AMIR YAGHOUBI'S 2008 SUMMER PROJECT

A FPGA Based Digital Data Acquisition System for MKIDs Experiment
Contents

- MKID Fundamentals
  - The Problem
- What are FPGAs?
  - Comparison to ASIC
  - What is an iBOB?
  - BEE2
  - The Future
- Digital Data Acquisition System
  - Caltech Suggested FPGA Implementations
- Progress
  - Current Design
- Conclusion
MKIDs Fundamentals

- **MKIDs** stand for Microwave **Kinetic Inductance Detector**
- The kinetic inductance of a superconductor is the inductive surface impedance of the superconducting Cooper pair condensate at microwave frequencies due to its inertia when an electric field is applied: $L_{\text{kin}}$ [1]
- The ohmic surface resistance is due to broken Cooper pairs: $R_{\text{kin}}$

MKIDs Fundamentals

- In order to calculate the input energy we are interested in measuring the density of quasiparticle and change when Cooper pairs are broken.
- To monitor these quantities the superconductor is placed in a resonant circuit.
- $F_0$ is the resonance frequency.
- $Q$ is the quality factor of the resonant circuit.
- The sensitive bandwidth of the MKID is therefore $F_0/2Q$. 
General Overview of MKID Operation

Figure courtesy of P.K. Day[1]

The Problem

- When an event happens i.e. we have an input energy, the circuit transmission changes due to changes in $F_0$ and $Q$
- In order to measure these changes we monitor the amplitude and phase of a drive signal at the $F_0$
- In case of multiple resonators, each would have a different $F_0$, separated by an integer multiple of resonator bandwidth
- A frequency comb of all the $F_0$ of the resonators excites the resonators
- At the receiving end on the DSP board we demodulate the comb at each comb frequency to find out the response of each resonator (i.e. phase and amplitude changes of each frequency component)
Block Diagram of the Caltech Proposed Readout System

Note: This is not a feedback system. The signal processing stage is done independently of signal generation on the FPGA board.

[1] Phil Maloney, Jack Sayers, Jonas Zmuidinas, & Ben Mazin; MKIDCam Readout Electronics Specifications
What Are FPGAs?

- FPGA stands for Field-Programmable Gate Array.
- It contains programmable logic gates and interconnects.
- FPGAs are basically large-scale configurable logic devices. Their logic gates can be programmed to perform functions of basic logic circuits such as NAND, XOR or combined together to perform much more advance functions.
- It allows designing a reprogrammable fully operational application-specific circuit board, in our case a fully functional multi-purpose DSP board.
- Its most important applications are digital signal processing, computer hardware emulation and sound processing.
Comparison to ASIC

- FPGA’s could theoretically provide up to 10 times more computing throughput than a DSP-based system with similar power consumption and cost or over 100 times of today’s microprocessors [1].

- In addition to potentially unpatrolled computing power, FPGA’s have the following advantages over ASICs:
  - They are reprogrammable
  - They have a flexible interconnect architecture
  - They pose a fully modular computing architecture
  - The non-recurring engineering costs are minimal for FPGA’s

What is an iBOB?

- **iBOB** stands for **Internet Break-Out Board**
- It’s is an FPGA-based DSP board designed and assembled by CASPER group
- The original motivation behind designing iBOB has been to have an accessory processing board that is capable of digitizing analog data, and doing relatively small scale signal processing tasks before a more advanced board
iBOB’s Brief Specification

- 1x Xilinx Virtex-II Pro XC2VP50-7FF1152 FPGA (Rated CLK 250Mhz) [1]
- An IBM PowerPC 405 core is integrated into the Virtex-II Pro device using the IP-Immersion architecture
- 72x512K SRAM for a total of 36Mbit of onboard memory

[1] Bob Brodersen, Chen Chang, John Wawrzynek Dan Werthimer, Melvyn Wright; BEE2: a multi-purpose computing platform for radio telescope digital signal processing applications; EECS, UC Berkeley
An Actual iBOB

Photo courtesy of UC Berkeley SETI Group
The iBOB’s ADC converts analog inputs to digital outputs. Every clock cycle, the inputs are sampled and digitized to 8 bit binary point numbers in the range of [-1, 1) and are then output by the ADC.

