Designing an RD53B Trigger Pattern Encoder for the YARR Readout System

Lucas Cendes

Agenda

- Background
  - Large Hadron Collider (LHC)
  - RD53B Readout Chip
  - YARR
- Design
  - Trigger Code Generator
  - Trigger Extender
- Testing
  - Testbench design
  - Simulation Results
  - YARR scans on real hardware

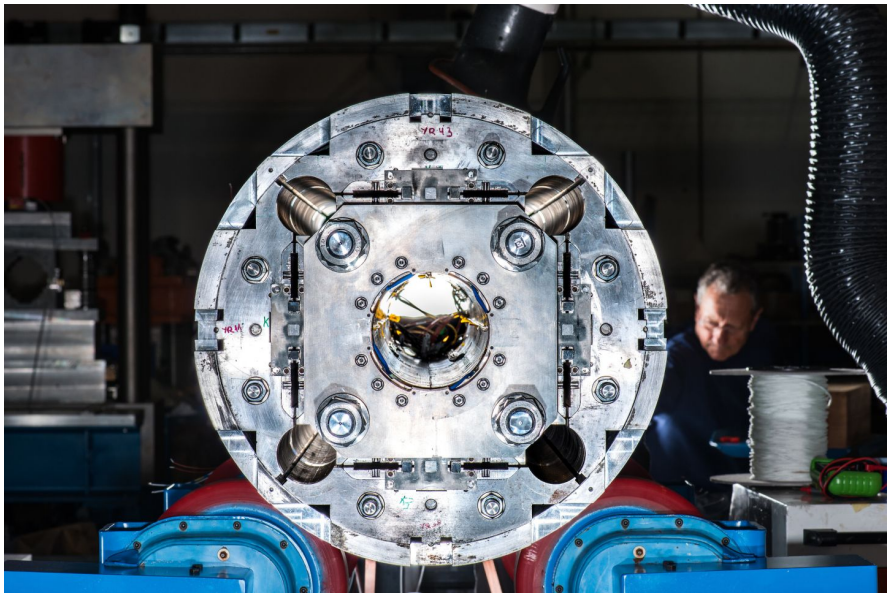
Background

Large Hadron Collider (LHC)

- Largest and most powerful particle accelerator in the world
- Collisions are produced by two particle beams travelling in opposite directions
    - Beams produced by several intense bunches of protons
    - Bunch crossing occurs when two beams cross
    - This happens at a rate of 40 MHz
- One of the main components of the LHC is the ATLAS detector
    - Designed to detect the smallest and most energetic particles
    - The Pixel Detector is one of the main components of the ATLAS detector

High-Luminosity Upgrade (HL-LHC)

- Intends to increase the luminosity by a factor of 10 times
  - This will result in a higher frequency of collisions
  - It will also allow the LHC experiments to observe rare physical processes
- The luminosity will be up to seven times greater than what the ATLAS detector was designed for
  - This will require new detectors to be developed and installed
  - New readout chips will have to be developed



RD53B Readout Chip

- The RD53B was designed to supersede the RD53A
  - The architecture of the RD53B was designed to be suitable for production
  - The chip also includes additional technology that facilitates its integration into the Pixel Detector
- Availability of the RD53B is currently limited
  - As a result of this, the ACME lab at UW developed an emulator for the chip
  - This will facilitate the process of testing and modifying the architecture of the chip
  - The two main components of the emulator are the TTC data processing block and the Command Processor

RD53B Command Protocol

- Commands are sent as a continuous stream of serial data
- Each command is a 16-bit frame composed of two 8-bit symbols
- Commands can span multiple frames
- The command frames are DC-balanced
    - This means that every command is composed of eight 0’s and eight 1’s
    - This allows for clock recovery and error detection

