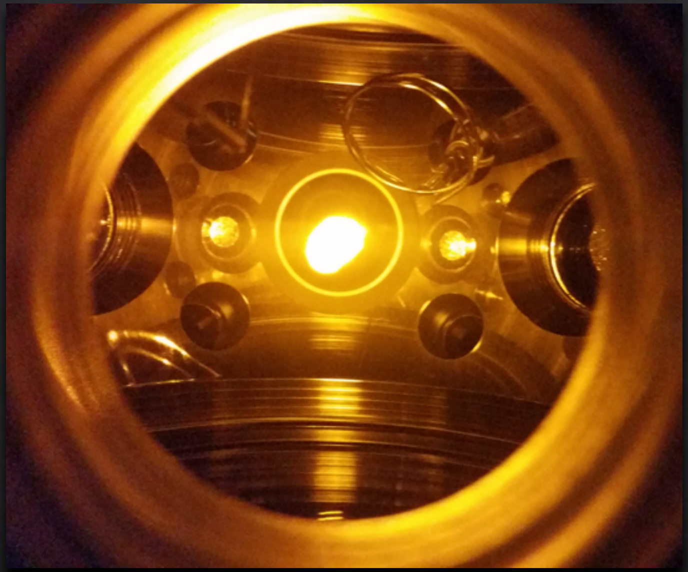
Stabilizing laser and microwave fields for coherent control of spin-exchange collisions in ultracold sodium gases



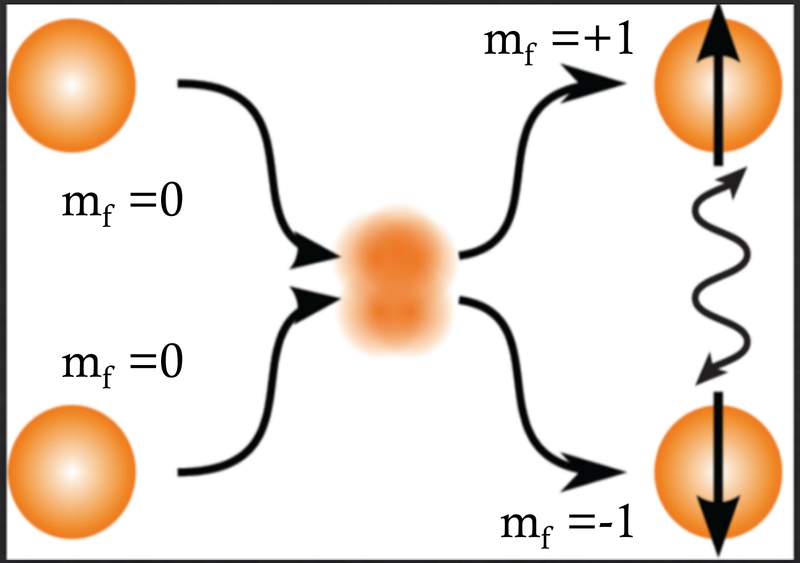
Jeremy Norris

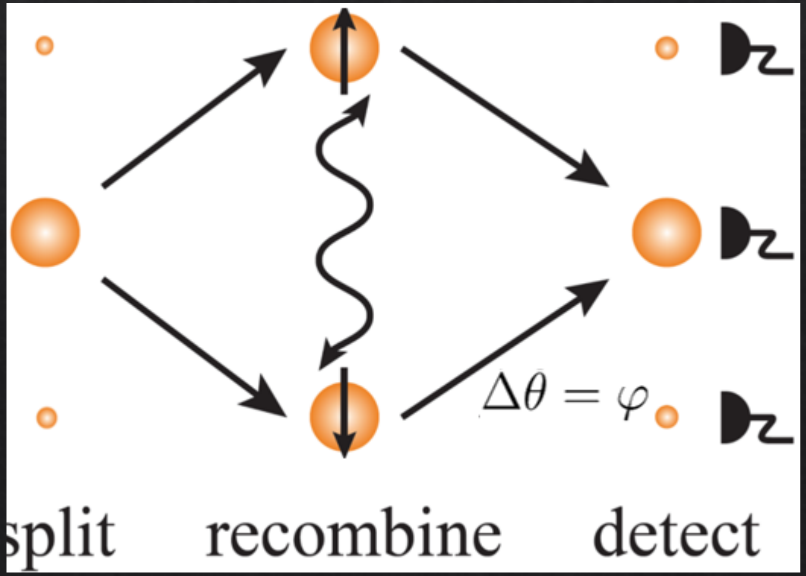
Graduate Student: Shan Zhong

Advisor: Dr. Arne Schwettmann

Spin-exchange Collisions

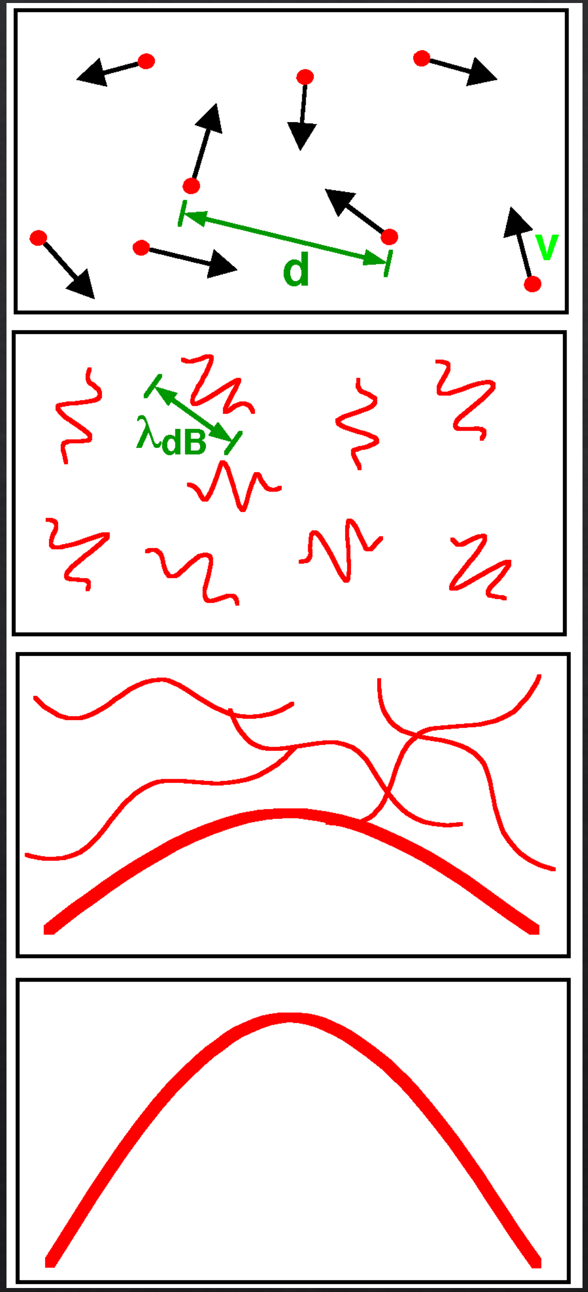
- Colliding atoms can change their spin to form entangled pairs
- Useful to build atom interferometer
- Collisions are random and uncontrollable at room temperature
- We can gain control by lowering the temperature of our sodium gas
- Coherent control is achieved by forming a Bose-Einstein condensate (BEC)





