DAMC02 Controller Firmware

Paweł Prędki
ppredki@dmcs.pl

Department of Microelectronics and Computer Science
Technical University of Lodz

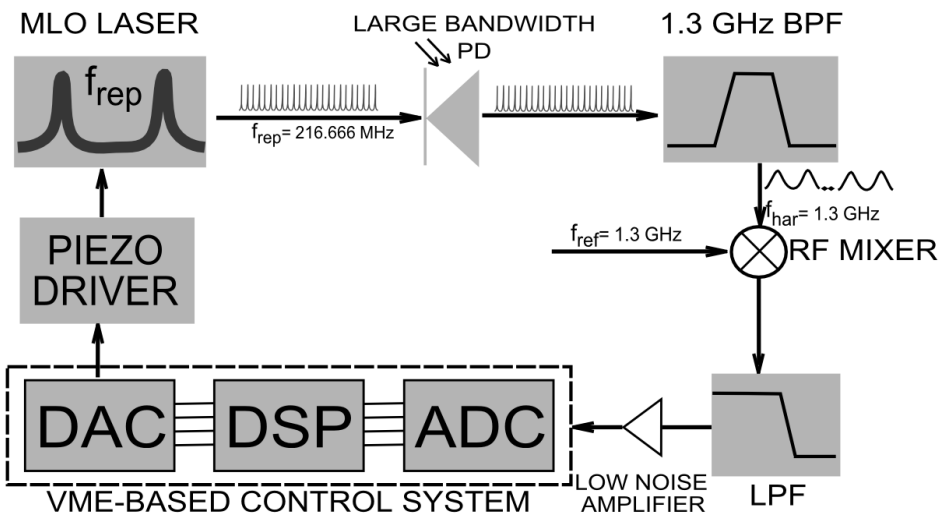
20.02.2013

Outline

Outline

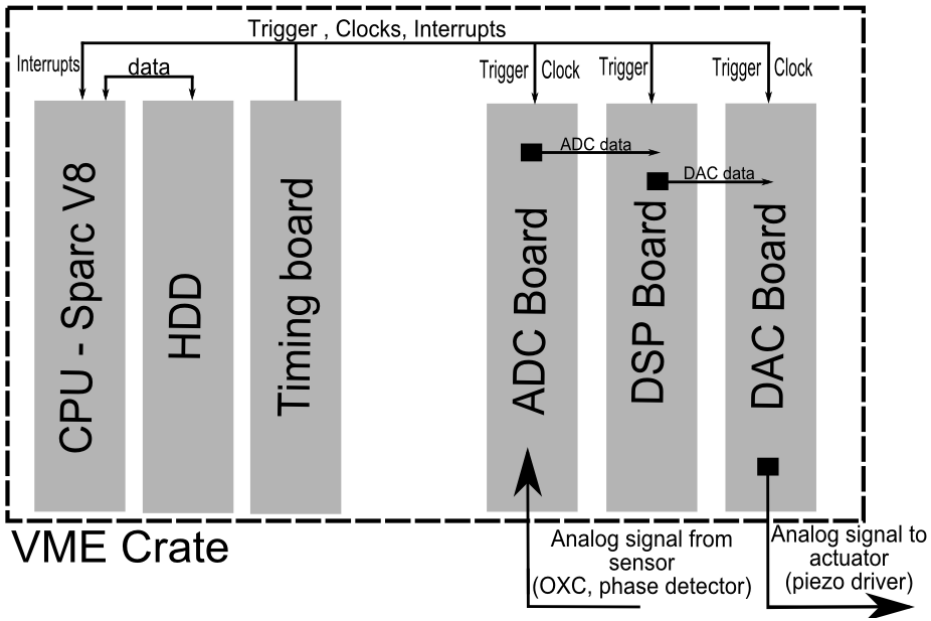
Motivation

- Design a versatile controller based on the DAMC02 board to be used for optical synchronization applications
- Use System Generator modules to implement the controller
- Makes the implementation more human-readable
- Makes it easier for non-VHDLers to implement a different controller
- Test application:



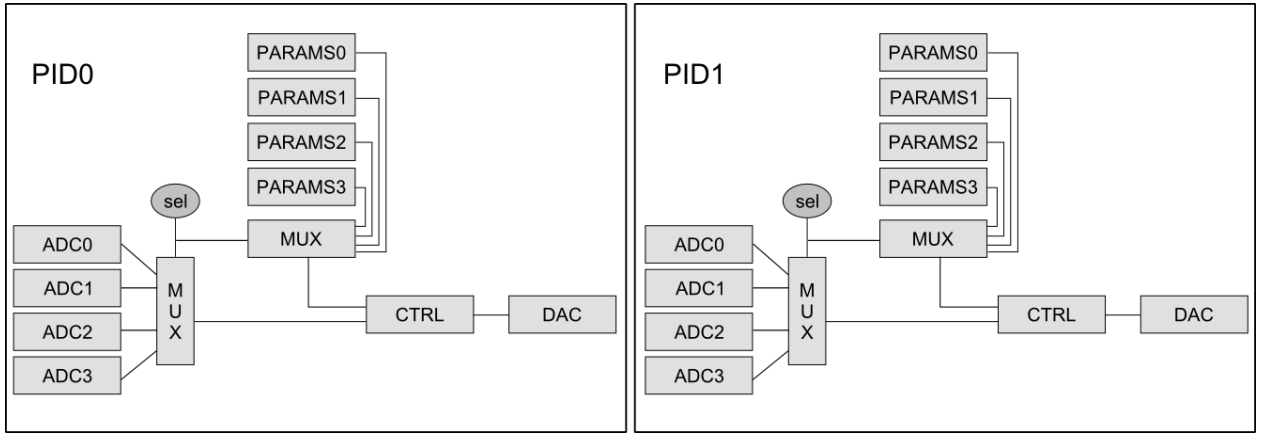
Outline

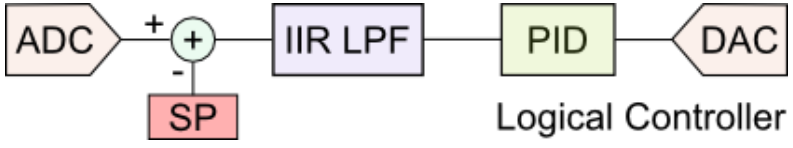
VME crate setup



- ADC board - 8-channel, 14-bit, up to 10 MSamples/s; All channels used in the system, 1 MHz clock used
- DAC board - 8-channel, 14-bit; Only two outputs available due to design flaws
- DSP - TMS320C6701 Texas Instruments chip, 164 MHz

DSP controller structure





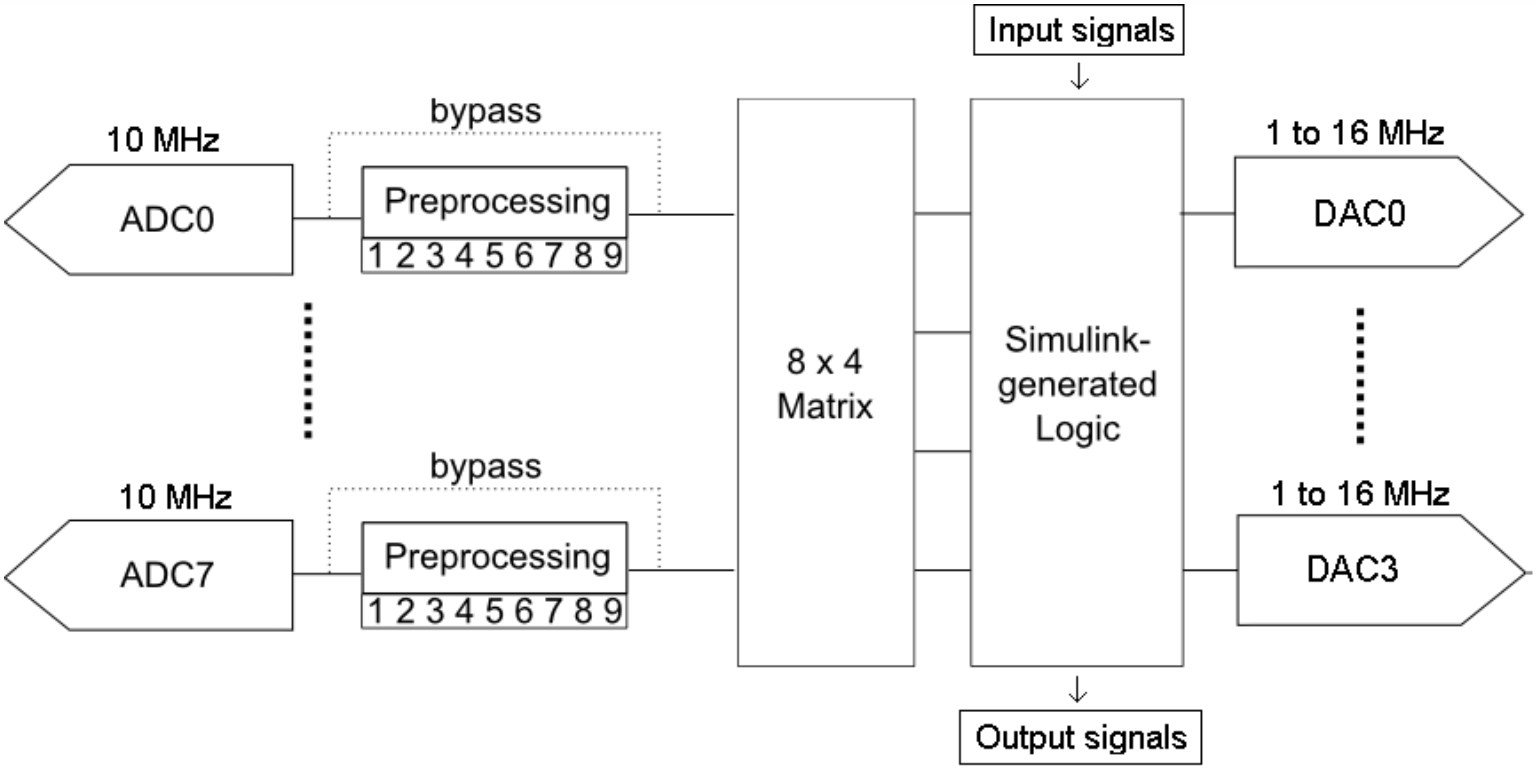
- Customizable parameters for each 'logical' controller:
- Data processing from 0 to 4 ADCs per PID - influences performance
- Independent operation of two PIDs
- PROBLEM: No parallel operation

