

# PIP-II IT Beam Pattern Generator / Kicker Driver Interfaces Discussion

John Dusatko AD-RF-LLRF  
PIP-II IT BPG/Kicker Discussion  
July 21, 2020

# Outline

- Who Am I?
- Introduction
- System Overview
- Hardware Interface
- Software Interface
- BPG HW Status
- Use Cases / Questions / Discussion

# Who Am I?

I have not had a chance to meet a lot of people due to the present situation, so you may be wondering who I am:

- John Dusatko
  - Started at Fermilab in March 2020 (AD-RF-LLRF)
  - Was at SLAC (AD-Controls / TID-RF) for the previous 21 years
    - Staff Engineer
    - Worked on many accelerator systems:
      - PEP-II LLRF
      - LCLS-I Timing System
      - LCLS-I MPS (beam loss monitors, ion chambers, etc.)
      - CERN SPS Longitudinal Feedback R&D
      - And some detector systems (BaBar, CMB Cryosensor)
  - Was at Univ of Chicago before that
    - Worked on HEP experiments (KTeV & CDF)

# Introduction

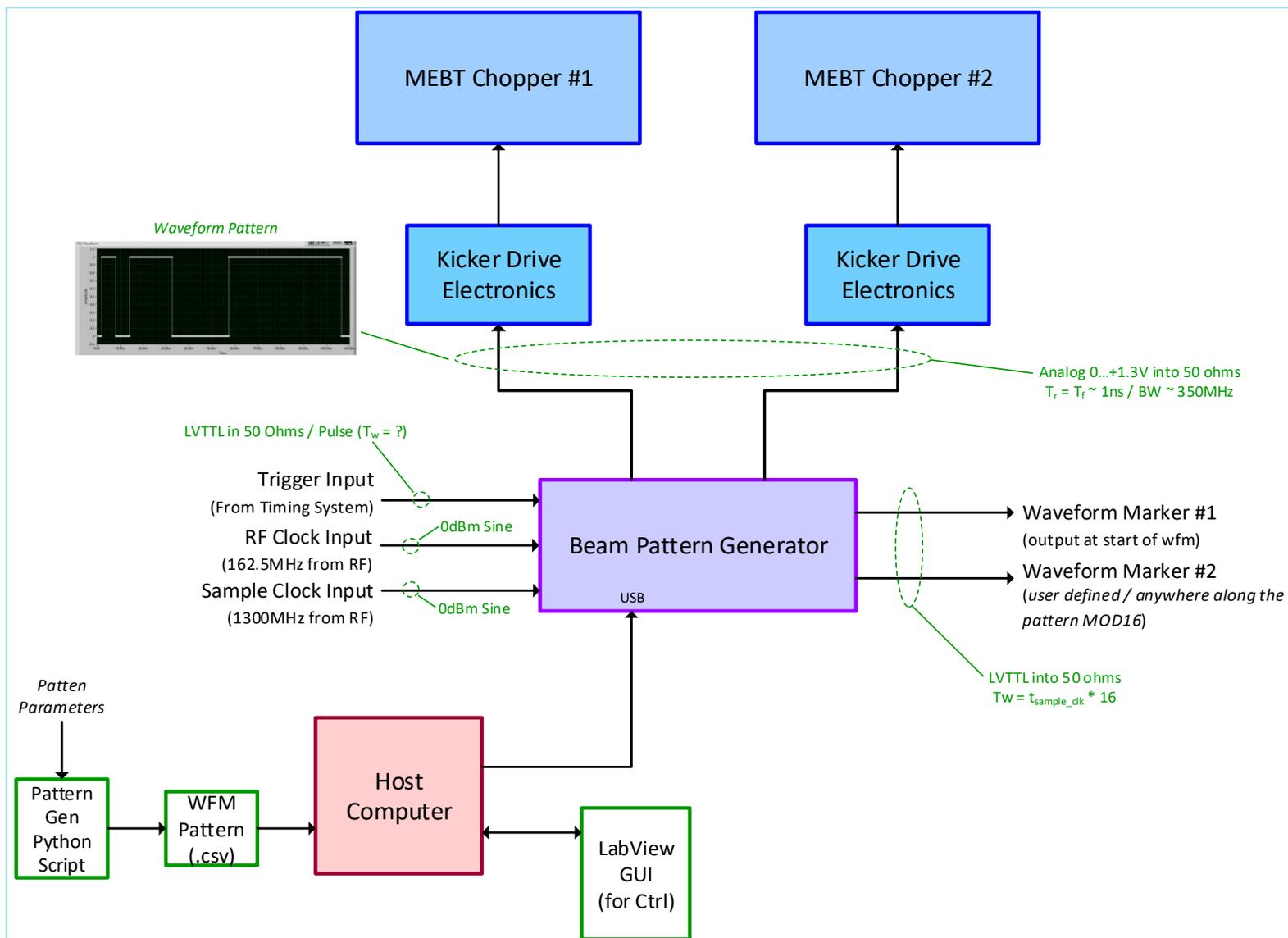
These slides are meant to inform you of:

- The system architecture
- The HW and SW interfaces (from the BPG perspective)

→ In order to stimulate discussion about the system interfaces as well as to understand use cases for the system

Caveat: I have come into the middle of this and am still learning about things...