Control Commands

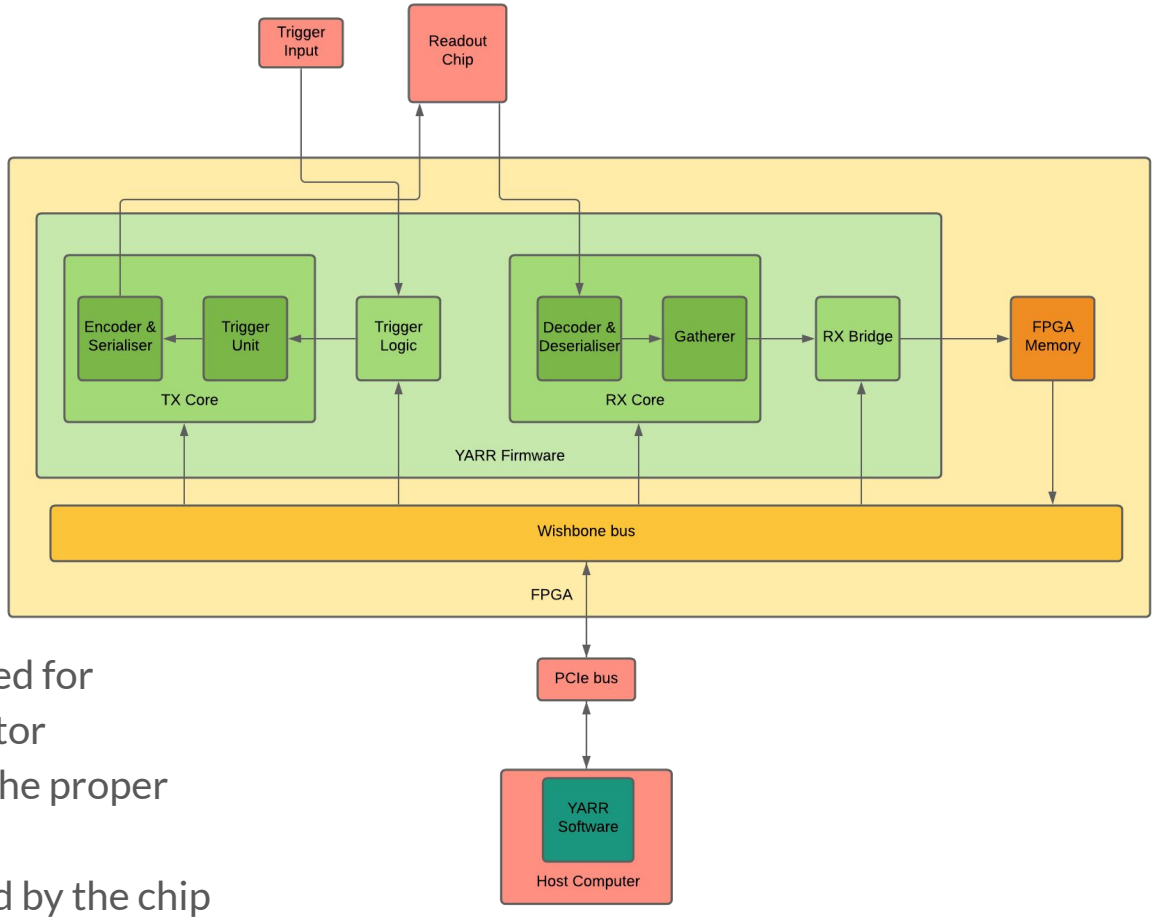
- PLL_Lock (Idle) Command
  - Alternating series of 1’s and 0’s
  - Simulates a clock signal
  - Used to ensure that the RD53B input circuitry is locked to the correct clock
  - Also used as an idle command
- Sync Command
  - Used to determine the proper frame boundaries
  - Must be sent at the start of operation
  - Cannot be produced by any combination of command words

Trigger Commands

- Used to tell the RD53B chip to sample data from a specific bunch crossing
- Composed of an 8-bit trigger encoding and and 8-bit tag
- Trigger encodings determine which bunch crossing should be samples
  - Each 8-bit symbol is mapped to a 4-bit trigger patterns
  - The presence of a trigger indicates that the data from the respective bunch crossing should be sampled
- Tags are used to identify the specific sequence of bunch crossing that correspond to a command
  - There are a total of 54 8-bit tags
  - Each tag is mapped to a 6-bit tag base

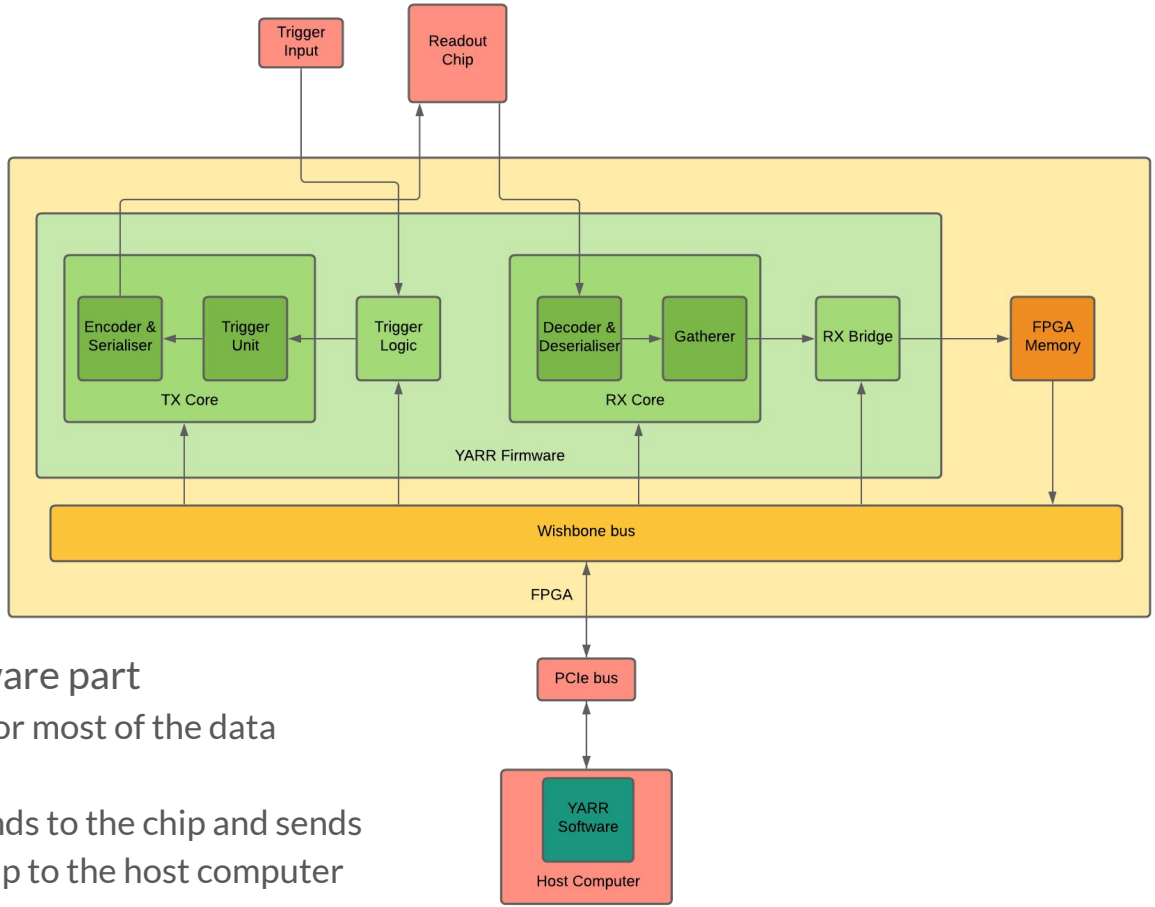
Trigger Pattern	Binary encoding	Hexadecimal Encoding
000T	0010_1011	0x2B
00T0	0010_1101	0x2D
00TT	0010_1110	0x2E
0T00	0011_0011	0x33
0T0T	0011_0101	0x35
0TT0	0011_0110	0x36
0TTT	0011_1001	0x39
T000	0011_1010	0x3A
T00T	0011_1100	0x3C
T0T0	0100_1011	0x4B
T0TT	0100_1101	0x4D
TT00	0100_1110	0x4E
TT0T	0101_0011	0x52
TTT0	0101_0101	0x55
TTTT	0101_0110	0x56

