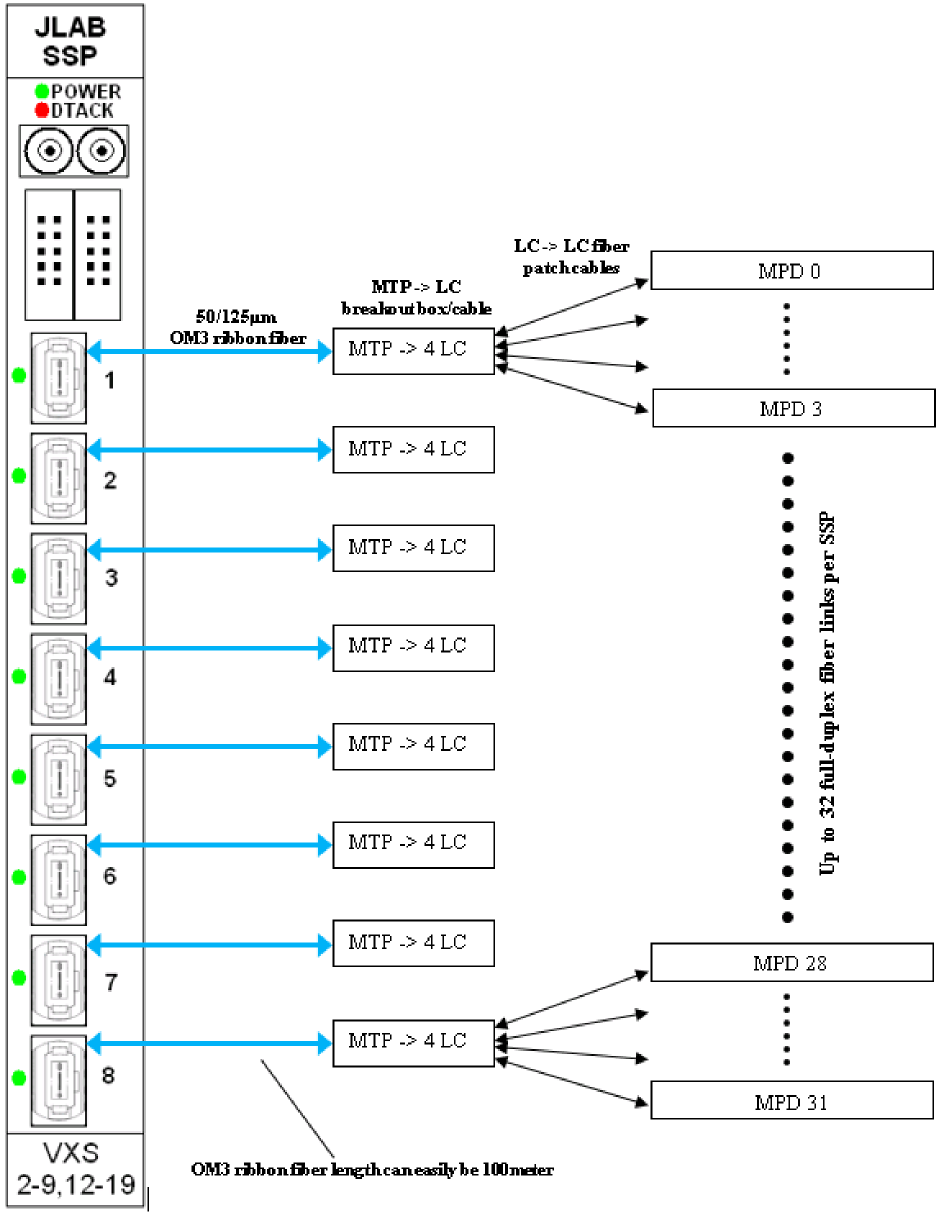
# SSP & MPD
# Zero suppression testing
# Updates

Ben Raydo

Danning Di

# SSP/MPD Zero Suppression Status


JLAB SSP
POWER
DTACK
1
2
3
4
5
6
7
8
VXS 2-9,12-19
50/125µm OM3 ribbon fiber
MTP -> LC breakout box/cable
LC -> LC fiber patchcables
MTP -> 4 LC
MPD 0
MPD 3
MPD 28
MPD 31
Up to 32 full-duplex fiber links per SSP
OM3 ribbon fiber length can easily be 100meter