# System Diagram



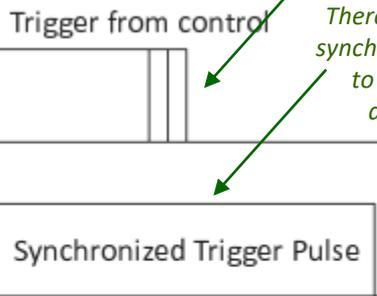
# System Timing Diagram

Trigger From Timing System

Trigger is synchronized, but absolute timing can walk over clock edges due to systematic conditions

Therefore, the BPG re-synchronizes the trigger to the 162.5 MHz accelerator RF

Timing Resolution = 40ps



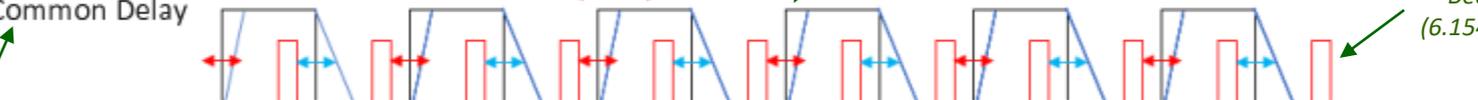
BPG Pattern (ideal) Selecting every other bunch here (black signal)

BPG Pattern (ideal) Selecting every other bunch here (black signal)

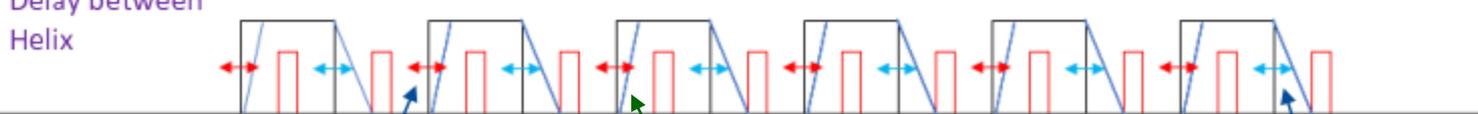
162.5 MHz Beam Bunch

Beam Bunch (6.154ns spacing)

Common Delay



Delay between Helix



Rising Edge Adjustment

Falling Edge Adjustment

The BPG provides an adjustable master delay between the trigger and start of the bunch pattern

BPG Pattern (actual) Showing ability to adjust rising and falling edges (same for all bunches) / (blue signal) / this allows us to select where in the bucket the bunch is placed

# BPG Hardware Interface

## Inputs

Source	Signal	Function	Specifications
Timing System	Trigger	BPG output "GO"	Pulse ( $T_{\text{wid}} = ?$ ) / LVTTTL into 50 ohms / Trigger Rep Rate = ??
RF System	162.5MHz	Trigger re-sync clock	Sine, 0dBm into 50 ohms
RF System	1300MHz	AWG sample clock	Sine, 0dBm into 50 ohms

## Outputs

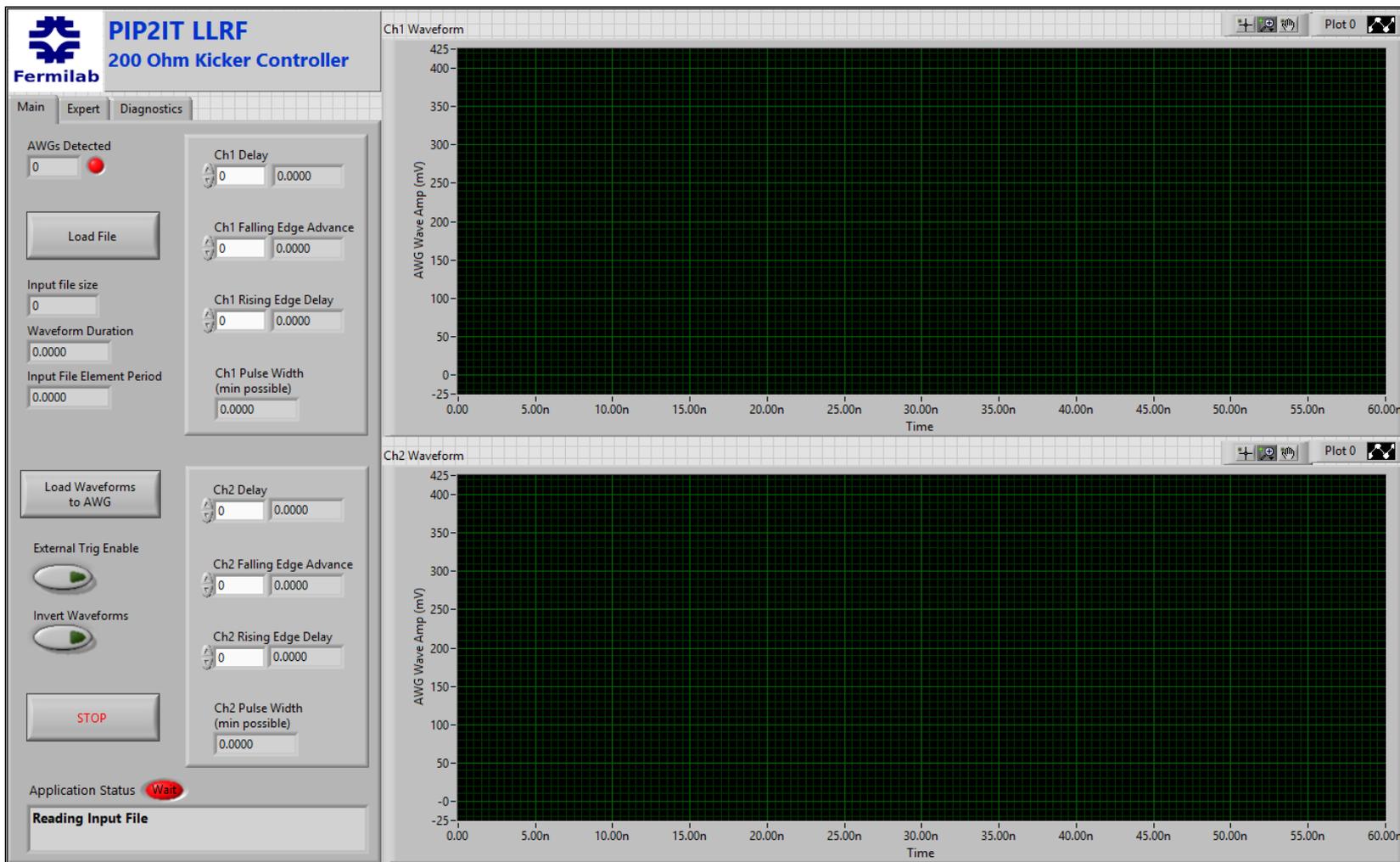
Destination	Signal	Function	Specifications
Kicker Drive Electronics	AWG Analog Output	Beam Pattern Drive	Analog 0...+1.3V into 50 ohms
<td / Oscilloscope>	AWG Marker #1	Strobe Pulse, marks the start of AWG pattern	Pulse ( $T_{\text{samp\_clk}} * 16$ ) / LVTTTL into 50 ohms
<td / Oscilloscope>	AWG Marker #2	Strobe Pulse, user-programmable	Pulse ( $T_{\text{samp\_clk}} * 16$ ) / LVTTTL into 50 ohms