YARR



- Data acquisition system designed for readout chips in the pixel detector
- Converts trigger patterns into the proper command encoding
- Also analyses the data produced by the chip

YARR

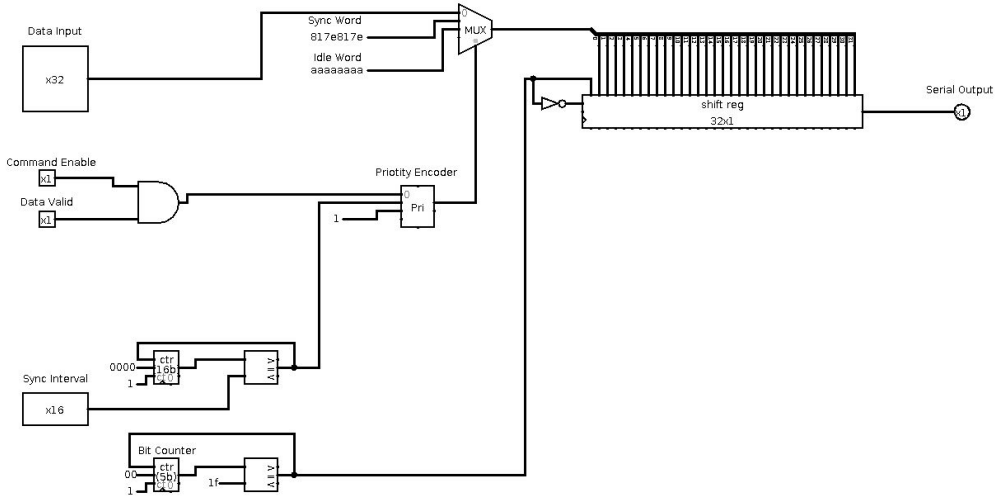


Composed of a software and a hardware part

- Software part is responsible for most of the data processing
- Hardware part sends commands to the chip and sends the data received from the chip to the host computer

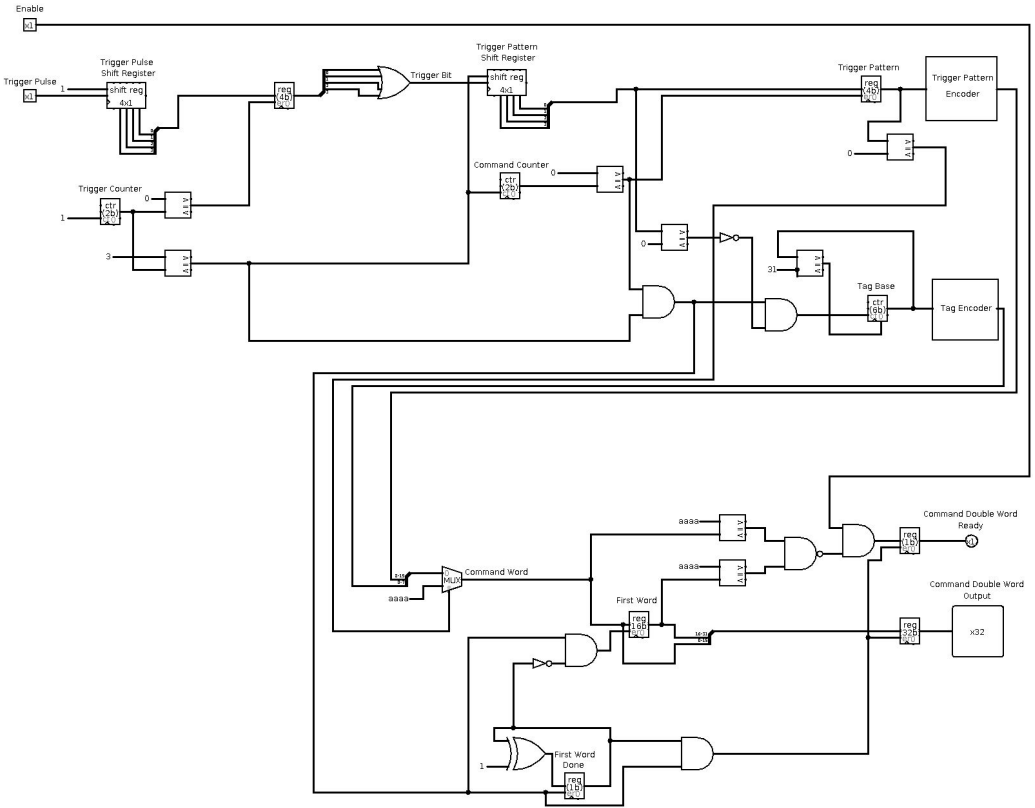
YARR Firmware TX Core

- The TX core is responsible for sending commands to the readout chip
- It is currently composed of a trigger unit and a TX channel
- It hosts a series of registers that can be configured through a Wishbone bus
- The TX core is currently being updated to support the command protocol of the RD53B chip
    - This requires the development of a trigger pattern encoder
    - The development of this module is the main focus of this project