## Event Size

- Without zero suppression:
  24Bytes/hit: 1 MPD (15APV, 6samples) = 46kBytes

- With zero suppression:
  12Bytes/hit: 1 MPD @ 50% Occupancy = 11.5kBytes

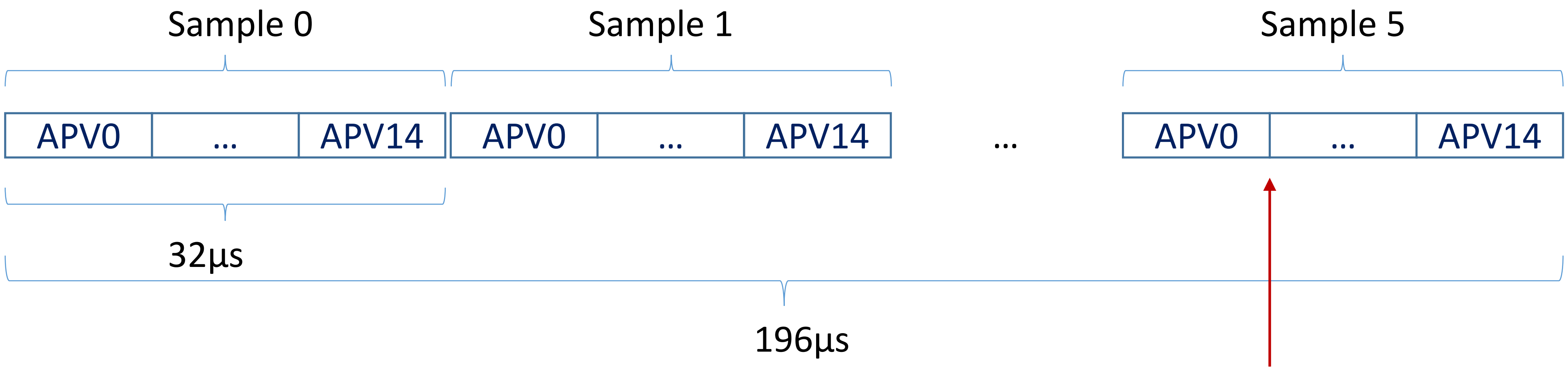
## Zero Suppression

- Common-mode subtraction must be done to efficiently zero suppression – done using Danning's algorithm.

- Common-mode subtraction and zero suppression is done in the SSP. First implementation is under testing that should be able to run up to 2kHz rate.

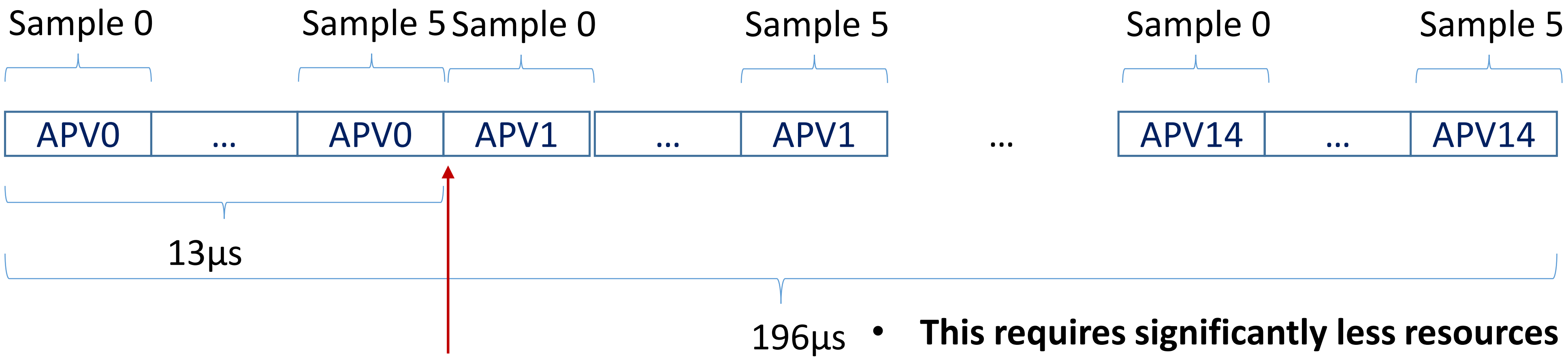
# MPD Data Format (sent to SSP over fiber)

**Current Data Format:**



- Common-mode/zero suppression can start to process here
- Must store last ~196µs of data

