam Goldhammer, Vince Hietala, and Kevin Malone

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94AL85000.

© 2013 Sandia Corporation
Further dissemination only as authorized; requests shall be approved by the originating facility or higher DOE programmatic authority.
Low-Power SDR: DSPs in GR

TI OMAP
(e.g. Gumstix in E100)

Easier Development*

TI 66AK2Hx DSP+ARM SoCs

Qualcomm Snapdragon 800

Ultimate Application
DSPs as Coprocessors in GNU Radio

- A coprocessor (DSP in this case, but similar to GRGPU*) can be inserted in a GNU Radio flowgraph by:
  1. Breaking the graph
  2. Inserting blocks to transfer data to the DSP and from the DSP
  3. Implementing the required functions in the DSP
  4. Reconnecting the graph

* Interactions with Will Plishker at UMD shaped the approach SNL has taken to using DSPs in GNU Radio
DSP functions are performed between an egress and ingress.

Functions can be placed on host processor before and/or after DSP blocks.

DSP blocks are parameterized just like their host-side counterparts.

References to data chunks on DSP are passed as integers between DSP blocks (thus the data type change at egress/ingress).
Offloading Processing to a DSP

Texas Instruments’ TMS320C6670 System-on-Chip

- Four DSP cores at 1.2GHz
  - 153.8 GMACS / 76.8 GFLOPS total (ideal)
  - Designed for SDR, radar, and broadband applications
  - Evaluation board supports PCIe interface
TMS320C6670 SoC Details

- PCIe 2.0 Interface @ 2 Lanes
  - 85% efficient \(\Rightarrow\) 850 MB/s theoretical
- IEEE-compliant single-precision floating point arithmetic
- Instruction set and execution unit optimized for digitally processing signals
- Fast Fourier Transform Coprocessor
  - 2048-point FFT (fixed-point) in 4.8us
- Viterbi and Turbo Decoder Coprocessors
- Future TI DSP SOCs will have multi-core A15 ARM GPPs and multi-core C6600-class DSPs
DSP Offloading Details

- Establish a DSP Process Queue (cmd, input, output) in DSP memory space
- Host driver writes command structure and input data to process queue
- Host driver sends “interrupt” to DSP
- DSP Master core searches queue for new job and assigns job to available DSP core via inter-processor communication
- Resultant data is placed in process queue and host is notified via interrupt
Neither GPP or DSP are “occupied” by data transfers (DMA is used)

The job queue allows the host to request new operations while the DSP is busy allowing for continuous utilization of the DSP
Benchmarking Performance
(Linux user space)

- GNU Radio 3.7 on Ubuntu 12.04 64-bit (workstation-class machine)
- Timing done using Linux system calls (~ μs accuracy)
- All operations are on complex floating point precision data
- Benchmarks are run on 1 million samples, results are averaged
- Data was collected using single DSP core
Ingress (read)

Egress (write)

**DMA over 2-lane PCIe v2.0**

- DSP PCIe peripheral 10 GT/s or 1 GB/s max
  - TI specifies slower read rates compared to write rates
  - Slowdown with increasing host threads is mostly benchmarking artifact

Egress (write):
- ~750 MB/s, \( \sigma < 30 \, \mu s \)

Ingress (read):
- ~600 MB/s, \( \sigma < 100 \, \mu s \)
Overall Throughput

- Tradeoffs are dependent on function type
  - Mostly computational intensity
  - Useful information for selecting transfer chunk size in GNU Radio

- ~180 MB/s, σ<500 µs

- ~110 MB/s, σ<500 µs
Flowgraph using DSP

- Flowgraph to check data validity
  - Multiple operations chained together
  - Input and output data difference of zero indicates no corruption
- Admittedly not a useful operation in practice, just illustrating functionality

Screenshots from GNU Radio companion, a part of GNU Radio licensed under GPL
Aside: liquid in GR

- Hackfest in June 2013 at Virginia Tech
- Exploration of using Joe Gaeddert’s C-based DSP routines
  - Available FEC and other modules might be of use
  - Apparent fixed-point support would be very attractive
- Decided to try simple cases to test integration
- Out of tree module at github.com/minimalwatts/gr-liquiddsp

Screenshots from GNU Radio companion, a part of GNU Radio licensed under GPL
Final Thoughts: Coprocessors in GR

• Digital signal processors offer unique features as a coprocessor for software-defined radio:
  ▪ Well-suited to the task: exemplary floating point performance and instruction set/execution unit tailored to multiply/accumulate operations
  ▪ Very power efficient with low-power variants available (applicable to embedded systems)
• There are tradeoffs between DSPs, FPGAs, and GPUs for accelerating SDR functions
  ▪ The ultimate best choice is always going to be application-dependent
  ▪ A general framework for using coprocessors in GNU Radio is desirable