# BPG Software Interface - General

## BPG Main Software Entities

Item	Function
LabView GUI	Main user interface for the system / controls & coordinates all functions
Pattern File	.CSV file containing the bunch pattern, generated by Python script / read by LabView GUI
Pattern Generator Script	Python program / Generates the .CSV file based on user input parameters

# BPG Software Interface – LabView GUI



# BPG Software Interface – LabView Controls

## Labview Software Controls

Control	Function	Comments
<b>Ext Trigger Enable</b>	Enables system to be triggered	When ena'd, the system will generate output waveforms from the timing system trigger
<b>Load File</b>	Loads in the .CSV waveform pattern	Loads in pattern file. Note that the same pattern is used for both AWG (kicker channels)
<b>Load Wfms to AWG</b>	Loads the processed wfm data to the AWG	Time-scaled/adjusted pattern is uploaded to AWG over USB
<b>Invert Waveform</b>	Inverts Polarity of AWG waveform signal	Applies to <u>both</u> AWG channels
<i>The following controls are independent to each channel (n = 1 or 2):</i>		
<b>Chn Delay</b>	Delay Start of waveform	Delays the output start of the entire pattern from the input trigger (resolution = 40ps)
<b>Chn Falling Edge Advance</b>	Moves falling edge fwd in time	Sets delay of pulse falling edge, adjustment resolution = 40ps / applies to all pulses (buckets) in pattern
<b>Chn Rising Edge Delay</b>	Moves Rising edge fwd in time	Sets delay of pulse rising edge, adjustment resolution = 40ps / applies to all pulses (buckets) in pattern

# BPG Hardware

## PIP-II IT Prototype System

- Built / Installed / Tested: 2017
- Verified on single kicker
- Things were “air-wired”
- **This latest work improves on the prototype:**

### Package into a proper chassis

Uses same COTS AWG as before

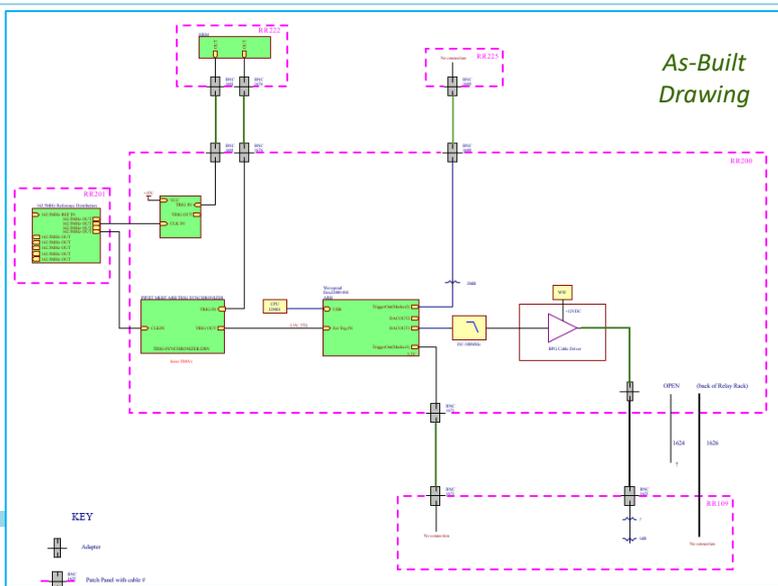
HW Design improvements:

Sync box – add 2<sup>nd</sup> DFF

Better driver circuit

Power Distribution board

System SW: Using original LabView application with any needed improvements



# BPG Chassis

Chassis Layout

3D CAD Rendering

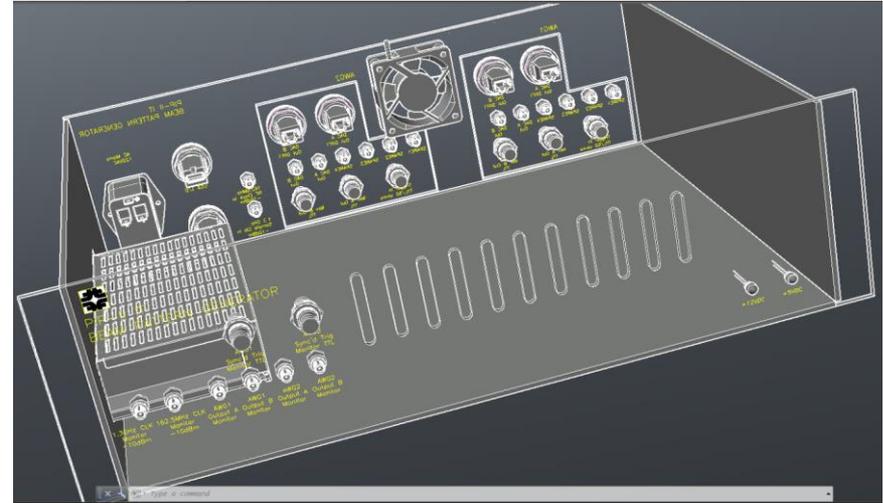
## Packaging of PIP-II IT BPG:

Components will be packaged into a 3U,  
19-inch rack mount chassis

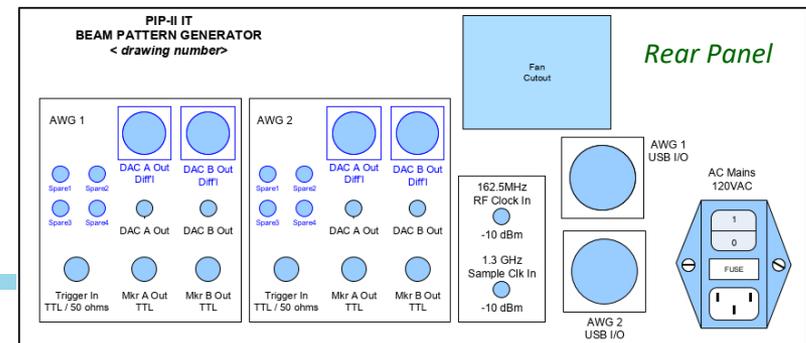
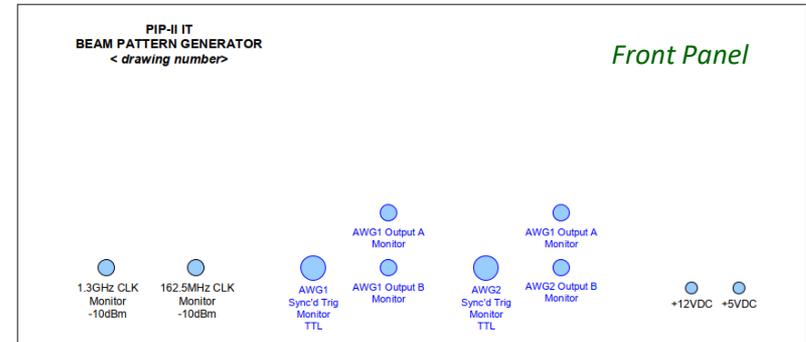
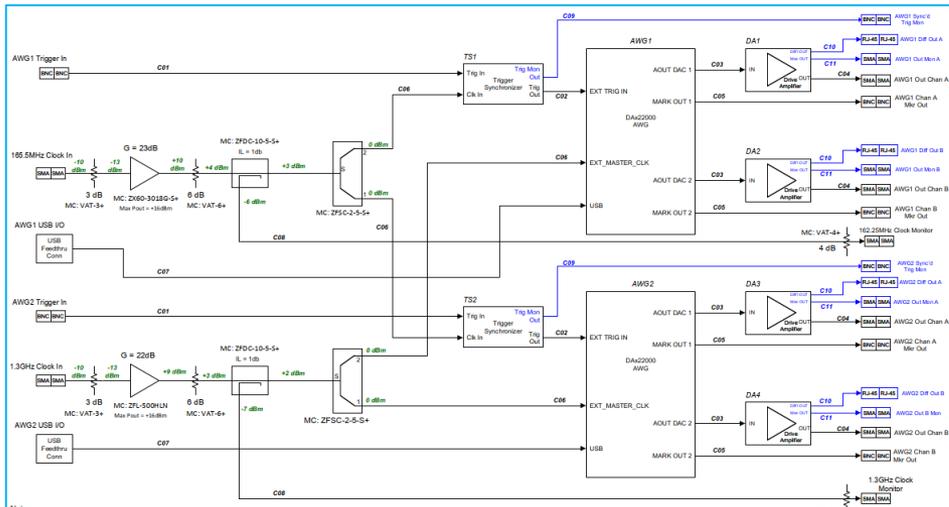
There will be four channels:

- (2) MEBT Choppers
- Booster Injection
- Spare/LEBT

Designed three new support electronics  
boards: Power Distribution Bd, Trigger  
Sync Bd, Drive Amplifier Bd



## Signal Interconnect





# System Use Cases

How will the system be used (for PIP-II IT)?