- It has two digitization modes:
  - Single stream mode: 8x of iBOB’s max clock frequency equal (2 Gsps)
  - Dual stream mode: 4x of iBOB’s max clock frequency (1Gsps)

- Specifications
  - 1 x Atmel/E2V AT84AD001B 8-bit Dual 1Gsps ADC
  - 1 x Z-DOK+ 40 differential pair connector
  - 3-wire serial
iBOB’s DAC

- The DAC board converts 4 digital inputs to 1 analog output. The DAC runs at 4x FPGA clock frequency (4\times250\text{Mhz} = 1\text{Gsps}), outputting analog converted samples 0 through 3 each FPGA clock cycle.
- These digital streams have to be 9bits, resulting in a dynamic range of [-256, 256).
Like all the FGPA based board, the preferred programming language of iBOB is a **HDL** based language.

HDL stands for **Hardware Description Language**.

Required software for programming iBOB:

- Matlab v7.0.4 (R14) SP2
- Simulink v6.2 (R14) SP2
- Xilinx System Generator v7.1
- Xilinx EDK v7.1.02i
- Xilinx ISE v7.1.04i
- MSSGE libraries
Programming iBOB

- The **BWRC** (Berkeley Wireless Research Center) has developed **MSSGE** or **bee_xps** tool flow.
- It’s a high-level design tool for the iBOB as well as BEE2 platform.
- The MSSGE runs in Matlab/Simulink environment. The MSSGE toolflow stitches together several design and implementation environments. In other words it’s an expansion of libraries supplied by Xilinx.
- For our design, we used both CASPER libraries and Xilinx libraries.
- The iBOB’s PowerPC runs a stripped down Linux kernel called **TinySH**.
The example program utilizes both Xilinx and CASPER blocksets.
BEE2

- In addition to iBOB, CASPER has designed a much more powerful board called BEE2.
- It stands for Berkeley Emulation Engine 2.
- The BEE2 is intended as an open source general-purpose high performance reconfigurable computer.
- It has five Virtex-2 Pro FPGA cores arranged in a star topology.
  - Each individual FPGA has four independent channels to DDR2 DIMM. For a total of twenty 400Mhz DDR2 memory slots allowing for memory expansion up to 20Gb.
  - Every FPGA core has two 10Gbe Ethernet ports for I/O purposes.
A Running BEE2 Board

Photo courtesy of UC Berkeley SETI Group
Direct Successor to iBOB is a board called ROACH. It’s being designed by CASPER hardware designing team and it’s scheduled to be available by the end of 2008.

ROACH stands for Reconfigurable Open Architecture Computing Hardware.
Roach’s Advantages Over Older Boards

- It merges aspects from the IBOB and BEE2 platforms into a single board while sporting an all new powerful Virtex-5 FPGA:
  - It is based on the new Xilinx Virtex-5 FPGA (BEE2 or iBOB are based on older Virtex-II architecture)
  - Two DDR2 memory slots for future memory expansion for a total onboard memory capacity of 4GB (Similar to BEE2)
  - A single FPGA board (iBOB)
  - Two interfaces that allow the use of the current ADC boards 2 Gs/sec in dual mode, or a new 3 Gs/sec or dual-board interleaved @ 6 Gs/sec ADC
ROACH vs. BEE2

- ROACH board has a single Xilinx Virtex-5 which contains 94,000 logic cells and 640 multiplier/accumulators for a total of 400 Gops/sec of processing power.