VME status

- DSP system used throughout the synchronization system
- Improved performance over previous implementation
- VME has been around for many years
- Becomes obsolete - too slow, too limited, no hot-plug functionality
- Increasing requirements - need for a better solution

Outline

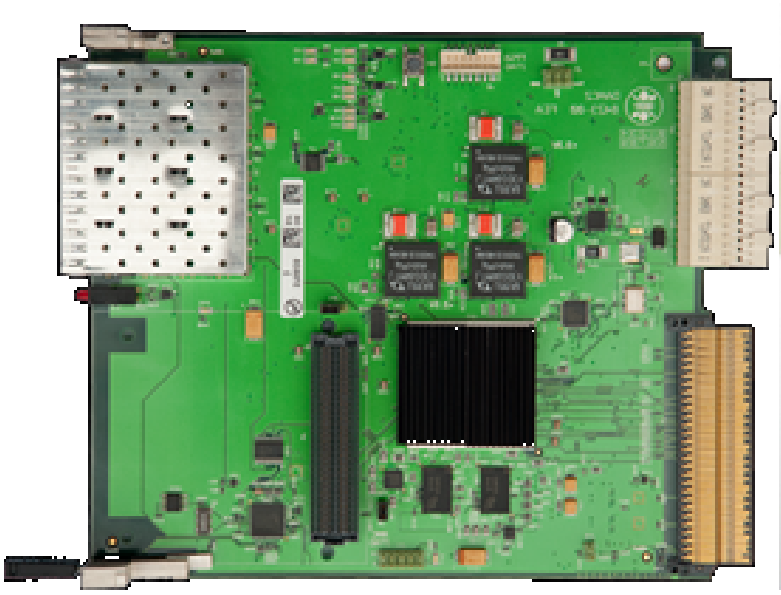
FPGA-based controller structure

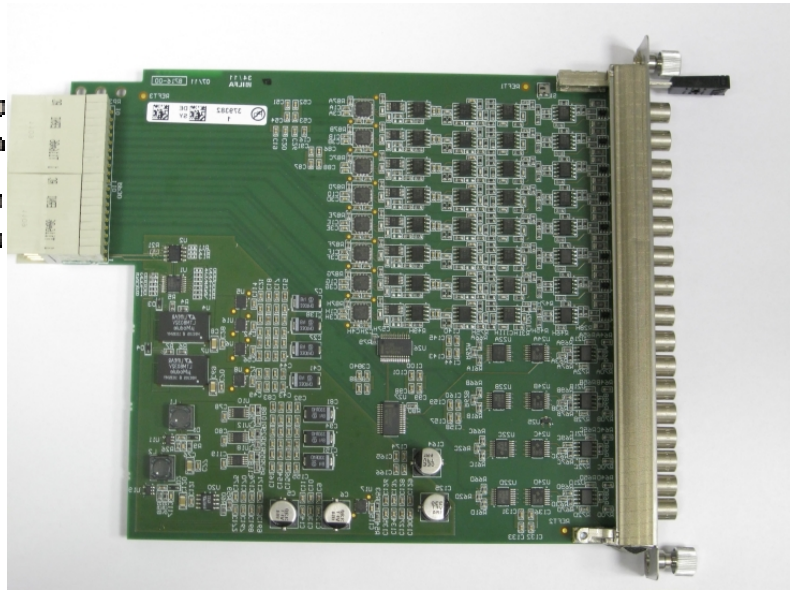