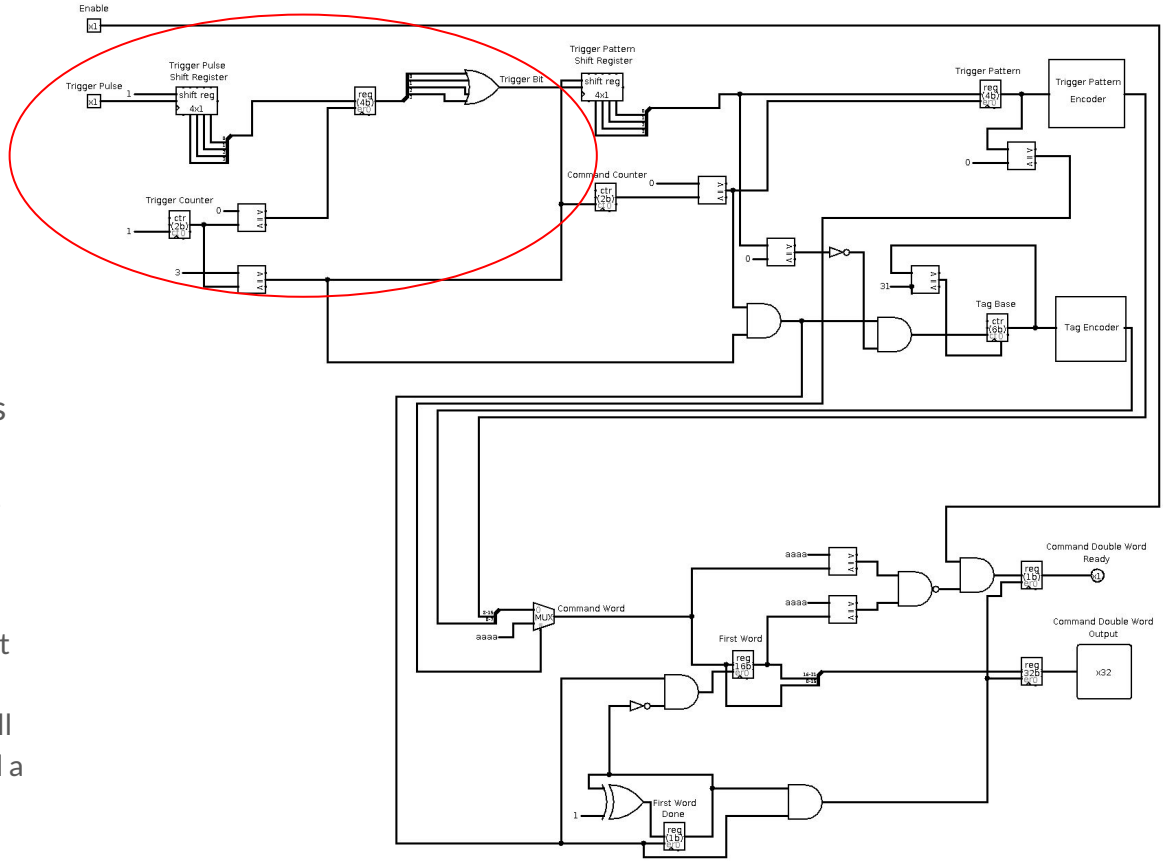
Design

Trigger Code Generator

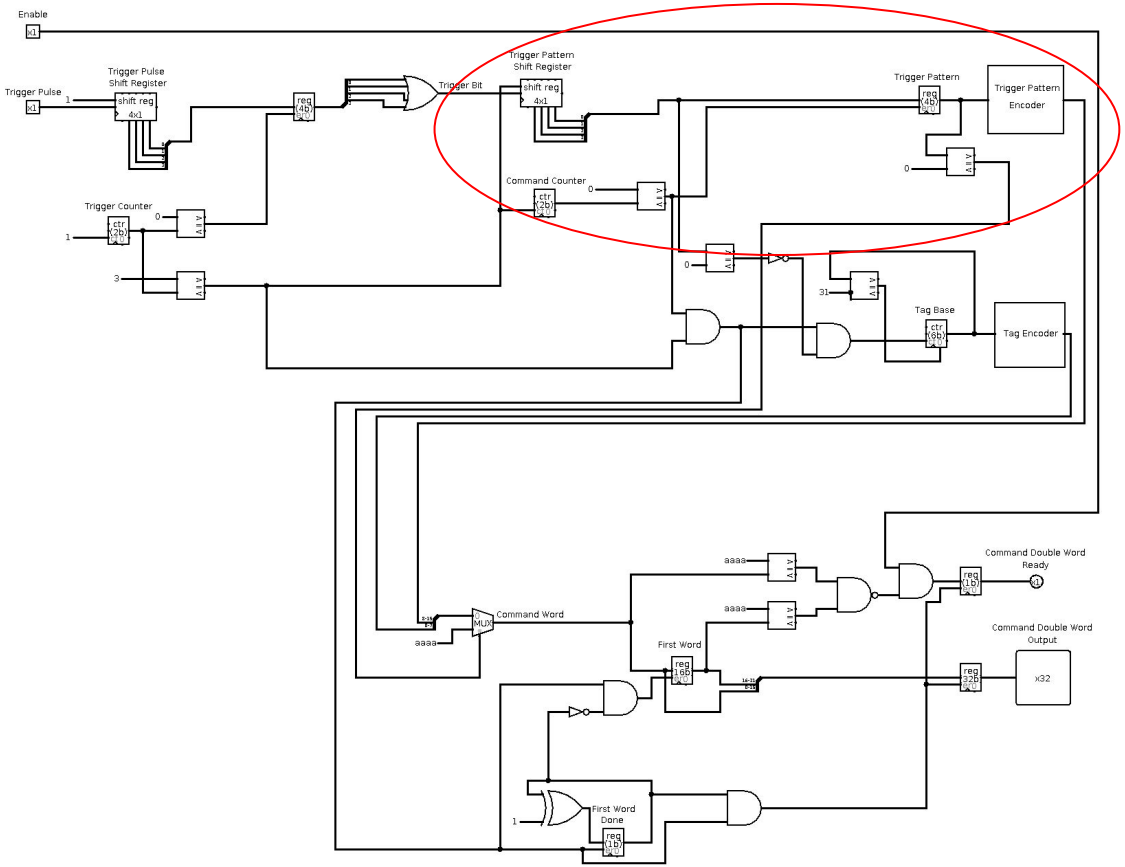


Trigger Pulse Processing

- Each Trigger Pulse lasts for 4 clock cycles
- A trigger is detected if the input port is set to 1 at any of the 4 clock cycles
- Trigger bits are generated through the usage of a 4-bit trigger pulse shift register and a 2-bit trigger counter
    - A bit is shifted into the shift register at each clock cycle
    - When the trigger counter back to 0, all bits in the shift register are OR'ed and a new trigger bit is produced