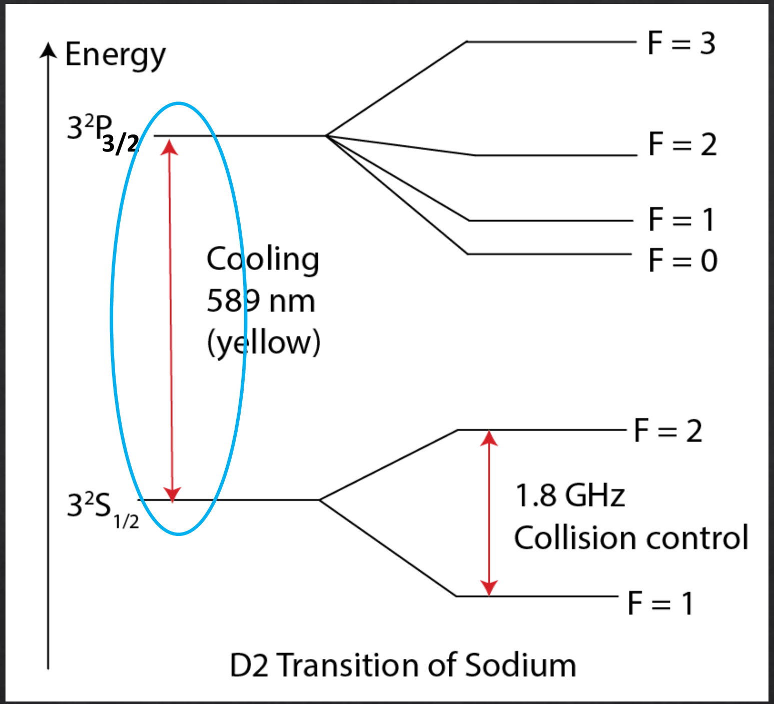
Bose-Einstein Condensate

- At room temperature atoms behave classically
- At lower temperatures wave packets overlap
- Below critical temperature of ~ 100 nK, BEC forms
- Giant matter-wave; macroscopic quantum mechanical object
- Here collisions happen controllably and coherently



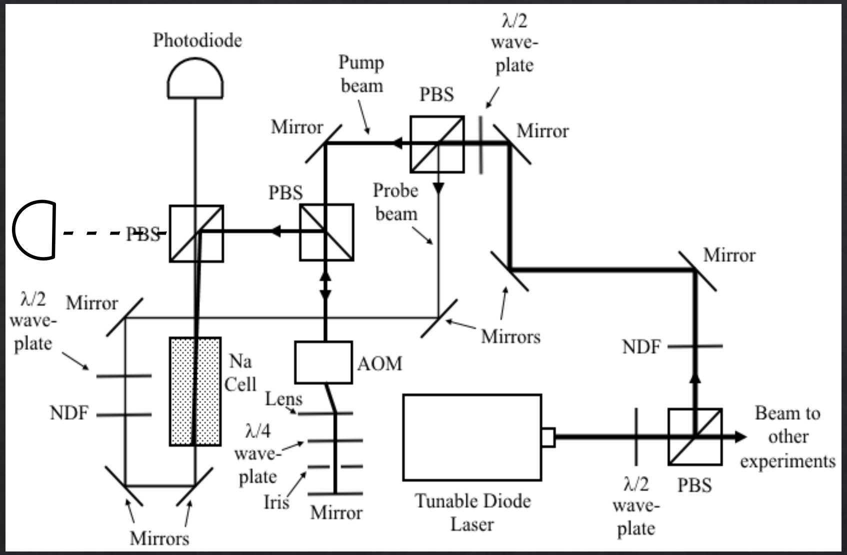
Laser Cooling of Sodium

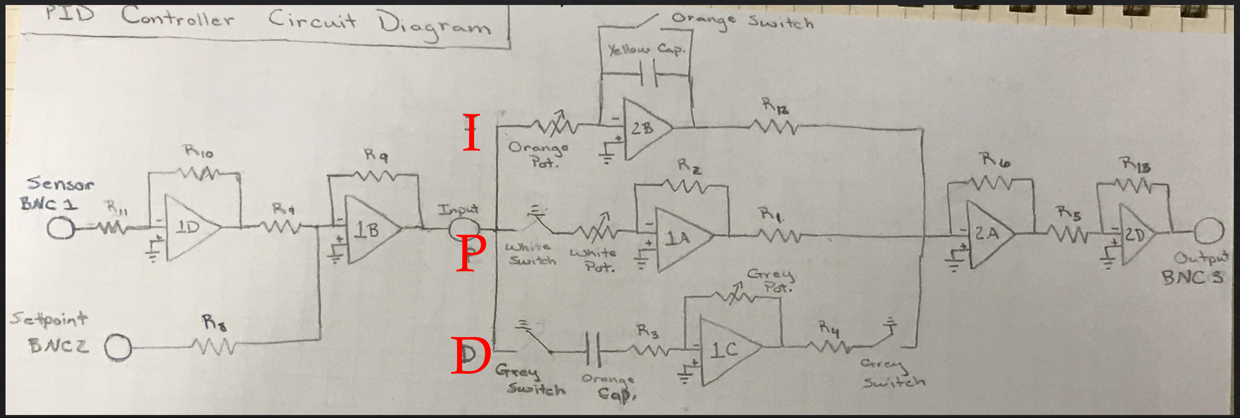
- Yellow laser at 589 nm is used to cool sodium gas
- Laser must be stabilized
- FM saturated absorption spectroscopy for stabilizing laser frequency
- Also need amplitude stabilization



Stabilization of Laser Amplitude

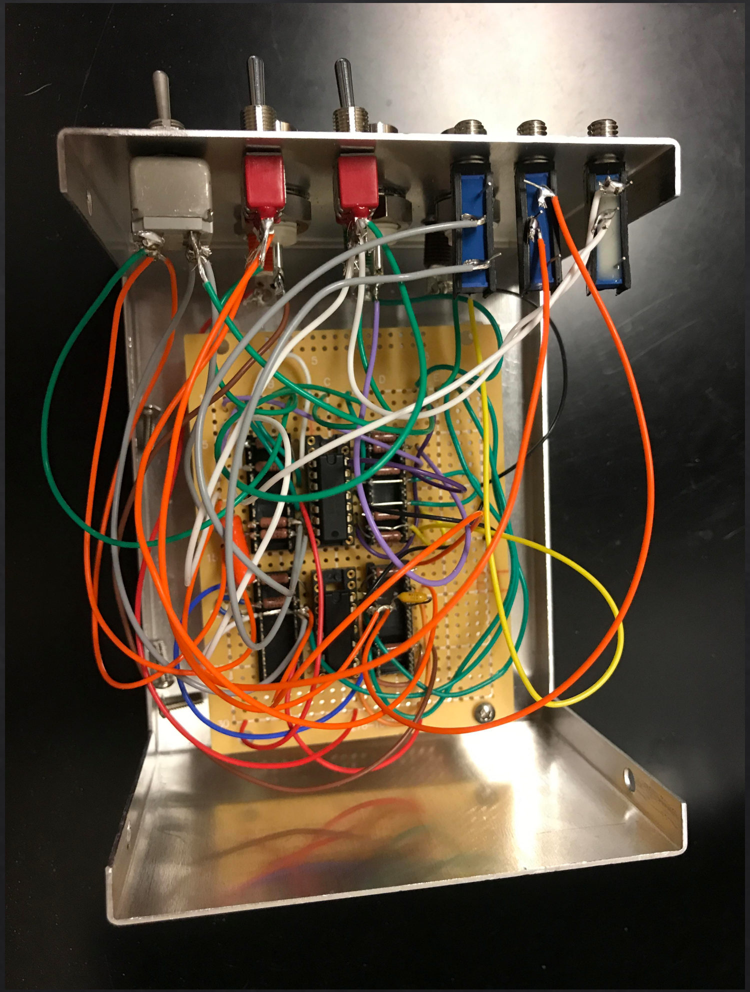
- Amplitude is measured with photodiode, adjusted using Acousto-Optic Modulator (AOM)
- Build a PID controller which is feedback loop
- Looks at measured amplitude and adjusts to reach set point
- Works like cruise control in a car
- PID: Proportional, Integral, Derivative
- Replicated existing PID controller used to stabilize laser frequency