Possible preprocessing - switchable at runtime

- Moving average
- Data decimation
- FIR/IIF filtering (not yet implemented)
- Subtraction within the signal (not yet implemented)

Hardware used





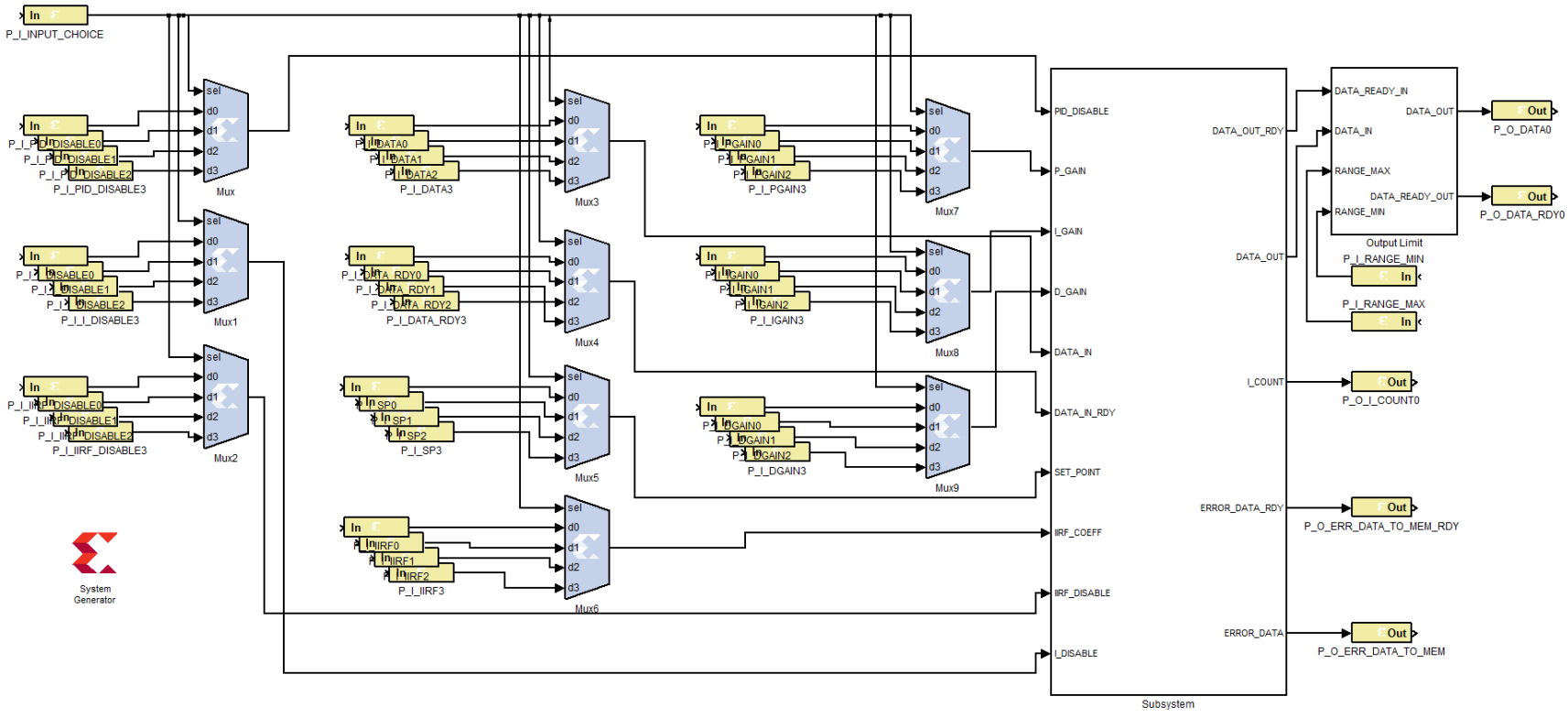
- AMC board - DAMC2
- RTM board - ADC-DAC RTM
- ADC - Analog Devices AD7626
- DAC - Texas Instruments DAC8580
- FPGA - Xilinx Virtex5 XC5VLX50T-1FF1136C