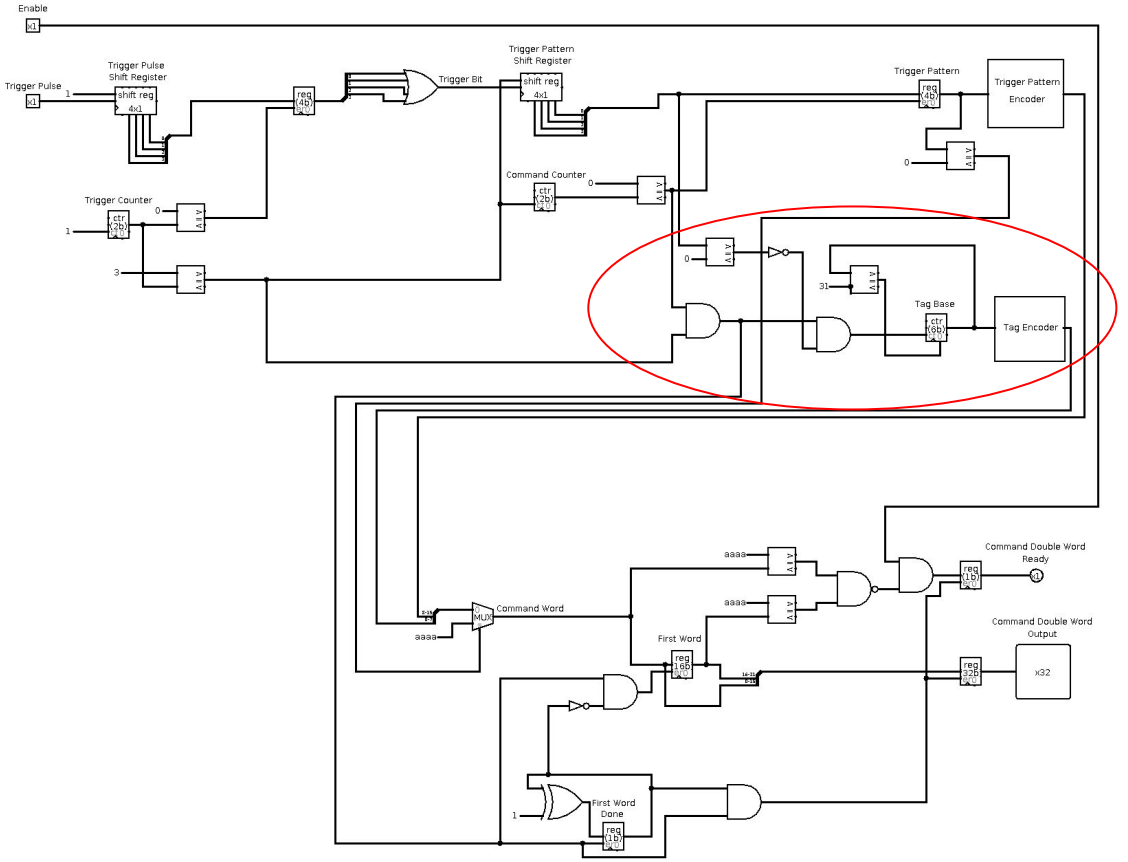


Trigger Command Generation



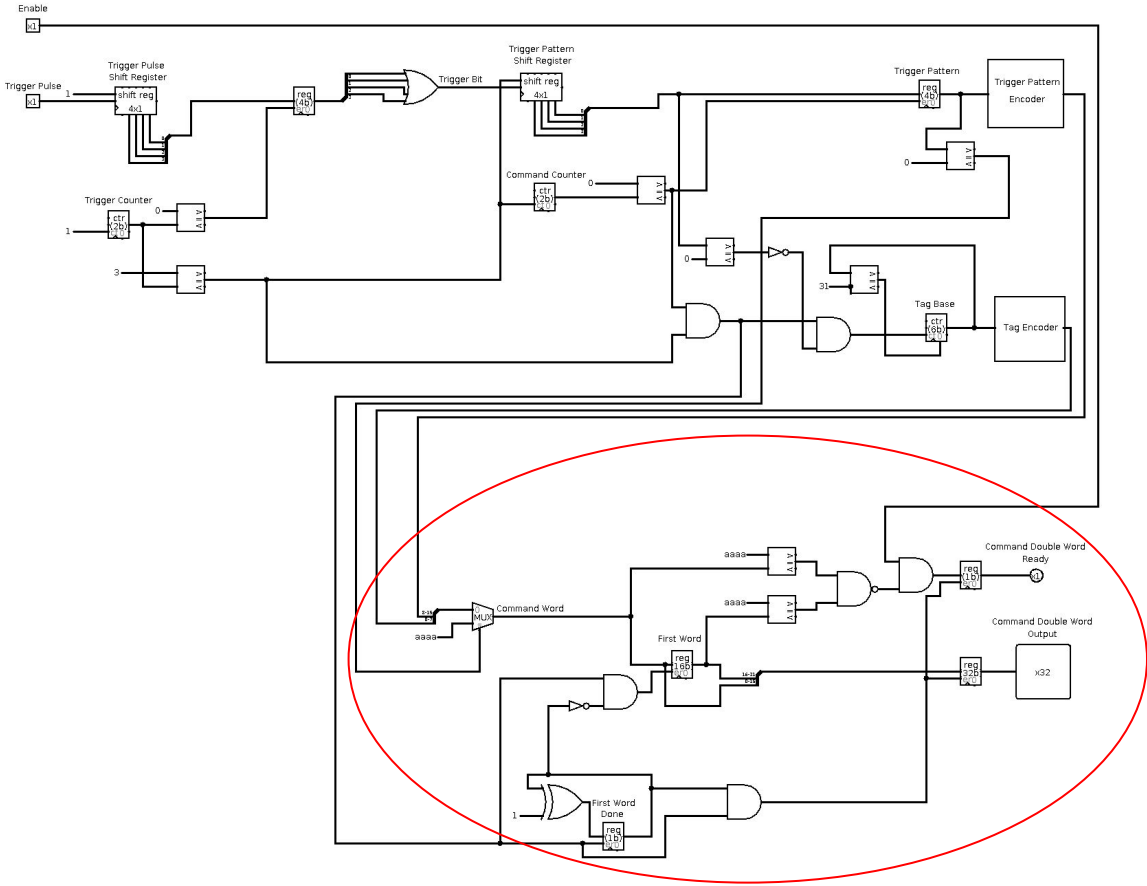
- The module keeps track of a 4-bit trigger pattern shift register and a 2-bit command counter
- When the trigger counter is set to 3, a new trigger bit is shifted into the trigger pattern shift register and the command counter is incremented
- When the command counter rolls back to 0, the trigger pattern is encoded into a 8-bit trigger encoding

Tag Generation



- A 6-bit counter keeps track of the tag base
- The counter rolls back to 0 whenever it is incremented past a value of 49
- Whenever a new trigger encoding is ready, the value of the counter tag is encoded into a 8-bit tag encoding
- The counter is incremented whenever a new command word is produced

Command Word Transmission



A new command word is produced every 16 clock cycles

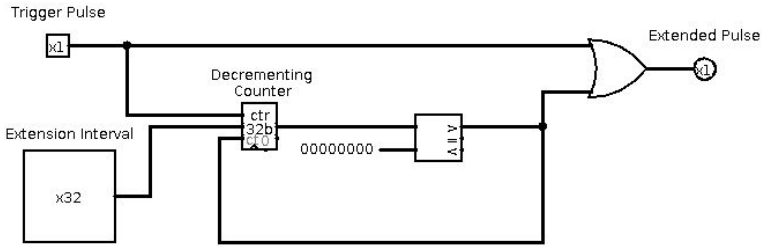
- If no triggers were detected in that interval, the command word is set to the idle command
- Otherwise, the command word is set to the concatenation of the 8-bit trigger encoding and the 8-bit tag encoding