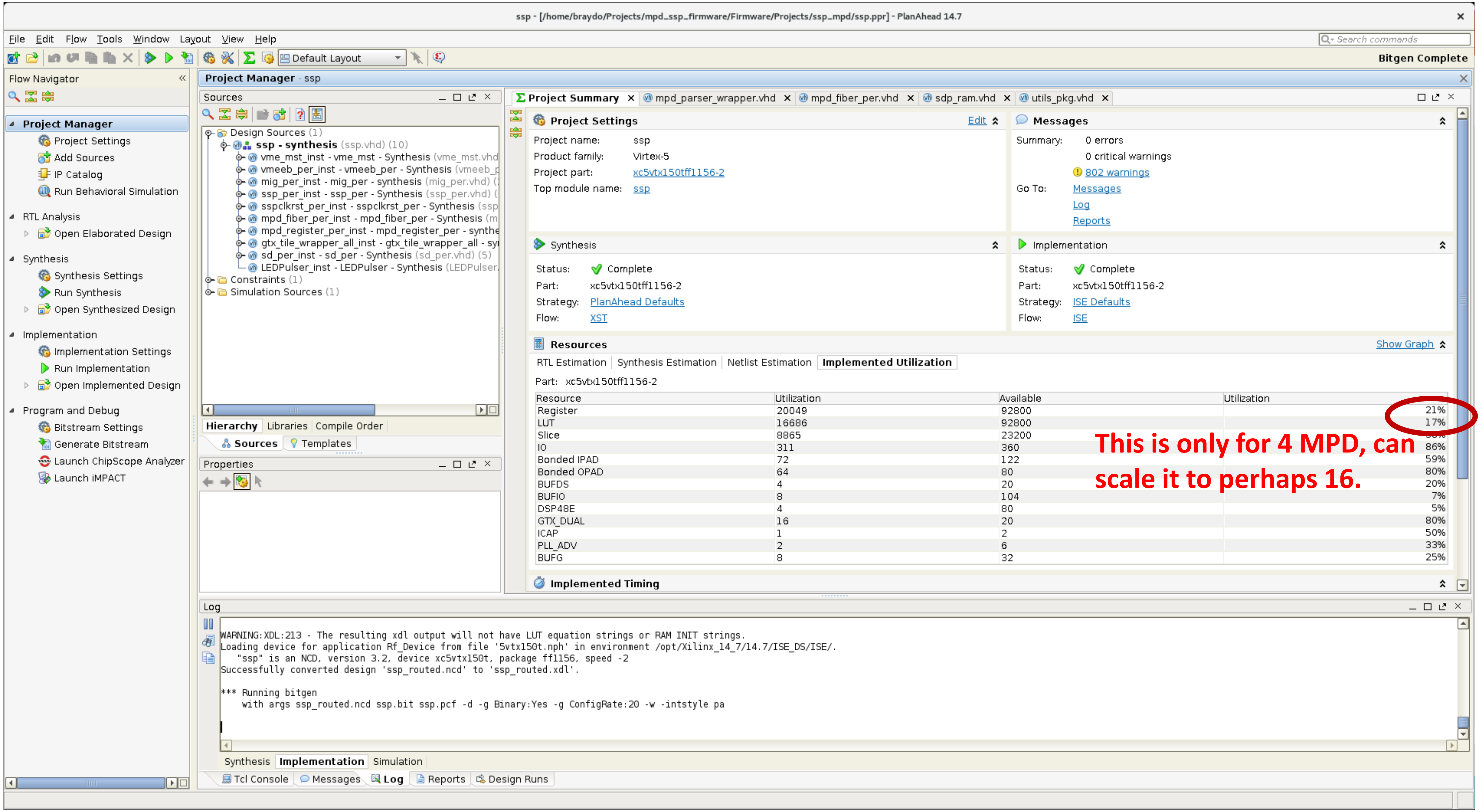
**Requested Data Order:**



- Common-mode/zero suppression can start to process here
- Must store last ~13µs of data

- **This requires significantly less resources in SSP to handle – scalable to 32 fiber ports**
- **Paolo is looking into this change on the MPD**

# SSP Firmware Compile Summary



Design Sources (1)
- ssp - synthesis (ssp.vhd) (10)
  - vme_mst_inst - vme_mst - Synthesis (vme_mst.vhd
  - vmeeb_per_inst - vmeeb_per - Synthesis (vmeeb_p
  - mig_per_inst - mig_per - synthesis (mig_per.vhd) (
  - ssp_per_inst - ssp_per - Synthesis (ssp_per.vhd) (
  - sspclkrst_per_inst - sspclkrst_per - Synthesis (ssp
  - mpd_fiber_per_inst - mpd_fiber_per - Synthesis (m
  - mpd_register_per_inst - mpd_register_per - synthe
  - gtx_tile_wrapper_all_inst - gtx_tile_wrapper_all - sy
  - sd_per_inst - sd_per - Synthesis (sd_per.vhd) (5)
  - LEDPulser_inst - LEDPulser - Synthesis (LEDPulser.