PID Controller

Front

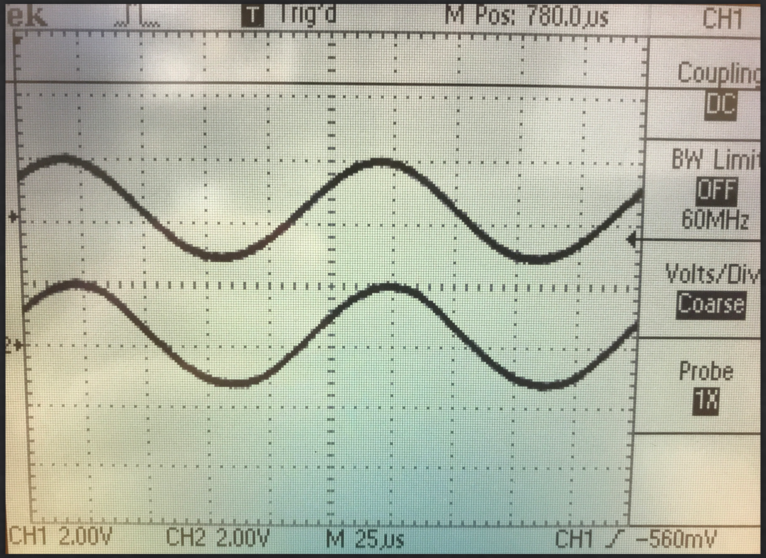


Top

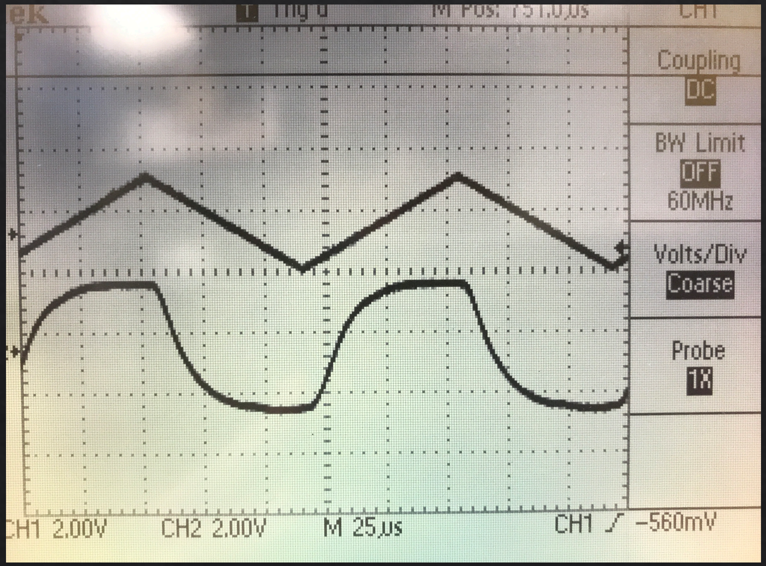
Testing PID Controller

- Artifact due to slow speed of analog chip
- Speed is ~ 100 kHz, we need response of 30 kHz (fast enough)
- Too disruptive to add to existing setup
- Laser working well enough without stabilization
- Can be added during down time