The output is updated every 32 clock cycles

- The first word produced in that interval is stored at the first word register
- The next word is concatenated to the first word the resulting 32-bit value is outputted
- The first word done signal is asserted if the module is enabled and if there is at least one non-idle command word being outputted

Trigger Extender

- Allows the YARR software to extend the duration of a trigger pulse
- Module functions as a pass-through if the extension interval is set to 0
- Otherwise, any trigger pulses received by the trigger extender will be extender by the given number of clock cycles



Modifications to Existing Code

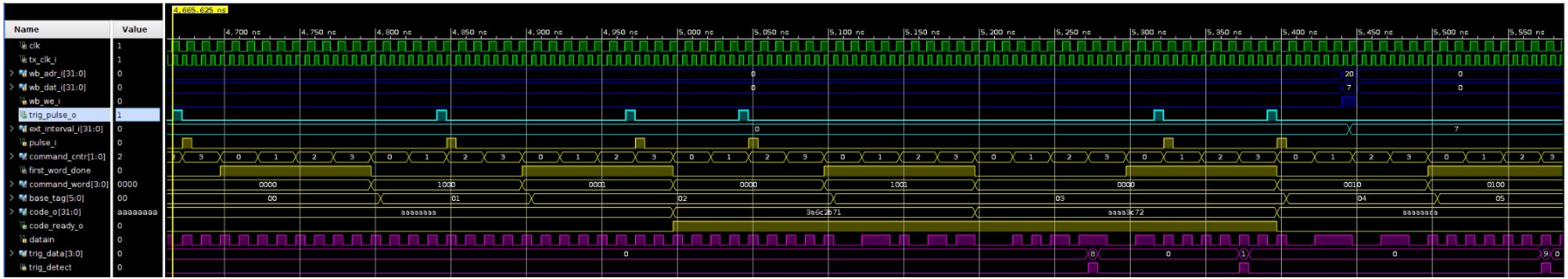
- Two new configurations registers were added to the TX core:
  - Trigger Extension Interval (Wishbone address 0x20)
  - Trigger Code Generator Enable (Wishbone address 0x21)
- The command words outputted by the trigger code generator were mapped to the highest priority of the priority encoder in the TX channel