Constraints (1)

Simulation Sources (1)

**Project Settings**

| | |
|---|---|
| Project name: | ssp |
| Product family: | Virtex-5 |
| Project part: | xc5vtx150tff1156-2 |
| Top module name: | ssp |

**Messages**

| | |
|---|---|
| Summary: | 0 errors |
| | 0 critical warnings |
| | 802 warnings |
| Go To: | Messages |
| | Log |
| | Reports |

**Synthesis**

| | |
|---|---|
| Status: | ✓ Complete |
| Part: | xc5vtx150tff1156-2 |
| Strategy: | PlanAhead Defaults |
| Flow: | XST |

**Implementation**

| | |
|---|---|
| Status: | ✓ Complete |
| Part: | xc5vtx150tff1156-2 |
| Strategy: | ISE Defaults |
| Flow: | ISE |

**Resources**

RTL Estimation | Synthesis Estimation | Netlist Estimation | **Implemented Utilization**

Part: xc5vtx150tff1156-2

| Resource | Utilization | Available | Utilization |
|---|---|---|---|
| Register | 20049 | 92800 | 21% |
| LUT | 16686 | 92800 | 17% |
| Slice | 8865 | 23200 | 38% |
| IO | 311 | 360 | 86% |
| Bonded IPAD | 72 | 122 | 59% |
| Bonded OPAD | 64 | 80 | 80% |
| BUFDS | 4 | 20 | 20% |
| BUFIO | 8 | 104 | 7% |
| DSP48E | 4 | 80 | 5% |
| GTX_DUAL | 16 | 20 | 80% |
| ICAP | 1 | 2 | 50% |
| PLL_ADV | 2 | 6 | 33% |
| BUFG | 8 | 32 | 25% |

**Implemented Timing**

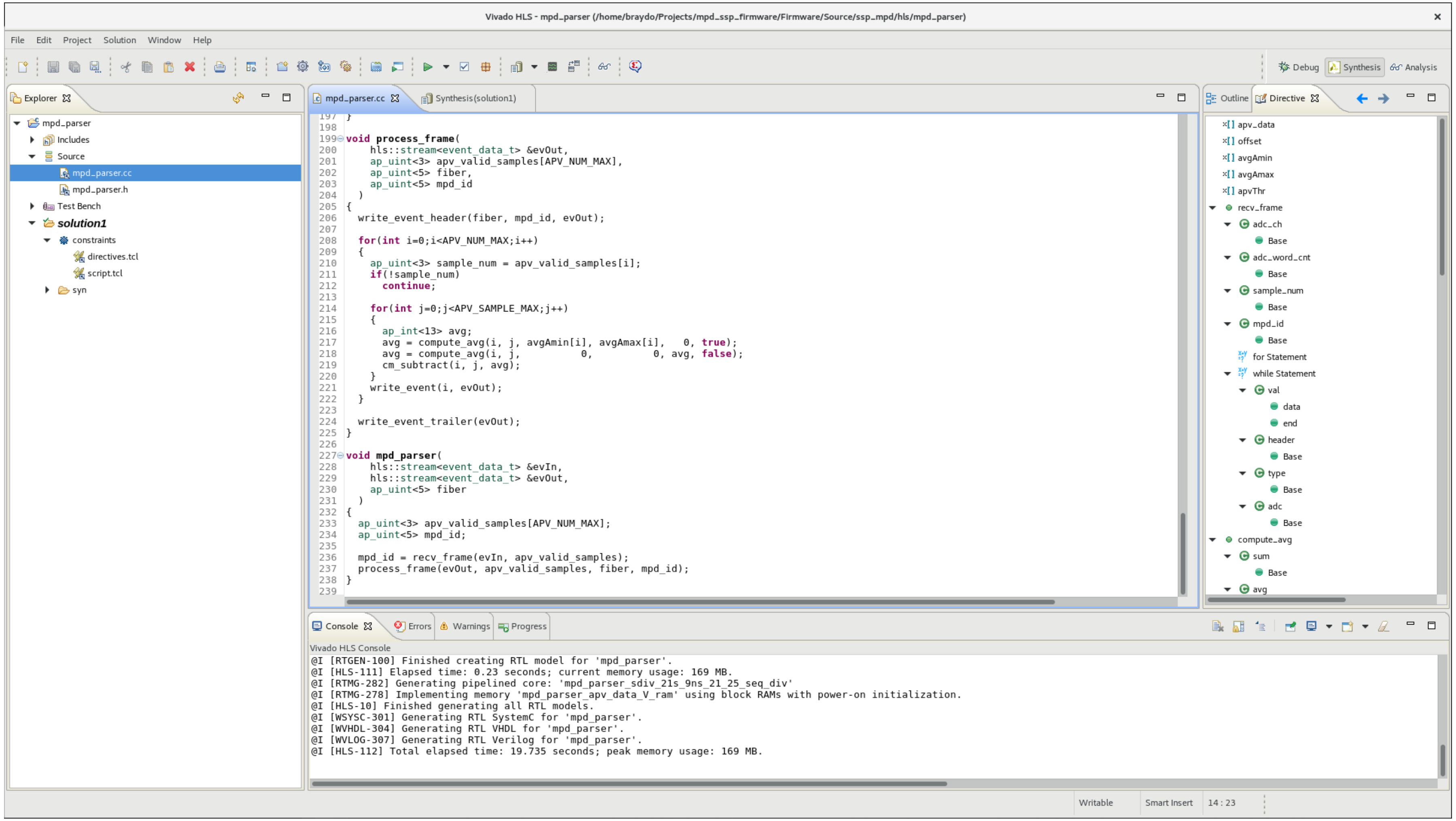
Log

```
WARNING:XDL:213 - The resulting xdl output will not have LUT equation strings or RAM INIT strings.
Loading device for application Rf_Device from file '5vtx150t.nph' in environment /opt/Xilinx_14_7/14.7/ISE_DS/ISE/.
   "ssp" is an NCD, version 3.2, device xc5vtx150t, package ff1156, speed -2
Successfully converted design 'ssp_routed.ncd' to 'ssp_routed.xdl'.

*** Running bitgen
    with args ssp_routed.ncd ssp.bit ssp.pcf -d -g Binary:Yes -g ConfigRate:20 -w -intstyle pa
```