- 1) When the BPG is disabled, how is the system “safe”? (beam is always being deflected? Or??)
- 2) Set and Forget? (presumably this is after everything has been adjusted...)
- 3) Interactive Tuning (for initial time-up):
  - How will we accomplish this? (set up oscilloscope to display relevant signals, adjust delay knobs?)
- 4) Special Cases?
- 5) Dynamic tuning of delays (current setup only allows human interaction for delay adjustments)
- 6) Anything else? (special concerns, operating modes, etc.)

# Open Questions

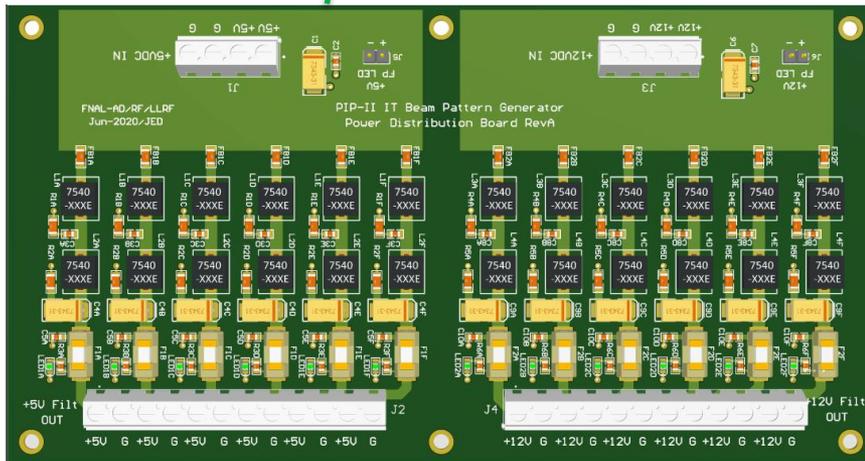
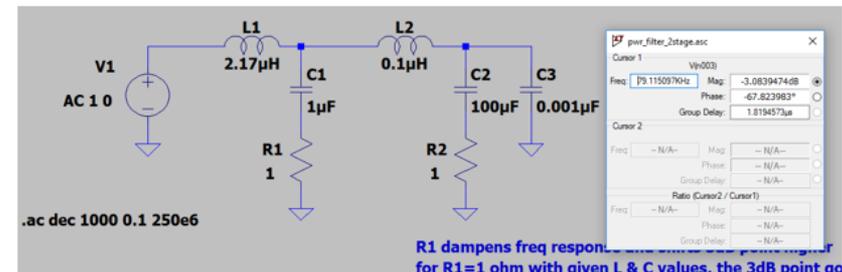
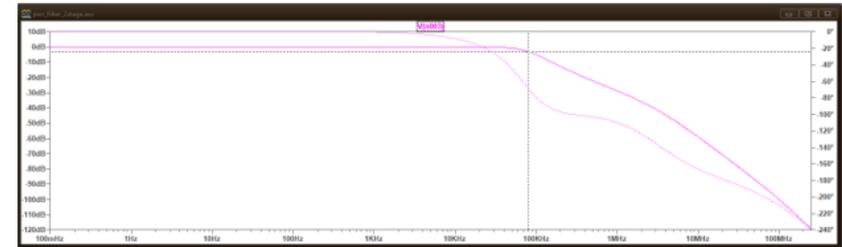
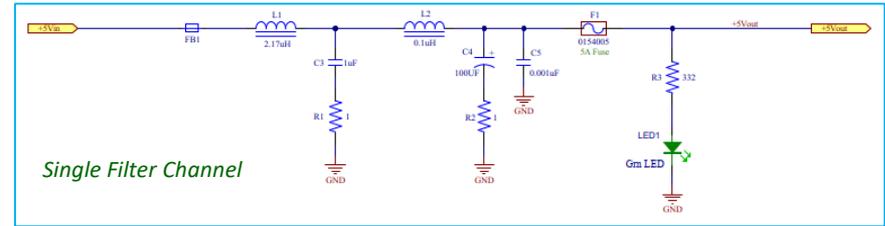
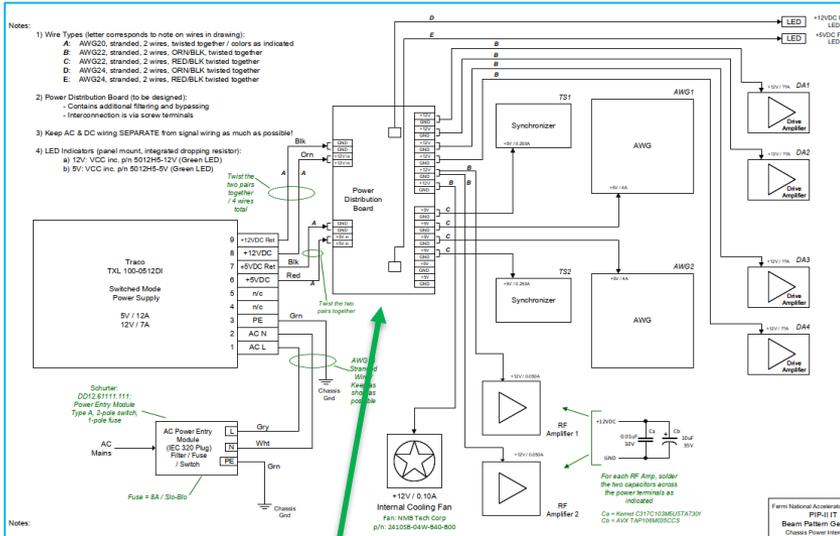
How will the system be used (for PIP-II IT)?