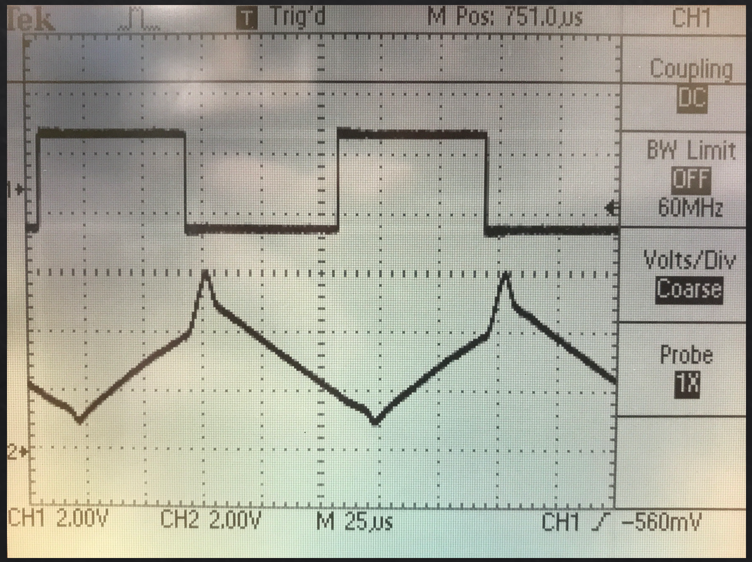
- Proportional



- Derivative

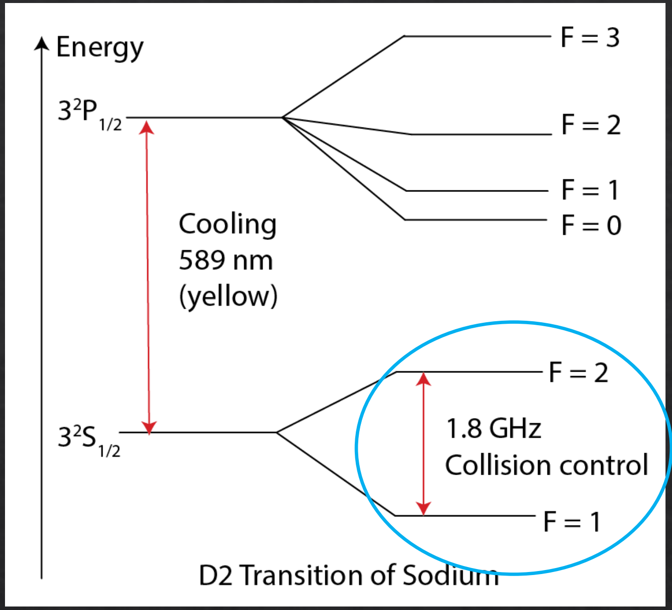


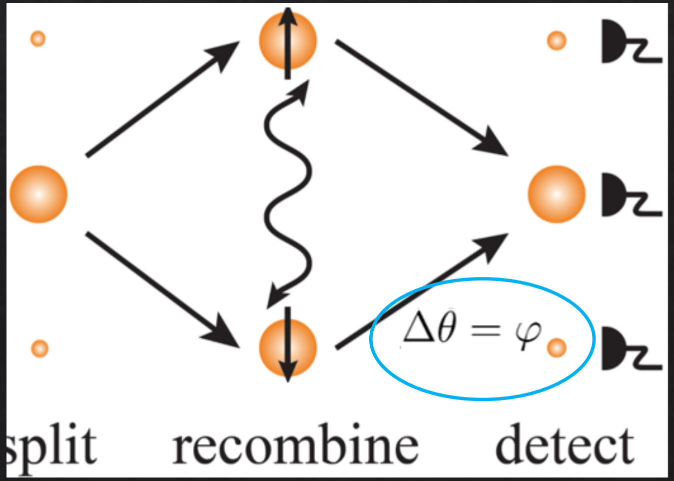
- Integral



Microwave Control of Collisions

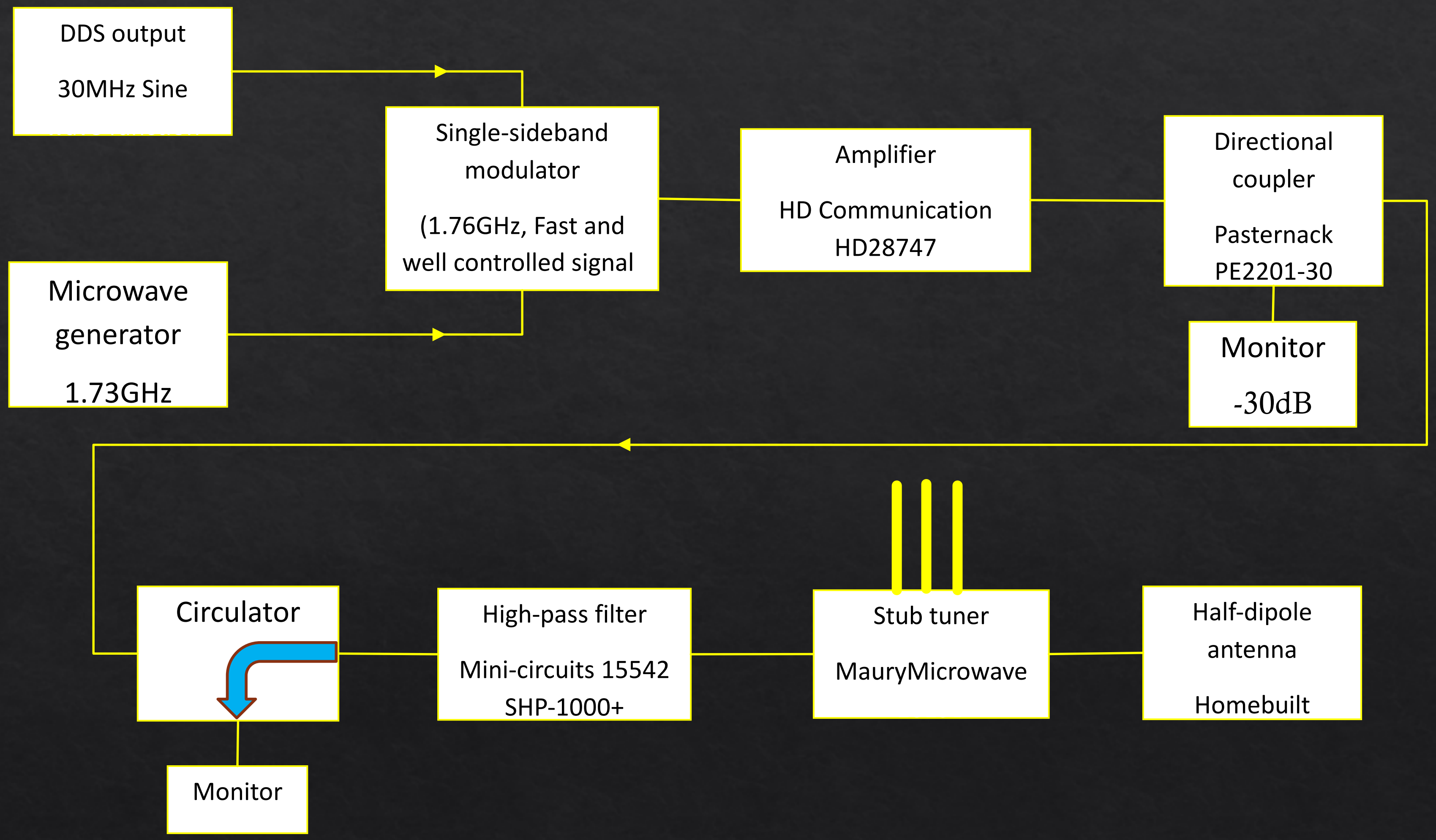
- 1.8 GHz microwaves must have fast amplitude, frequency, and phase control
- Solution is to use field programmable gate array (FPGA) that controls direct digital synthesizer (DDS)
- FPGA
  - Huge array of thousands of programmable logic gates
  - 40 digital outputs run in parallel at high speeds
  - Faster than microcontroller
- DDS
  - Dedicated chip that produces sine wave outputs with controlled amplitude, frequency, and phase