- BEE2’s five Xilinx Virtex-II Pro FPGAs, each contains 328 multipliers and has 74,000 configurable logic cells for a total of 500 Gops/sec of processing power for the whole board.
## Price and Power Consumption of CASPER Hardware

<table>
<thead>
<tr>
<th>Board</th>
<th>Board Cost</th>
<th>Cost with FPGAs</th>
<th>Gops per Sec</th>
<th>Power (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBOB</td>
<td>$400</td>
<td>$2700</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>BEE2</td>
<td>$5000</td>
<td>$23500</td>
<td>500</td>
<td>150</td>
</tr>
<tr>
<td>ROACH*</td>
<td>$1000</td>
<td>$3200</td>
<td>400</td>
<td>50</td>
</tr>
<tr>
<td>ADC (1 Gs/s)</td>
<td>$200</td>
<td>$200</td>
<td>N/A</td>
<td>2</td>
</tr>
<tr>
<td>ADC (3 Gs/s)*</td>
<td>$1000</td>
<td>$1000</td>
<td>N/A</td>
<td>5</td>
</tr>
</tbody>
</table>

*Note: Estimated from prototype versions currently in the development

(All the estimations were done by the designing team at CASPER)
Caltech Suggested Digital Read-out FPGA Implementations

1) Fine channelization using FFTs:

- Cascading two smaller FFTs, and using a “corner turn” or matrix transpose operation in between these two FFTs. The second FFT operates on the output channels of the first FFT, and has to be time-multiplexed to save FPGA resources.

   - **Pros:**
     - Fine channelization is performed at the desired frequency resolution (for instance a 512K point FFT would give us 112K channels in excess of the desired 400 MHz/1kHz = 400,000 channels).
     - It doesn’t need modification in design when resonant frequencies of the MKIDs change.

   - **Cons:**
     - This requires a substantial amount of memory for storage of intermediate results when doing matrix transpose (Unattainable on an iBOB due to memory and processing power limitations).
     - Large number of redundant channels makes it necessary to have a final channel selection stage.
Block Diagram

Photo courtesy of CalTech[1]

FFT Channelizer

[1] Phil Maloney, Jack Sayers, Jonas Zmuidinas, & Ben Mazin; MKIDCam Readout Electronics Specifications
2) Parallel digital down conversion using pre-computed LO

- Each readout channel consists of a numerical LO waveform which is computed ahead of time and stored separately in memory on the FPGA and a complex mixer, followed by a FIR decimator and a complex low-pass filter.

**Pros:**

- It requires a very moderate amount of memory for storage of intermediate results.
- It allows for the waveform for a single resonator to be individually adjusted while keeping the other waveforms fixed.

**Cons:**

- FIR multiplication taps are intensely resource consuming on the iBOB. This limits the desired precision of the filters.
- The onboard DDS generator with higher precision is resource consuming.
- It would require major modification in parameters when resonant frequencies change.
**Block Diagram**

Digital downconverter (DDC)

Photo courtesy of CalTech[1]

[1] Phil Maloney, Jack Sayers, Jonas Zmuidinas, & Ben Mazin; MKIDCam Readout Electronics Specifications
Our design of frequency filter bank utilizes both DDC and FFT channelization methods:

- It’s based on a conventional M-channel channelizer, in other words a bank of M fixed frequency band-pass filters.
- The channels get mixed with a LO of the user configure frequency.
- The digitally mixed signals then get coarse low-pass filtered while at the same time they get decimated down to the critical sampling rate (2x of the LPF bandwidth).
- At the fourth stage each channel goes through a fine FIR-low pass filter.
Input Signal

Amplitude (Scaled from 1) vs. Time (µS)

Note: The first wave is in-phase wave and the second is quadrature wave of the input signal.
The digitized I & Q signals from the ADC enter iBOB

Each signal has been time interleave into 4 channels

Each sampled is a 8-bit word

Every I channel gets concatenated with its corresponding Q channel to form a single FPGA complex number of length 16-bit in the form of:

\[ Z = I + jQ \]
Block Diagram of Data Aligner

Inputs are at signed 8-bit real and imaginary numbers and outputs are single 16-bit complex numbers.
Mixers