- 1) Are the HW interfaces sufficient for system operation?
- 2) Are the SW interfaces sufficient for system operation?
- 3) Are the SW GUI controls sufficient? (anything else needed? e.g. a way to save the settings)
- 4) Anything else?

**End of Talk**

# Supporting Slides

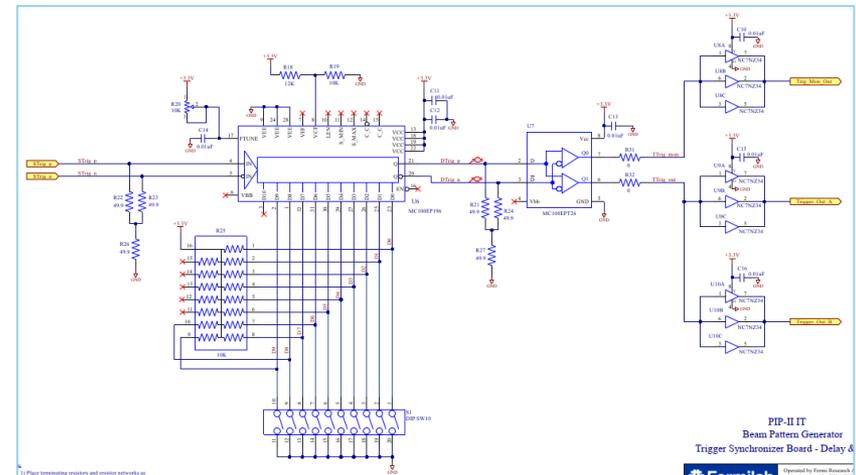
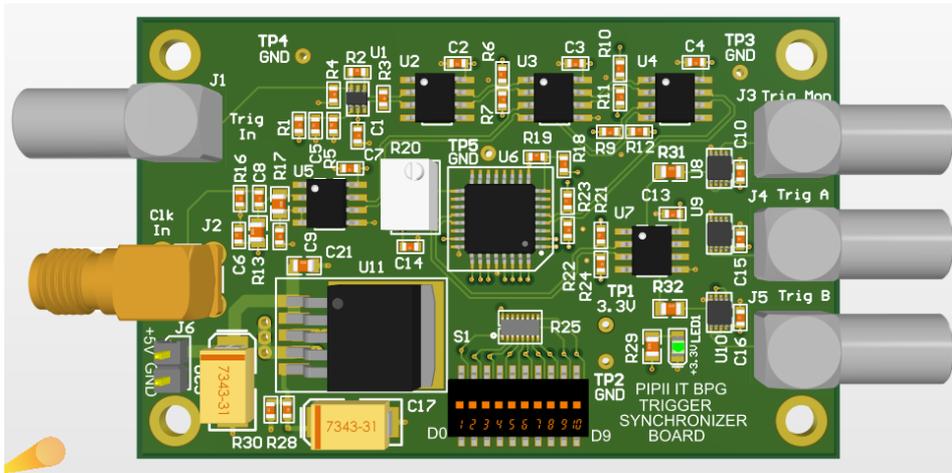
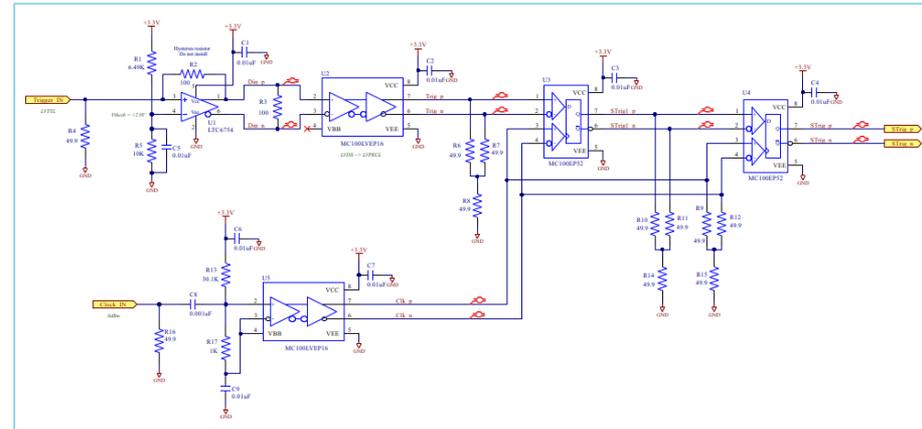
# BPG Power Distribution Board