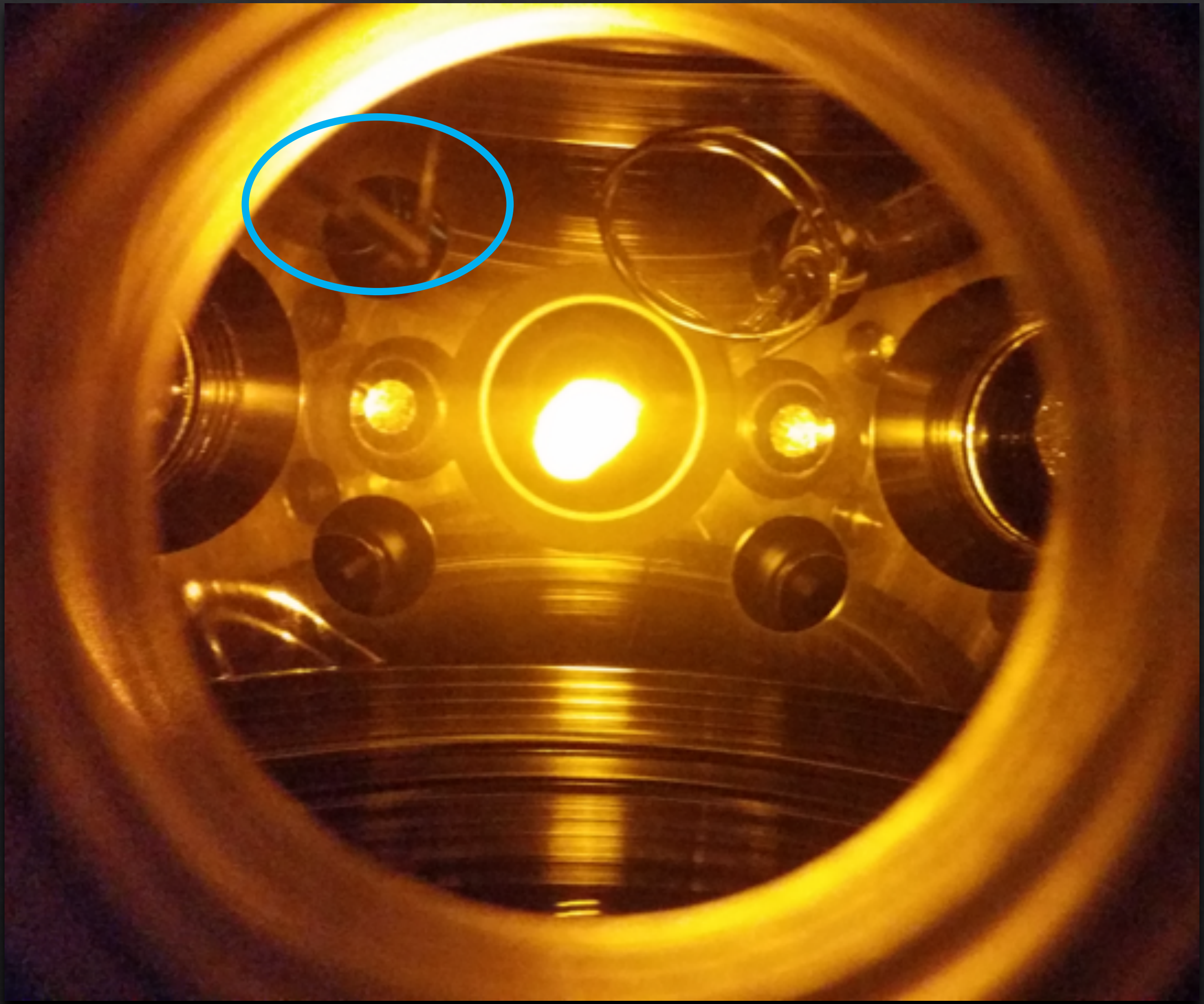




Microwave Setup

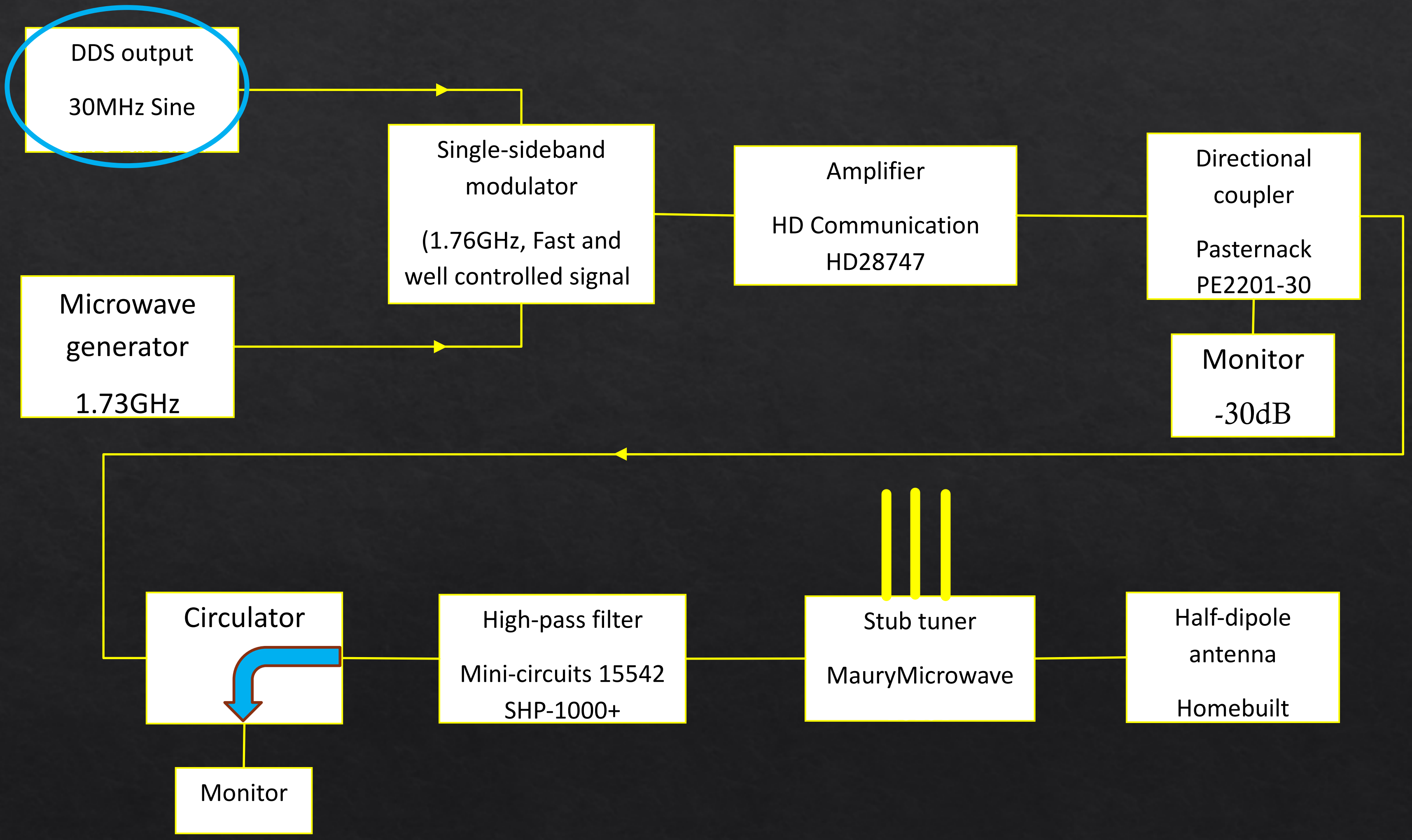
Well controlled slow signal





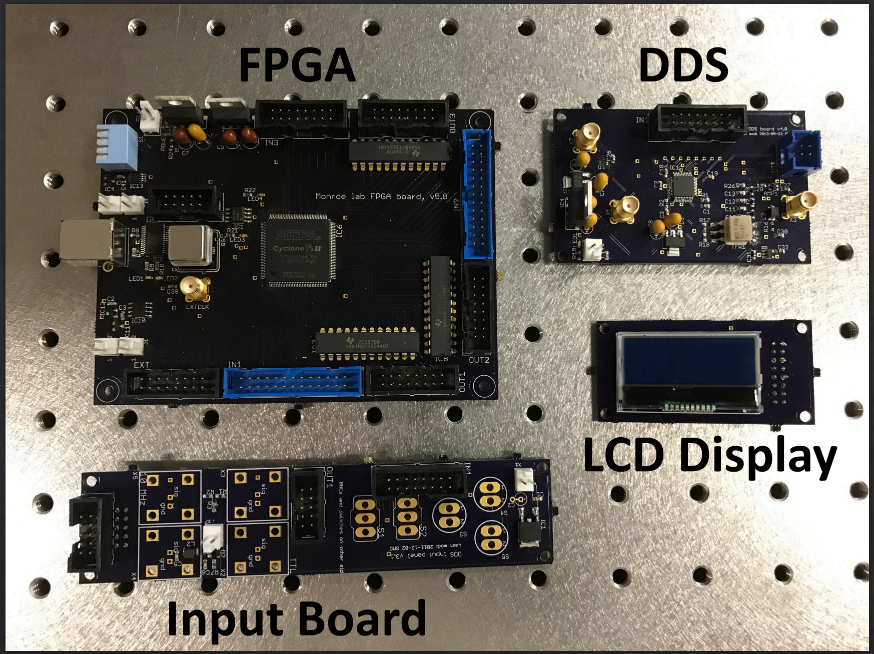
Microwave Setup

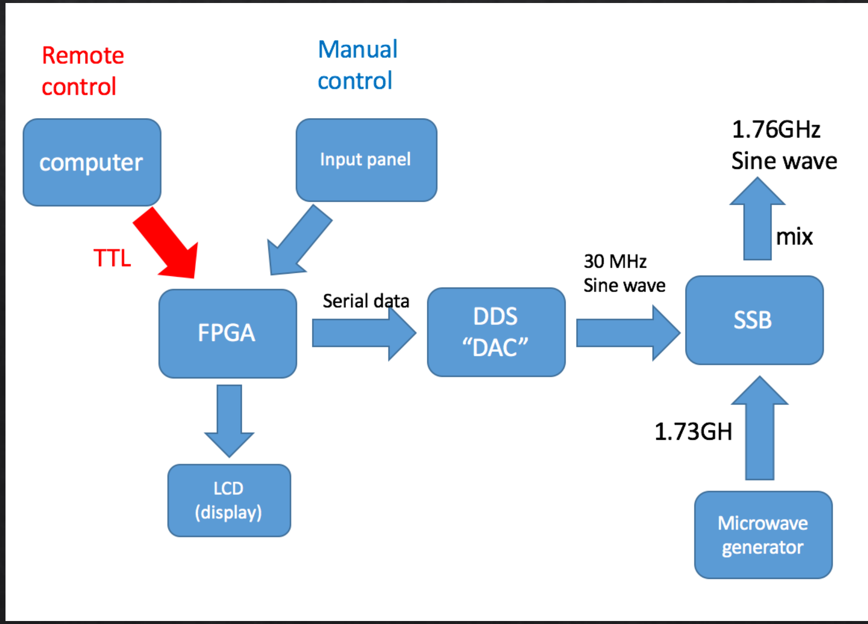
Well controlled slow signal



Where I Started

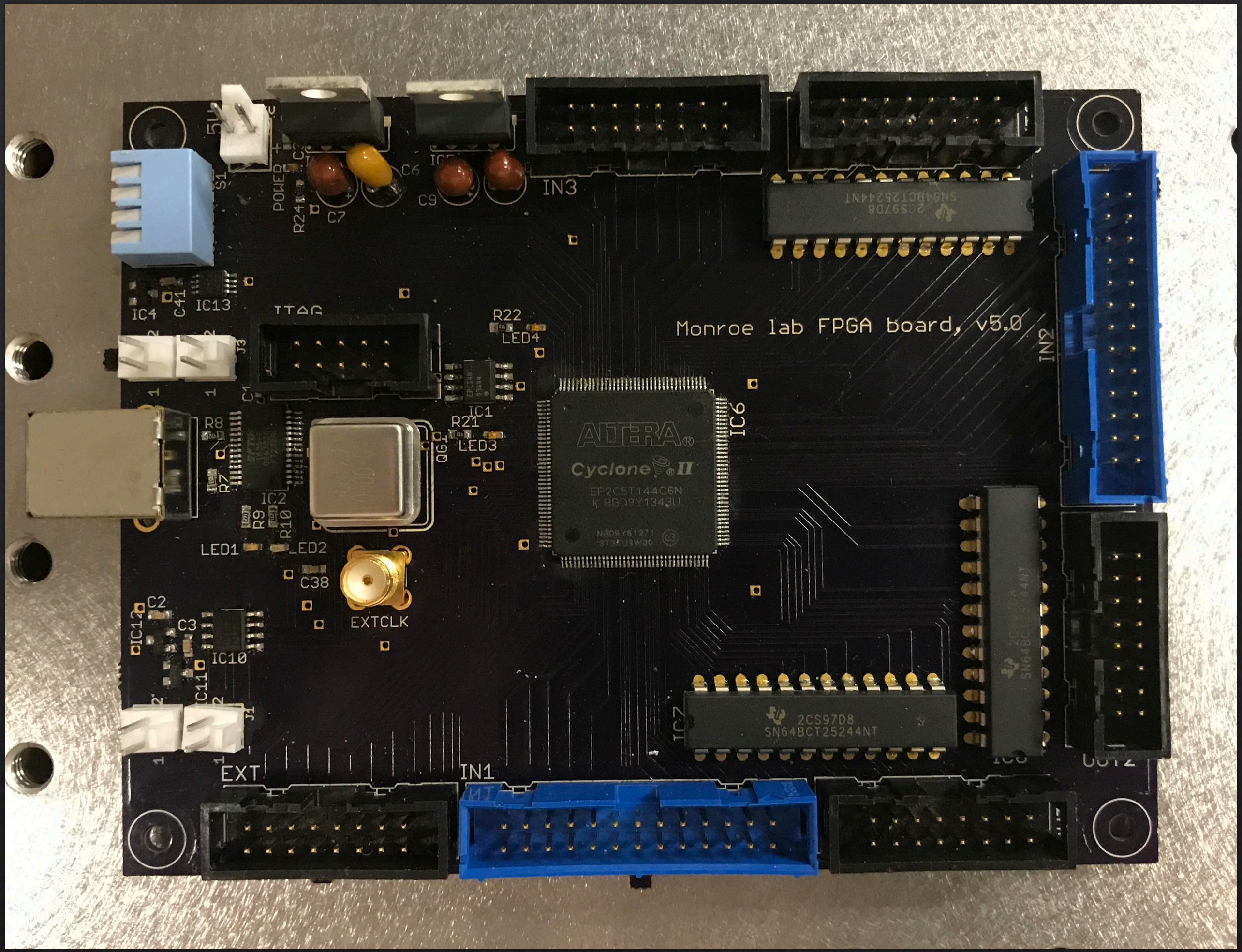
- Custom FPGA, Input, DDS and LCD Display boards
- Designs from Chris Monroe ion trapping group at University of Maryland
- IC chips and other pieces mounted by Shan Zhong
- My goal was to get these boards working





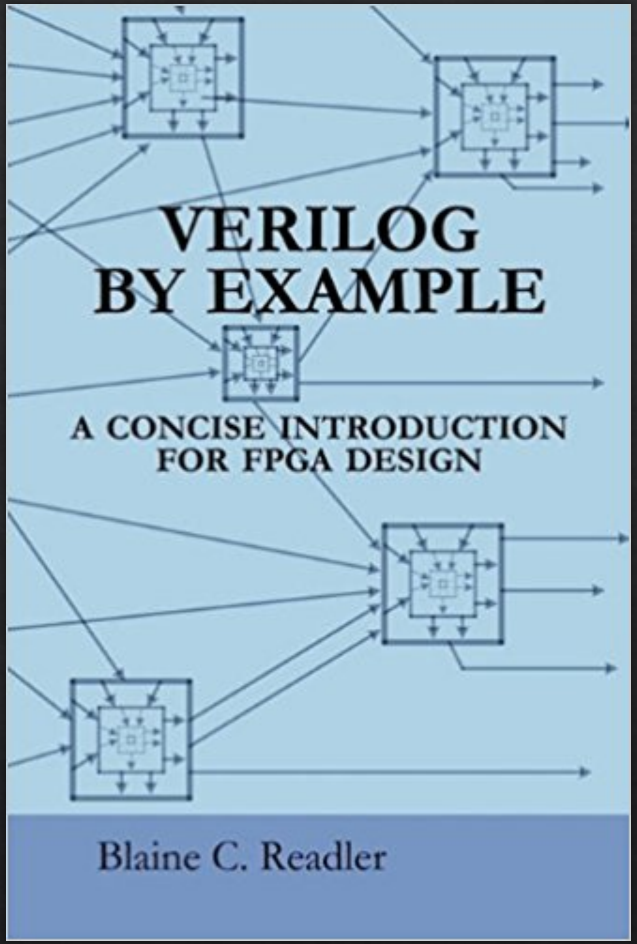
Debugging Boards

- Fixed power connections
- Touched up soldering
- Removed bridges on pins of FPGA chip