Controller highlights - Matrix

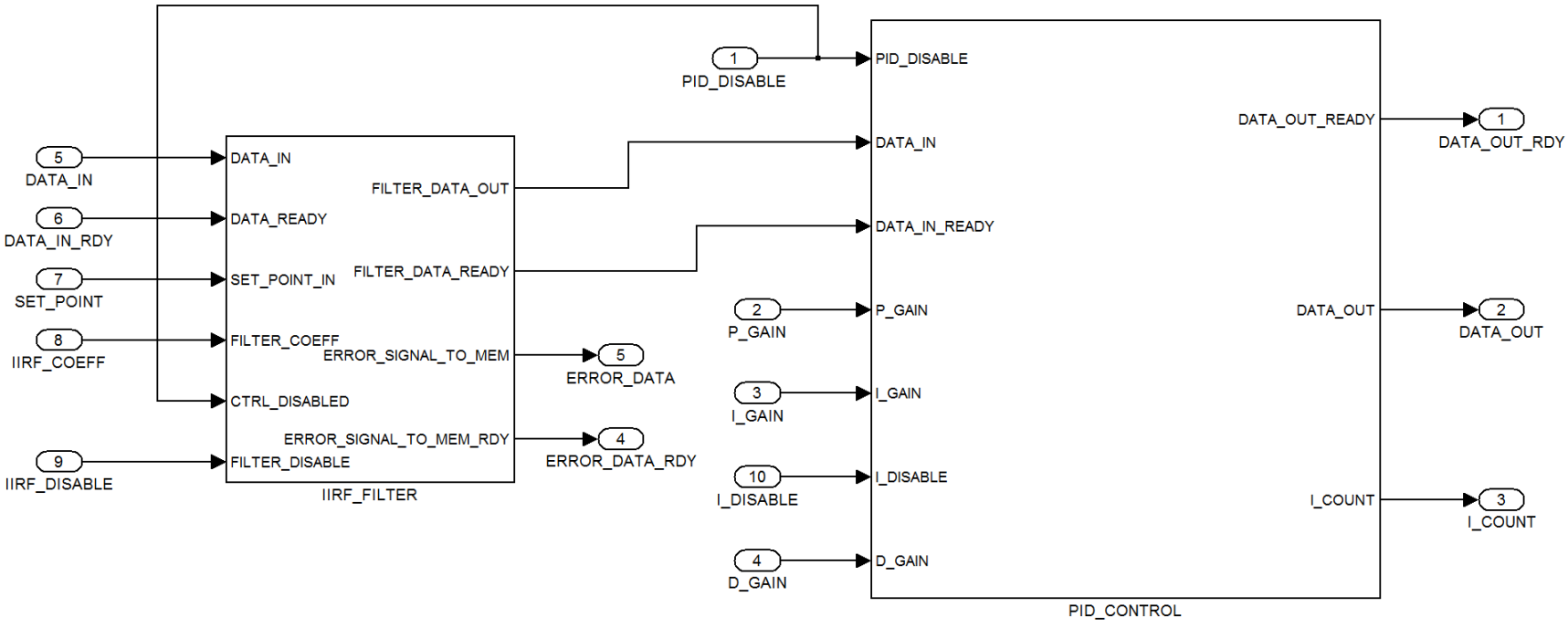
$$\begin{pmatrix} O_0 \\ O_1 \\ O_2 \\ \dots \\ O_7 \end{pmatrix} = \begin{pmatrix} C_{00} & C_{10} & \dots & C_{70} \\ C_{01} & C_{11} & \dots & C_{71} \\ C_{02} & C_{12} & \dots & C_{72} \\ \dots & \dots & \dots & \dots \\ C_{07} & C_{17} & \dots & C_{77} \end{pmatrix} \times \begin{pmatrix} I_0 & I_1 & \dots & I_7 \end{pmatrix}$$