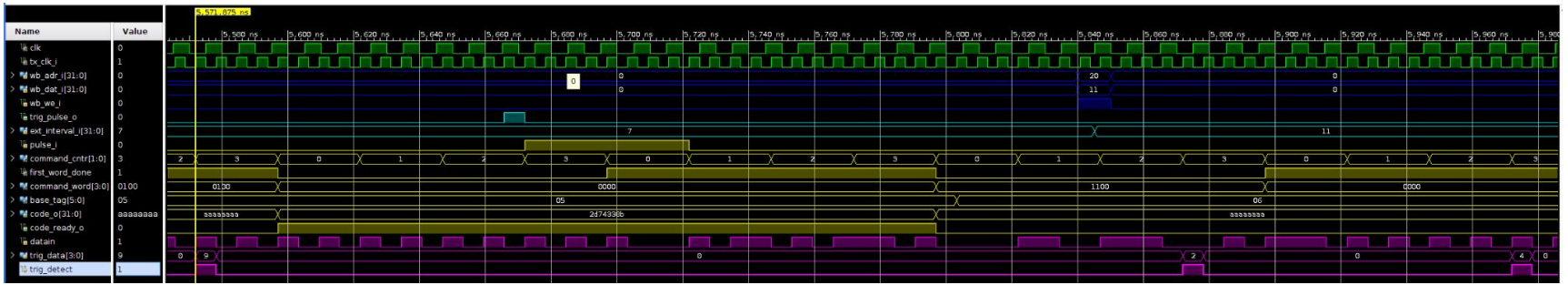
Testing

Simulation Testbench Design

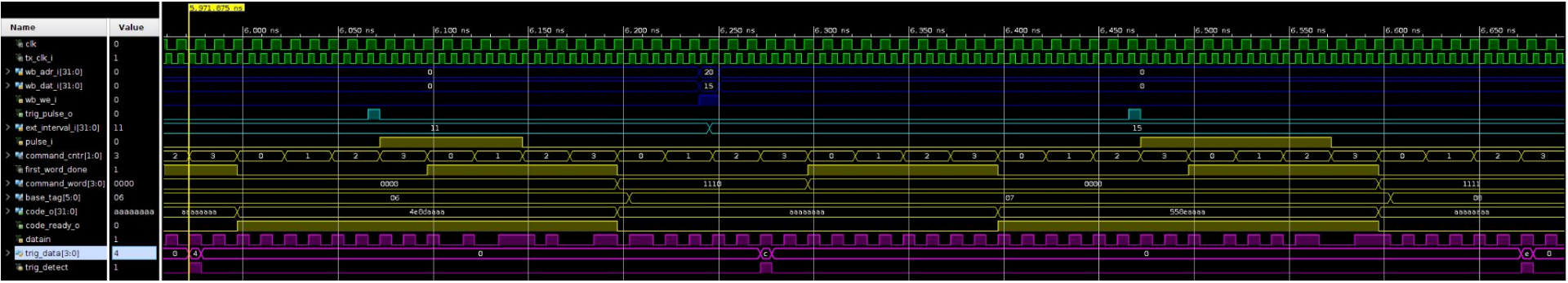
- The YARR firmware was connected to the RD53B emulator
- A counter was used to synchronize the trigger pulses with the code production intervals of the trigger code generator.
- The necessary TX core configuration registers were configured at the start of the simulation
- The trigger patterns 1000, 0001, 0000, 1001 were simulated
- Trigger extension intervals of 7, 11 and 15 were tested with the 1000 trigger pattern