- Ordered missing parts (connectors, clock)
- Ordered USB Blaster programming device



FPGA Programming

- We use a hardware description language (HDL) called Verilog
- Syntactically similar to C
- Compiles into a physically realizable system of logic gates
- Can execute multiple processes in parallel
- Learned by reading Verilog by Example

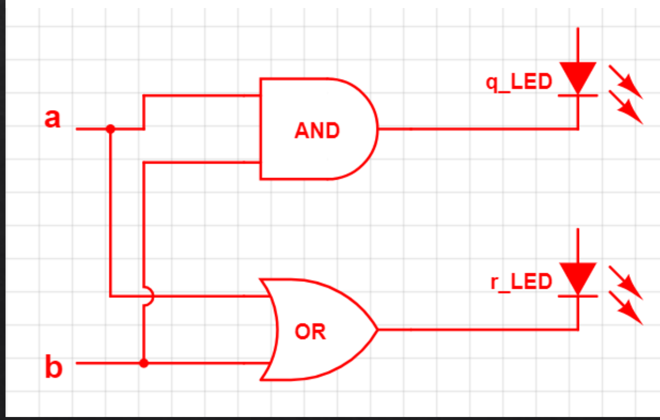


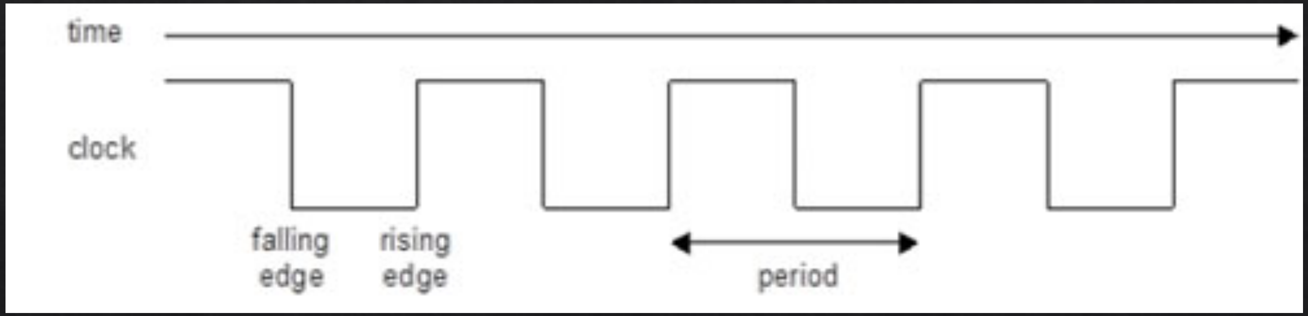
Verilog Example

- Contained within modules
- Data types: ports (inputs/outputs), wires, registers
- Logic gates are created with “assign”
- Clocks used for timing

Verilog

```
module gates(a, b, q, r);
input a, b;
output q, r;

assign q = a & b;
assign r = a | b;
endmodule
```