# BPG Trigger Synchronizer Board

## Trigger Synchronizer Board

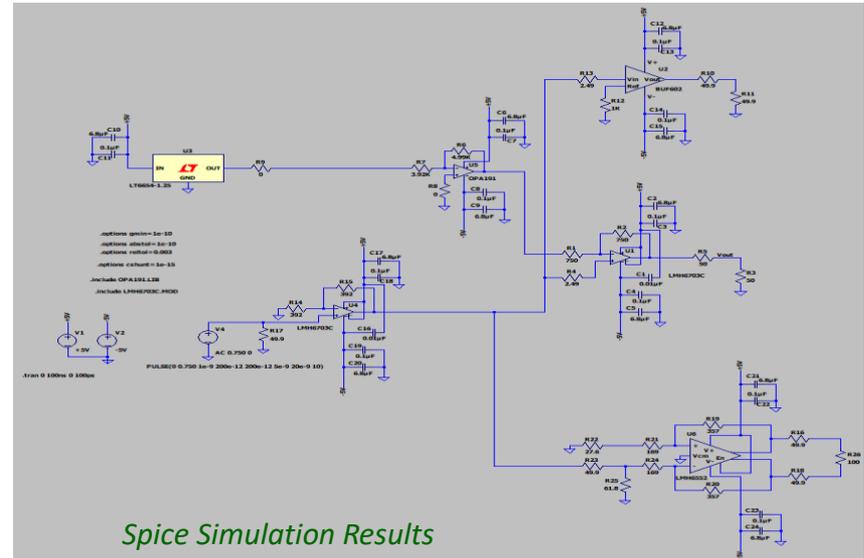
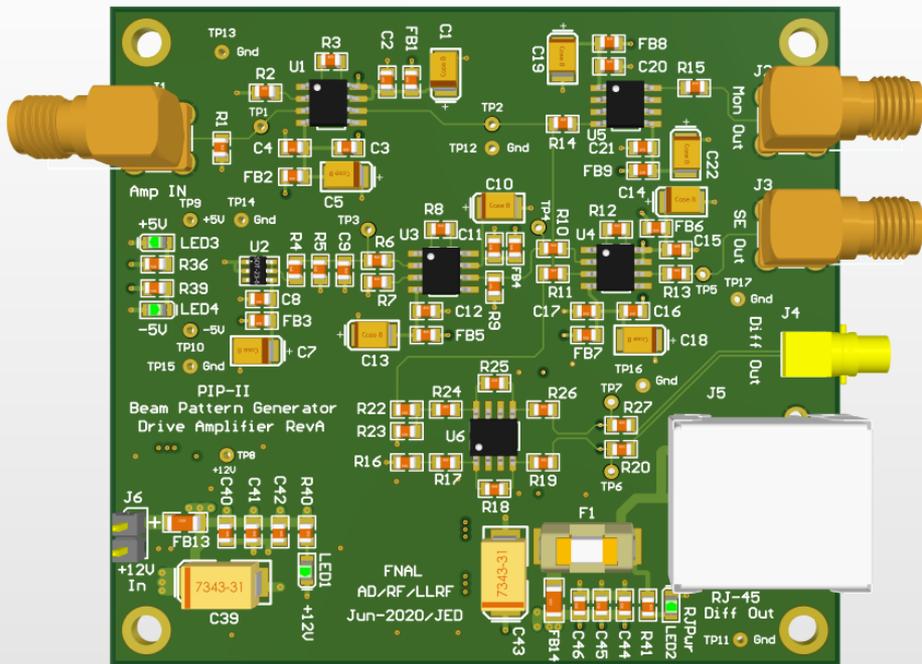
- Receives trigger from timing system
- Re-Synchronizes (samples) trigger to the 162.5MHz RF clock
- Contains adjustable delay line to ensure timing meets setup & hold times



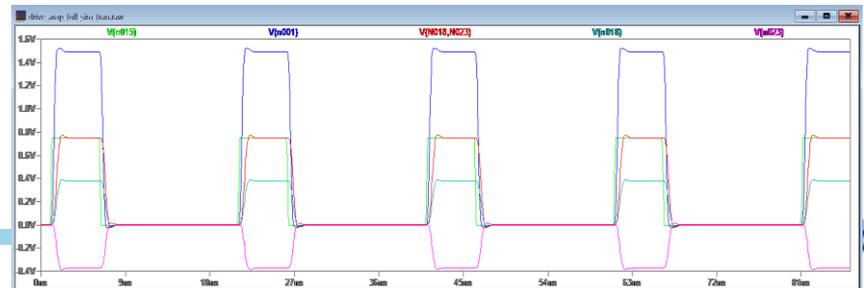
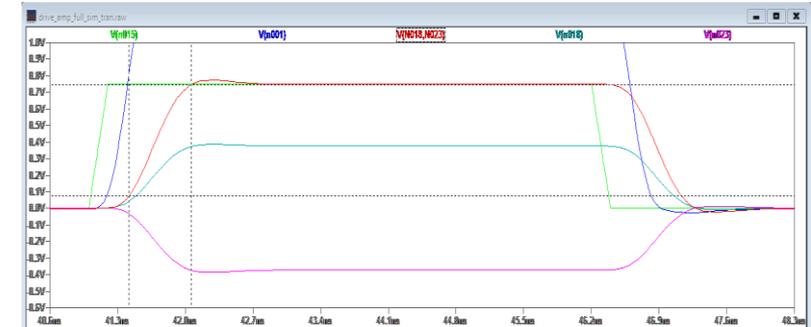
# BPG Drive Amplifier Board