This is only for 4 MPD, can scale it to perhaps 16.

# Using HLS C/C++ for algorithm



```cpp
}

void process_frame(
    hls::stream<event_data_t> &evOut,
    ap_uint<3> apv_valid_samples[APV_NUM_MAX],
    ap_uint<5> fiber,
    ap_uint<5> mpd_id
  )
{
  write_event_header(fiber, mpd_id, evOut);

  for(int i=0;i<APV_NUM_MAX;i++)
  {
    ap_uint<3> sample_num = apv_valid_samples[i];
    if(!sample_num)
      continue;

    for(int j=0;j<APV_SAMPLE_MAX;j++)
    {
      ap_int<13> avg;
      avg = compute_avg(i, j, avgAmin[i], avgAmax[i],   0, true);
      avg = compute_avg(i, j,          0,          0, avg, false);
      cm_subtract(i, j, avg);
    }
    write_event(i, evOut);
  }

  write_event_trailer(evOut);
}

void mpd_parser(
    hls::stream<event_data_t> &evIn,
    hls::stream<event_data_t> &evOut,
    ap_uint<5> fiber
  )
{
  ap_uint<3> apv_valid_samples[APV_NUM_MAX];
  ap_uint<5> mpd_id;

  mpd_id = recv_frame(evIn, apv_valid_samples);
  process_frame(evOut, apv_valid_samples, fiber, mpd_id);
}
```

```
@I [RTGEN-100] Finished creating RTL model for 'mpd_parser'.
@I [HLS-111] Elapsed time: 0.23 seconds; current memory usage: 169 MB.
@I [RTMG-282] Generating pipelined core: 'mpd_parser_sdiv_21s_9ns_21_25_seq_div'
@I [RTMG-278] Implementing memory 'mpd_parser_apv_data_V_ram' using block RAMs with power-on initialization.
@I [HLS-10] Finished generating all RTL models.
@I [WSYSC-301] Generating RTL SystemC for 'mpd_parser'.
@I [WVHDL-304] Generating RTL VHDL for 'mpd_parser'.
@I [WVLOG-307] Generating RTL Verilog for 'mpd_parser'.
@I [HLS-112] Total elapsed time: 19.735 seconds; peak memory usage: 169 MB.
```

- **Allowed quick testing on previous data since we can easily integrate this into programs that work with EVIO**
- **The tool is not very efficient at using resources, but seems like it may be a good fit for this project for the moment.**

# Status

- Danning’s common-mode algorithm is implemented in SSP and testing began this week using the VME based test stand.

- Algorithm is running and first glance appears to function, but much left to check (currently we need to debug a problem with writing some configuration registers to allow testing to proceed in depth).

- Current implementation limits:
  1. rate to <2kHz (assuming no VME bottleneck)
  2. 4 MPD per SSP (probably can scale to 16)

- MPD data re-ordering will allow a less resource intensive solution on SSP that will allow 32 MPD per SSP and 5kHz trigger rate processing.