- Every single digitized channel is inputted to 4 separate digital mixers
- The digital mixer would digitally mix the input signal with an LO whose frequency has been specified by the user
- The mixing frequency has to be an integer fraction of the native FPGA clock (in this design 200/x MHz)
- Each of these mixer’s frequency has been configured to down-mix a specific frequency component of the frequency-comb signal. (In this design they are \((n\pi) \times (200/8\pi)\) where \(n = 1, 2, 3, 4\)
Time Sample Decimation

- Since the bandwidth of the low-pass filter is 2Mhz, we can decimate down our digitally mixed signal to critical sampling (2x of Nyquist frequency = 4 MHz)
- The each of FIR-based decimation filters in this design handles 4 complex (4 real and 4 imaginary) streams and decimates them down to 1 time sample
- This would save a lot of computing resources and reduce the total number of data lanes for each channel on the iBOB prior to the fine low-pass filtration stage
Each digital mixer (channel_1) would digitally mix the input signal with an LO and a decimation FIR filter (dec_fir) would decimate down the resulting wave.
First Filter’s Magnitude Response

Magnitude (dB) vs. Frequency (MHz)

Note: This is the magnitude response for the first stage
Fine Low-Pass Filter

- The low-pass filters are 32 tap Xilinx-based FIR-based filters
- Their design method is “Hamming Window” with a filter order of 31
- Their magnitude response have been designed through the FDATool (Filter Design and Analysis Tool)
- FDATool has a GUI user interface for designing and analyzing filters. It provides tools for analyzing filters, such as magnitude, phase response and pole-zero plots
- FIR filter’s coefficients and their structure are inherited from FDATool
- The output of each filter is $A(t)\cos\Phi$ where $A(t)$ and $\cos\Phi$ are the phase and amplitude for the given signal
Second Filter’s Magnitude Response

Magnitude (dB) vs. Frequency (MHz)
Final Output of the Cascaded Filters

Phase Difference (Normalized to 1) vs. Time (µS)
FFT Channelization

- In order to extract the amplitude of each channel, the input signal goes through a N-point (in this design 32 points) FFT.
- The stream of computed FFT then goes through a time division multiplexer of the same size to extract each bin on a single channel with a sampling rate of fs/N (in the current design fs/32).
- If frequency components of the comb signal are integer multiples of the fundamental frequency of FFT, then each channel would give the amplitude of the representing frequency.
FFT blocks (white blocks) receiving digitized input signal from ADC and outputting to down-sampler
Output of FFTs

Frequency Bin Amplitude (Normalized to 1) vs. Time (µS)

Note: For the first diagram, each interval between the two samples is the size of FFT.
Memory and Ethernet Output

- These four wave data are recorded on four $2^{11}$ @ 32-bits distributed memory blocks.
- For every channel, its amplitude $A(t)$ and the product of amplitude and cosine of its phase $A(t)\cos\phi$ are concatenated to a 64-bit word.
- Every wave amp and phase is concatenated and sent to 10Gbe block to be wrapped in a UDP frame for transmission.
- The packet size should be at minimum 4x64-bits.
Design Notes

- Xilinx supplied FDATool greatly facilitated the design of band-pass filters
- FDATool via its user friendly GUI interface provided easy means to calculate FIR coefficients and various filtering methods such as Windowing or Equiripple
- CASPER block provided an extensive set of tools for our read out system
- The only drawback to CASPER blocks was the required substantial optimization of them in order to properly utilize them in the design
Conclusion

- Starting from mid-May till mid-August the project progressed very well from a very basic design to a much more elaborate and complex design.
- The current design has been devised and programmed to have solid foundations for future expansion and addition of new features.
- Many aspects of the design were hardware tested at the later stages of the project due to unavailability of iBOB for the most part of summer however the majority of these stages of the design ran almost flawlessly on the board in comparison to simulation.
- More hardware testing of the design will become necessary as features become more close to the actual specifications.
- iBOB exhibits unprecedented sign of promise for future MKID data acquisition systems, even surpassing earlier expectations due to its inherit flexibility and a staunch and unwavering support from CASPER.
- The promised fully backward compatibility between ROACH and iBOB would ensure that any investment on iBOB will be an investment on future systems; more specifically on ROACH.