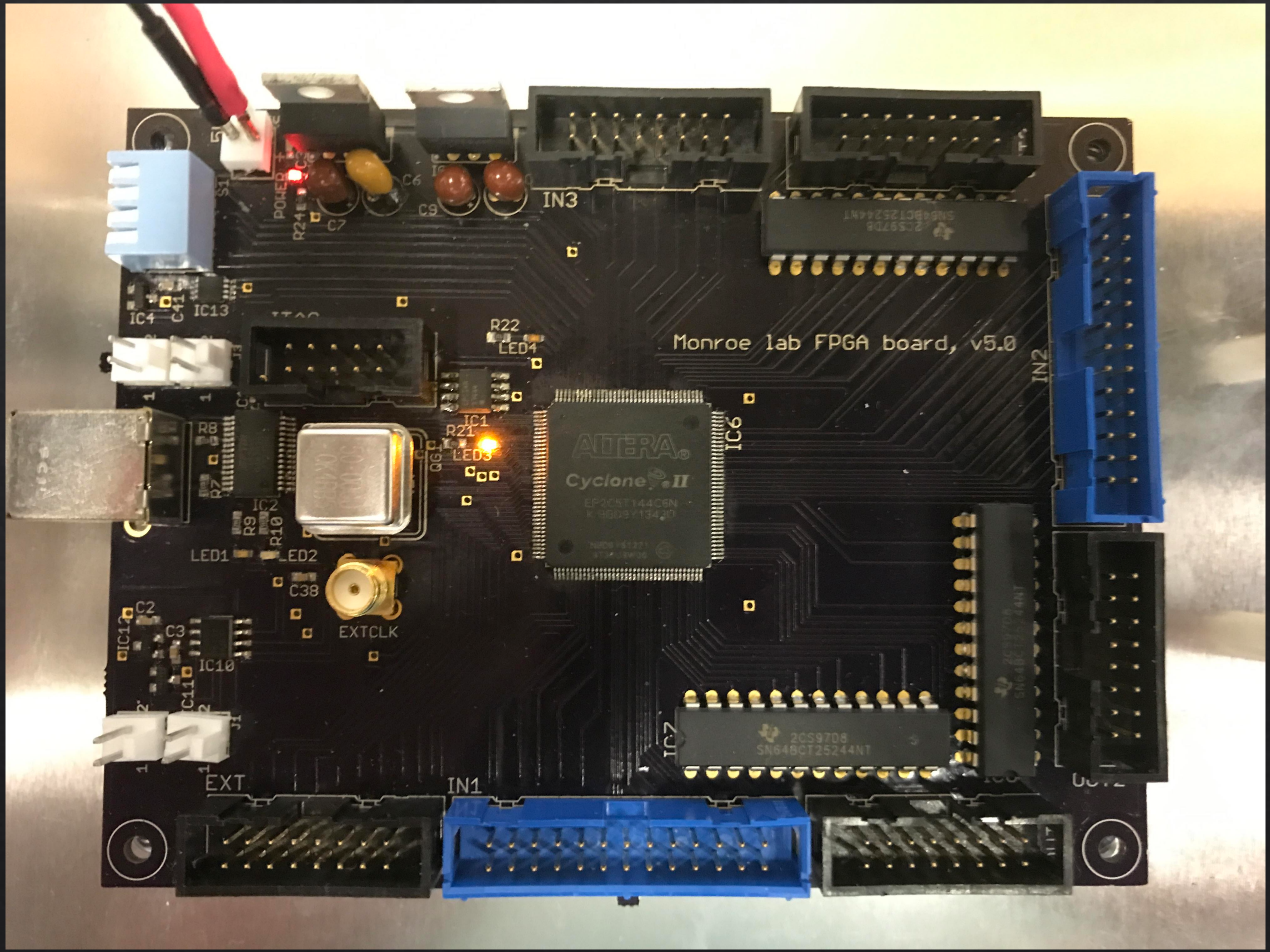




Test Program

aom_driver.v | dds_write.v | led_test.v | manual_freq.v | fifo_write.v | lcd

```verilog
//
// This is a test module for a Cyclone II FPGA.
//

// Module and ports
module led_test(
     clk,
     led);

// Assign inputs and outputs
     input clk;  // 50 MHz clock
     output led;

     reg [31:0] counter;

     // LED is toggled using the 23rd bit of the counter
     assign led = counter[22];

always @(posedge clk)
     begin
          counter <= counter + 1;

     end

endmodule
```



Compiler

- We use Quartus II software by Altera
- Compile code and assign pins
- Download program via USB to JTAG port
- JTAG is debugging port which all chips have
- Can also be used to program

Source Code

- We are using source code written by Steve Olmschenk from Joint Quantum Institute
- Program with full software for microwave control
- Designed to control DDS to produce 30 MHz sine wave
- Allows manual and remote control of signal using LabVIEW on computers in lab
- I added documentation to code package