## Drive Amplifier Board

- Translates 750mVp-p AWG output into 0...+1.3V signal for input to kicker drive electronics
- Single-ended output into 50 ohms
- Has differential drive option (using RJ-45 connector -or- Samtec Twinax coax conn)



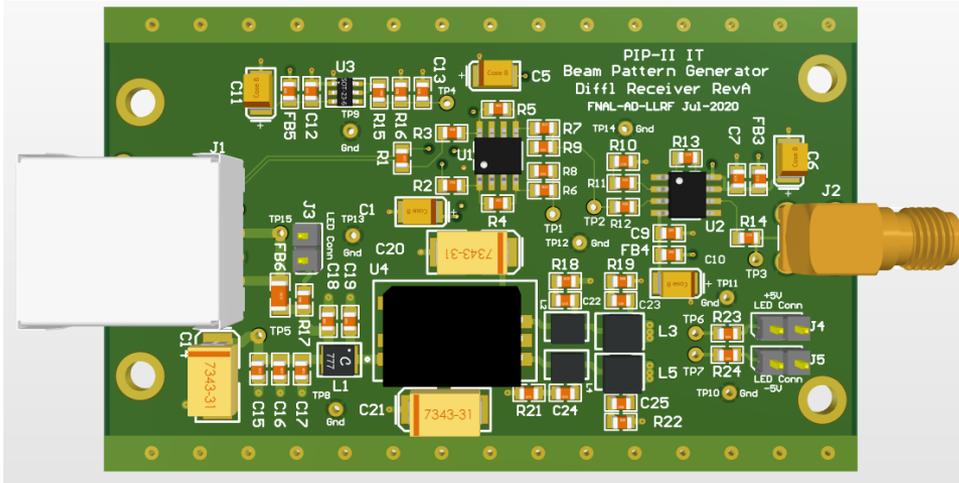
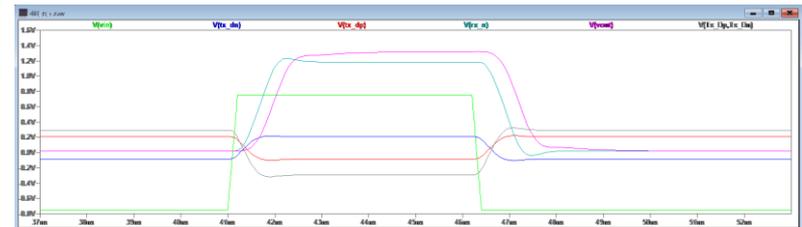
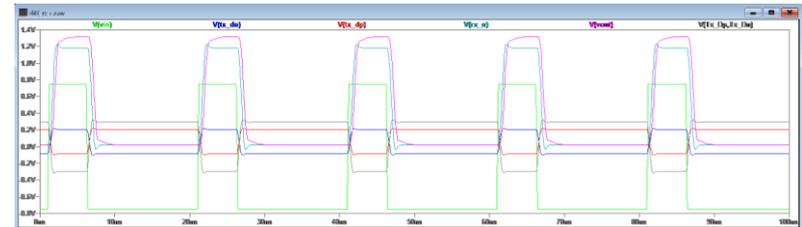
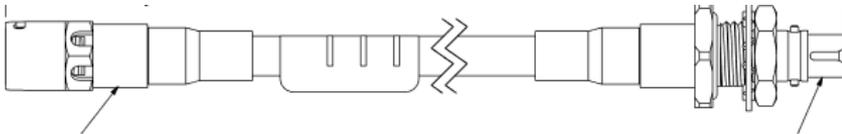
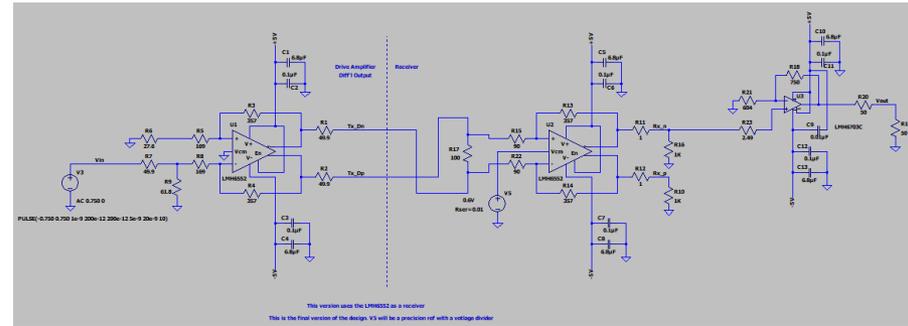
Spice Simulation Results



# BPG Differential Receiver Box

## Differential Receiver Box

- Receives differential AWG drive signal and translates back to level-shifted single-ended
- Small box would be placed near the kicker drive electronics
- Signal can be transmitted over Cat7 shielded Ethernet cable / Power is sent over cable as well
- Has option to use Samtec twinax coax (good to 4GHz)



Cat7 Shielded Solid PVC Cable