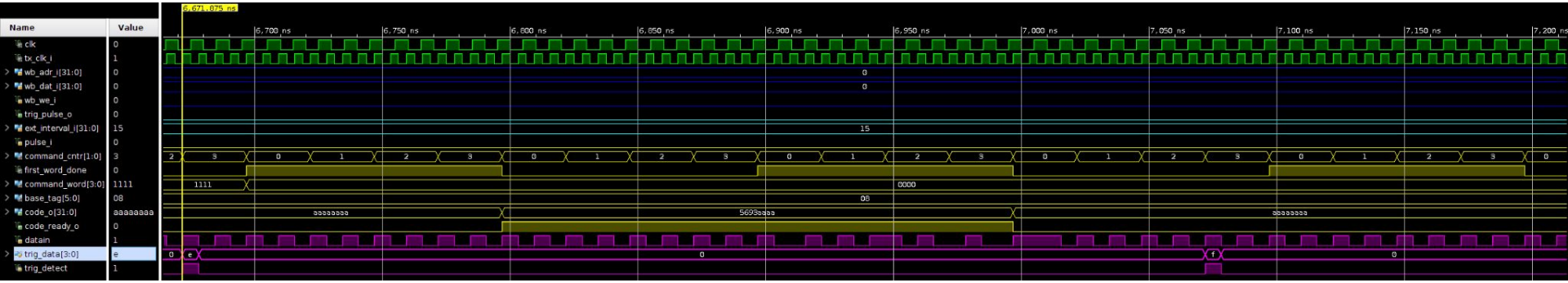
Simulation Results





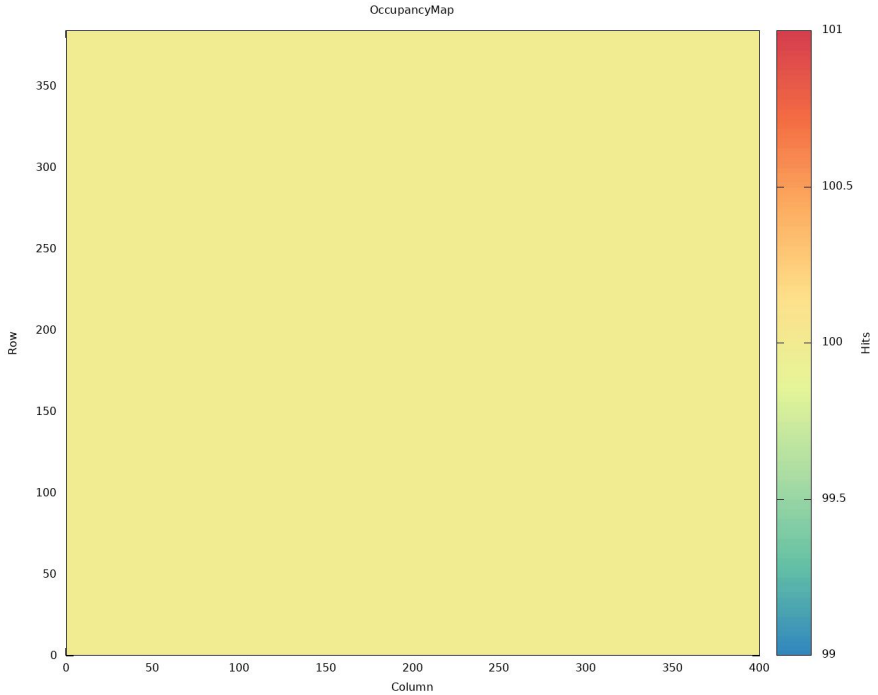
Simulation Results





YARR Scans in Real Hardware

- This test was used to check if the modifications done to the YARR do not interfere with the existing functionality
- The scans inject 100 digital pulses into each pixel of the RD53B chip
- The modified YARR firmware was loaded into a Trenz Electronic TEF1001 FPGA board
- The board was connected to the PCIe port of a computer
- The output port of the board was connected to the CMD/DATA port of a real RD53B chip
- The scans were then run from the computer connected to the Trenz Electronic TEF1001 board and a occupancy map was produced



References

- K. Anthony. “Preparing ATLAS for the future”. In: CERN (Dec. 20, 2018). url: https://atlas.cern/updates/news/preparing-ATLAS-for-future.
- F. Arteche Gonzalez et al. Extension of RD53. Tech. rep. Geneva: CERN, Sept. 2018. url: https://cds.cern.ch/record/2637453.
- CERN. High-Luminosity LHC. url: https://home.cern/science/accelerators/high-luminosity-lhc (visited on 05/14/2021)
- CERN. The ATLAS Detector. url: https://atlas.cern/discover/detector (visited on 05/14/2021).
- CERN. The ATLAS Experiment. url: https://atlas.cern/about (visited on 05/14/2021).
- CERN. The Large Hadron Collider. url: https://home.cern/science/accelerators/large-hadron-collider (visited on 05/14/2021).
- CERN. The RD53B Pixel Readout Chip Manual. May 27, 2020.
- T. Heim. “YARR - A PCIe based Readout Concept for Current and Future AT-
- LAS Pixel Modules”. In: Journal of Physics: Conference Series 898 (Oct. 2017), p. 032053. doi: 10 . 1088 / 1742 - 6596 / 898 / 3 / 032053. url: https://doi.org/10.1088/1742-6596/898/3/032053.
- T. Heim, A. Sautaux, and V. Baratham. Yarr Firmware. Aug. 31, 2020. url: https://github.com/Yarr (visited on 05/16/2021).
- F. Hugging. “The ATLAS pixel detector”. In: IEEE Transactions on Nuclear Science 53.3 (2006), pp. 1732–1736. doi: 10.1109/TNS.2006.871506.
- N. Mittal. “Development of an FPGA emulator for the RD53B chip”. MSc Thesis. University of Washington, Aug. 14, 2020. url: http://hdl.handle.net/1773/45785 (visited on 05/14/2021).
- A. Sautaux. Wishbone Express Core. Aug. 30, 2017. url: https://github.com/Yarr/Yarr-fw/blob/master/rtl/kintex7/wbexp-core/README.md (visited on 05/14/2021)
- B. Schmidt. “The High-Luminosity upgrade of the LHC: Physics and Technology Challenges for the Accelerator and the Experiments”. In: Journal of Physics: Conference Series 706 (Apr. 2016), p. 022002. doi: 10.1088/1742-6596/706/2/022002.
- url: https://doi.org/10.1088/1742-6596/706/2/022002.
- N. L. Whallon et al. “Upgrade of the YARR DAQ system for the ATLAS Phase-II pixel detector readout chip”. In: PoS TWEPP-17 (2018), 076. 5 p. doi: 10.22323/1.313.0076. url: https://cds.cern.ch/record/2312402.