- Allows the operator to mix all the inputs
- Weighing in the range from -2 to 2 for each input
- Outputs from the matrix enter the Simulink-generated 'Black Box'
- Useful for processing but also when signals change and access to cables is limited

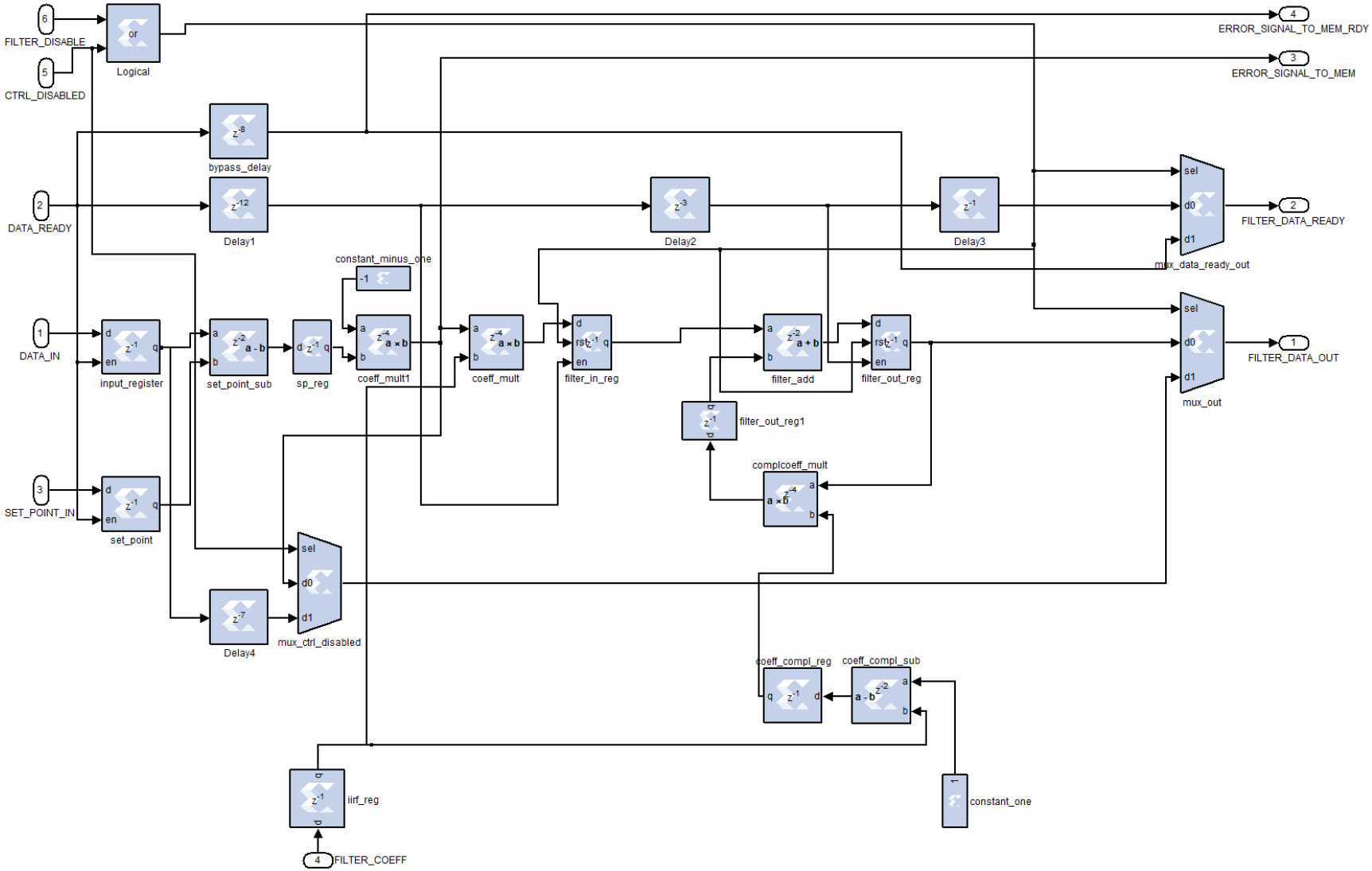
Controller highlights - Sysgen modules (top)