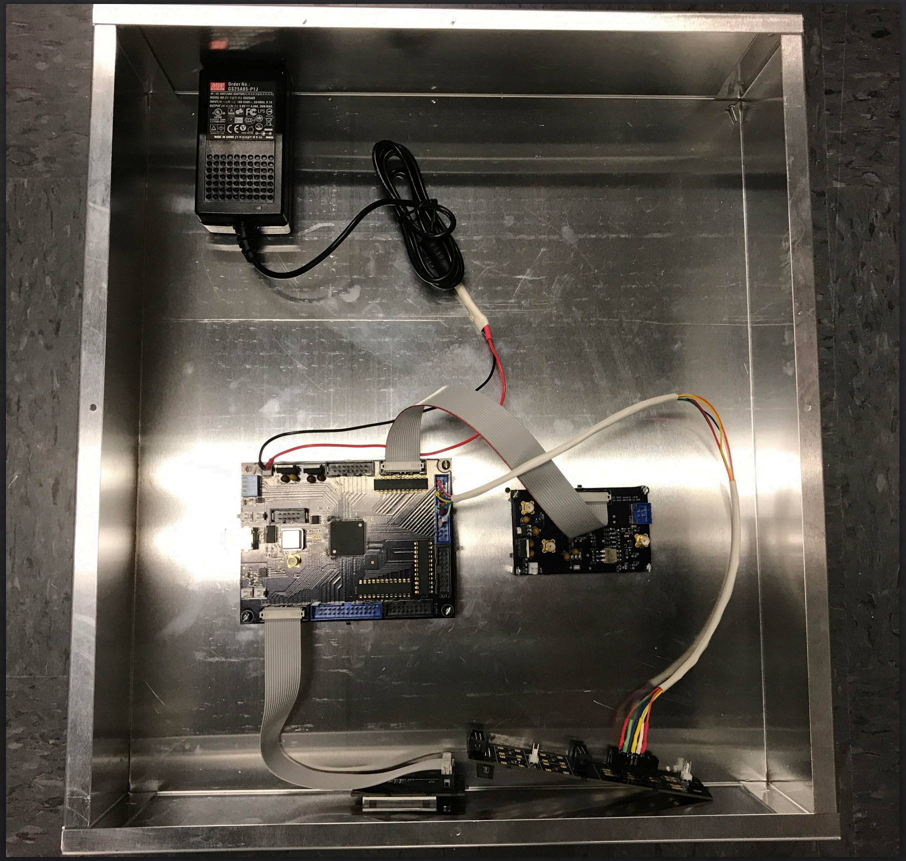
⋮

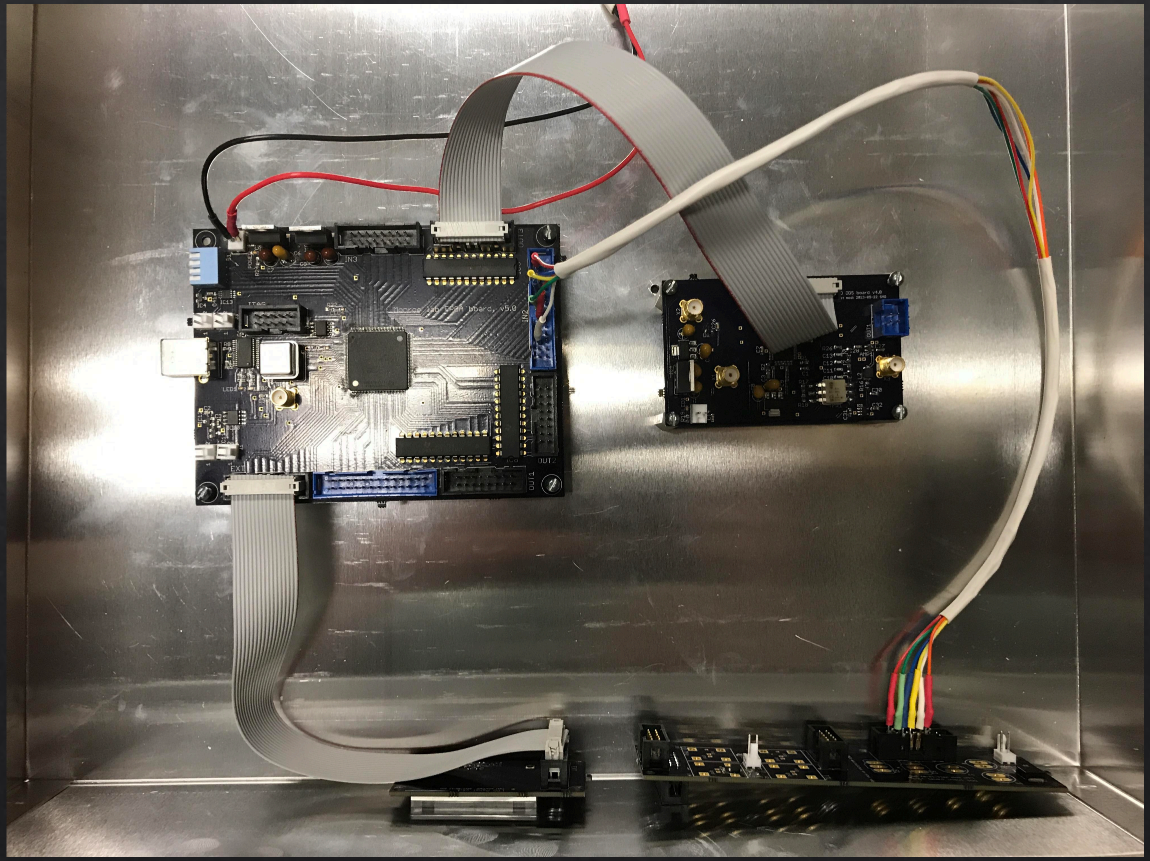
```
reg      [3:0]   freqChange; // determines how to change the frequency
reg      [2:0]   step;       // step in changing the frequency
reg      [1:0]   convStep;   // step in converting from binary to BCD
reg      [23:0]  waitCnt;    // counter to wait some time between triggers

parameter   MHz     = 32'h00A3D70A;
parameter   kHz     = 32'h000029F1;

always @(posedge clk)
   begin
      freqChange  <= {(active == 4'h0), (active == 4'h1), trigup, trigdown};

      case (step)
         0:  // look for frequency change trigger
            case (freqChange)
               4'b0110:    // +1 kHz
                  begin
                     if (freqDDS <= (32'h7FFFFFFF - kHz))     // check tha
                        begin
                           freqDDS <= freqDDS + kHz; // increases freq
                           freqBIN <= freqBIN + 1;   // same as above
                        end
```

⋮

Example from a module in source code

Enclosure





Conclusion:

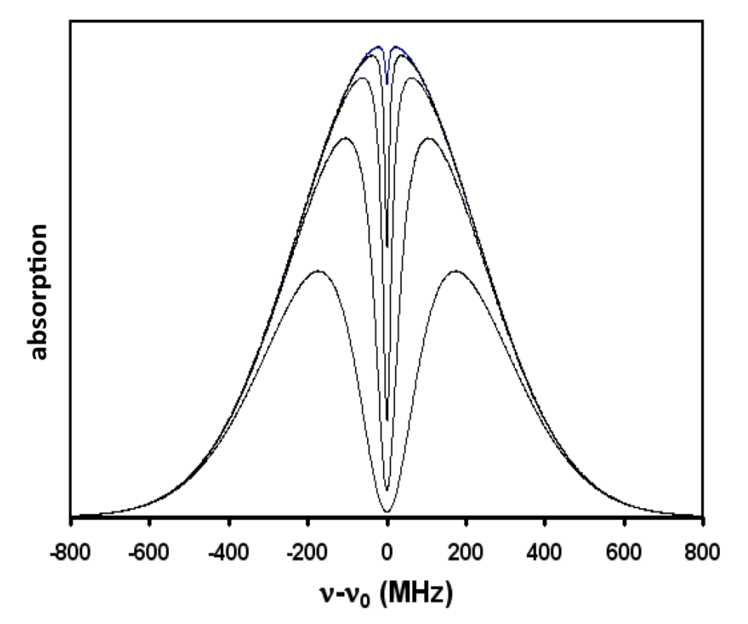
- I learned skills associated with building and debugging electronic circuits
- I learned how to program hardware using Verilog
- Successfully built a working PID controller that can serve as a feedback loop for amplitude stabilization
- Powered and programmed the FPGA board
- Began building an enclosure for mounting and housing the FPGA board and other components needed for the microwave generation

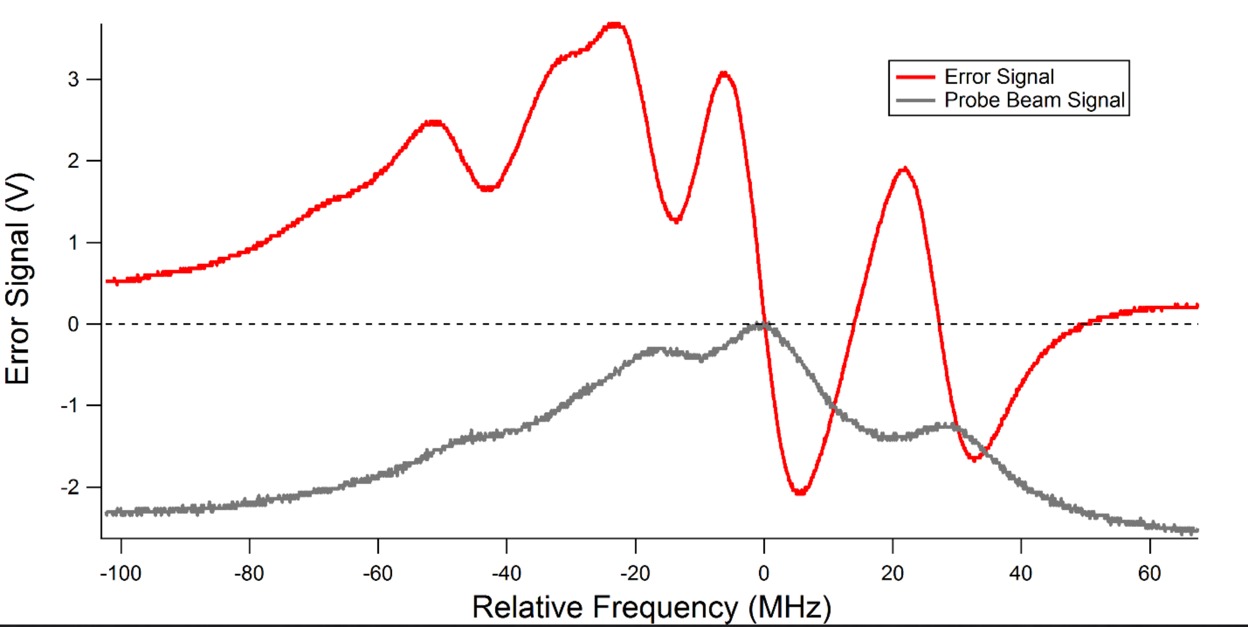
Outlook:

- PID controller can be implemented in the future if additional stabilization is needed
- Use FPGA and DDS to output 30 MHz sine wave
- Upon reaching BEC, FPGA controlled microwave pulses will allow for matter-wave interferometry experiments

Thank you.

Questions?


absorption
-800
-600
-400
-200
0
200
400
600
800
ν-ν0 (MHz)


Error Signal (V)
3
2
1
0
-1
-2
-100
-80
-60
-40
-20
0
20
40
60
Relative Frequency (MHz)
Error Signal
Probe Beam Signal