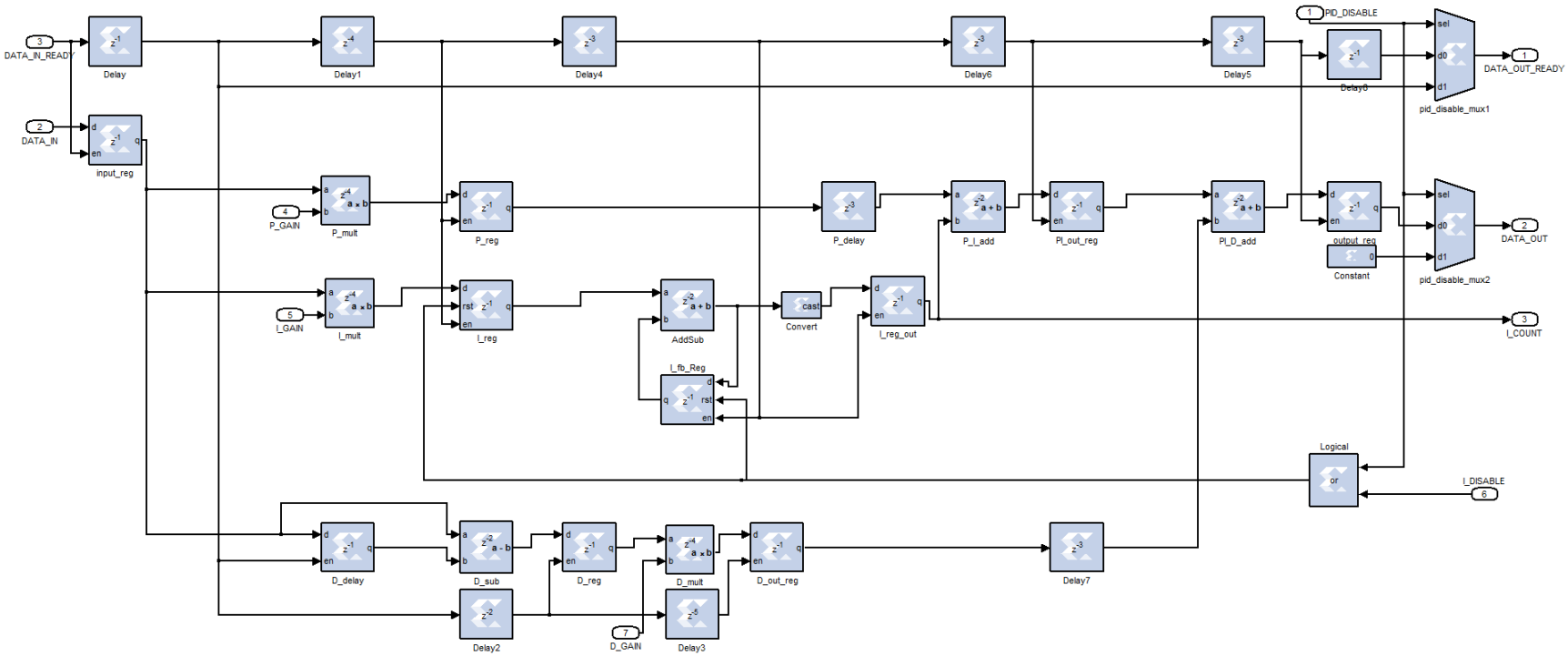
Controller highlights - Sysgen modules (IIRF + PID)



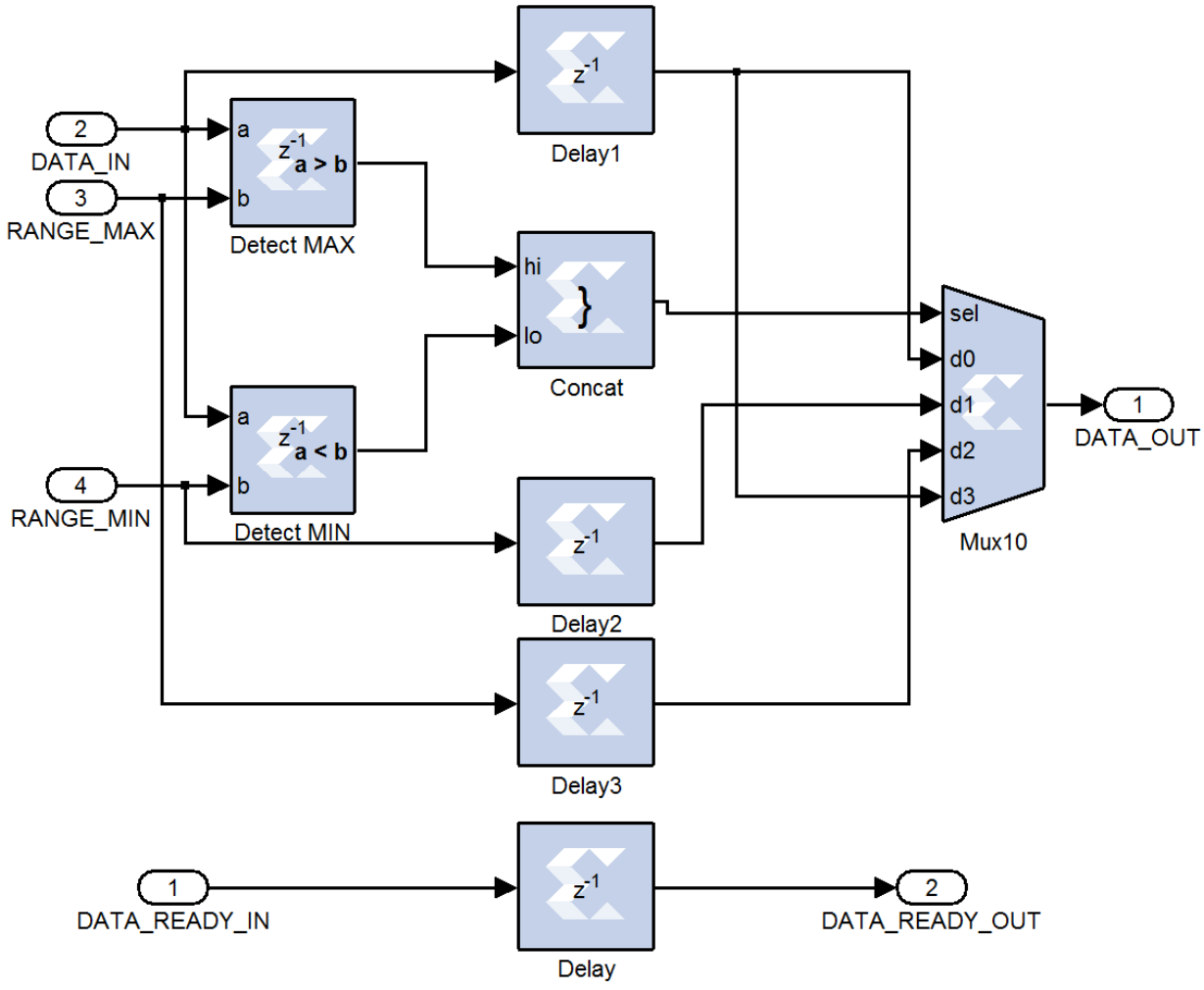
Controller highlights - Sysgen modules (IIRF)



Controller highlights - Sysgen modules (PID)

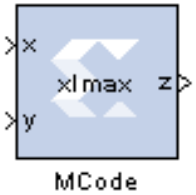


Controller highlights - Sysgen modules (DAC Clipping)

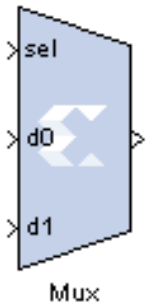


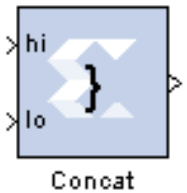
Sysgen pitfalls

- Can become resource hungry if advanced elements used



- Use of simple elements recommended





- Proper synchronization of signals necessary

Results and conclusions

- ☺ First functionality tests conducted with the MENLO laser in OCAS lab
  - The laser has been locked successfully
  - Unidentified glitches visible in the time domain
  - Timing jitter analysis was not yet performed
  - Initial very subjective impression - IMPROVED quality of laser lock

Results and conclusions, cont.

- ☹ FPGA resource utilization
  - (Matrix + Single PID) with Speed optimization:
    - Slices: 89%
    - BRAM and FIFO: 98%
    - Total memory used (kB): 85%
    - DSP Slices: 21 out of 48
  - (Preprocessing + Matrix + Single PID) don't fit in the FPGA
  - Limited number of DSP slices (48) is another problem

Outline

Next steps

- Get quantitative proof of the improvements introduced by this implementation
- Investigate the glitches and get rid of them
- Add functionality to the implementation
  - More preprocessing algorithms
  - System Identification capabilities
- Change the controller implementation after SI investigation
- Use external DDR to store samples
- Use PCIe DMA access

Thank you for your attention

Questions?

Comments?