Санкт-Петербургский государственный университет

Кафедра системного программирования

Создание IP-ядра полукогерентного обнаружителя частотно модулированных сигналов с непрерывной фазой

Никита Михайлович Фаст, группа 24.М41-мм

Научный руководитель: к.ф.-м.н. Д.В. Луцив, доцент кафедры системного программирования

Консультант: А.С. Кривоногов, начальник отдела программирования АО «Концерн ГРАНИТ»

Санкт-Петербург
2025

Введение

- При беспроводной передаче данных к полезному сигналу добавляются шумы и помехи
- При использовании слотовой передачи данных, полезный сигнал присутствует в эфире лишь в определенные моменты времени
- Для проверки наличия сигнала в радиоэфире используется обнаружитель
- В рамках работ по разработке канала беспроводной связи, проводимых АО Концерн «ГРАНИТ», требуется создать соответствующий обнаружитель на ПЛИС, т.к. имеющаяся реализация на CPU чрезмерно ресурсоемка

Постановка задачи

Целью работы является создание IP-ядра полукогерентного обнаружителя частотно модулированных сигналов с непрерывной фазой. Для достижения данной цели были поставлены следующие задачи.

1. Сформулировать критерии выбора алгоритма обнаружения
2. Осуществить выбор алгоритма обнаружения путем выполнения обзора существующих алгоритмов обнаружения и их сравнения по выбранным критериям
3. Спроектировать архитектуру обнаружителя
4. Реализовать обнаружитель на одном из языков HDL
5. Разработать тестовое окружение под процессор ARM и с его помощью выполнить тестирование разработанного IP-ядра

Существующие решения

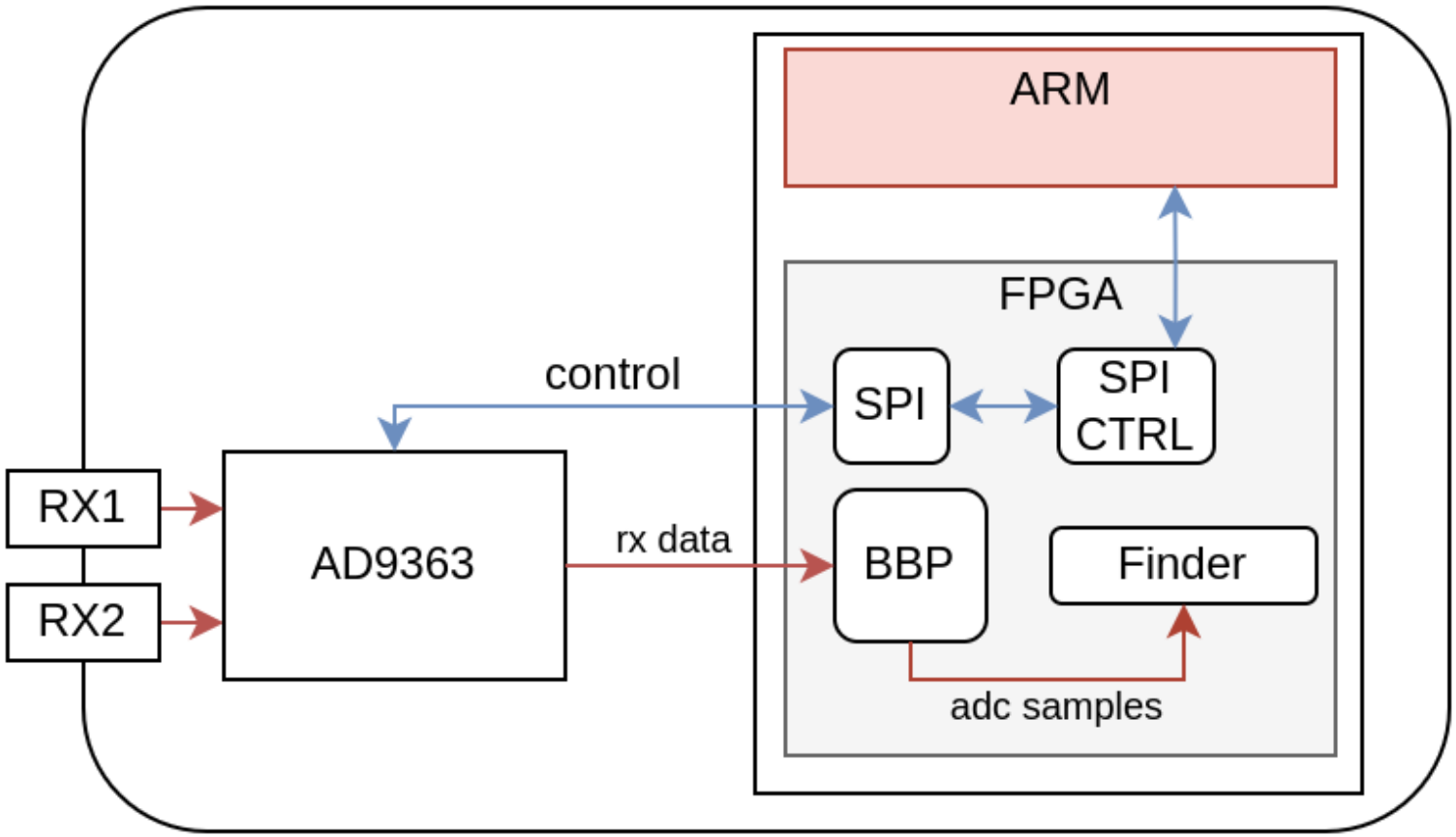
- На основе IP-ядер могут быть созданы микросхемы
- Поэтому IP-ядра могут обладать высокой коммерческой ценностью
- Из-за этого число открытых IP-ядер крайне ограниченно
- Готовое IP-ядро может работать лишь на конкретной ПЛИС или семействе ПЛИС
- Для платформы AMD Zynq™ 7000 SoC, готового решения найдено не было

Используемые технологии

- В силу соотношения цена/качество платформой был выбран PlutoSDR+, главными компонентами которого являются:
  - AMD Zynq™ 7000 SoC
  - Приемопередатчик AD9363
- Поскольку производителем ПЛИС является Xilinx (AMD), то используются IDE Vivado и Vitis
- Для разработки под ПЛИС выбран VHDL, т.к:
  - Использует строгую систему типов
  - Обладает детерминированностью симуляции
  - Является основным языком описания аппаратуры в организации

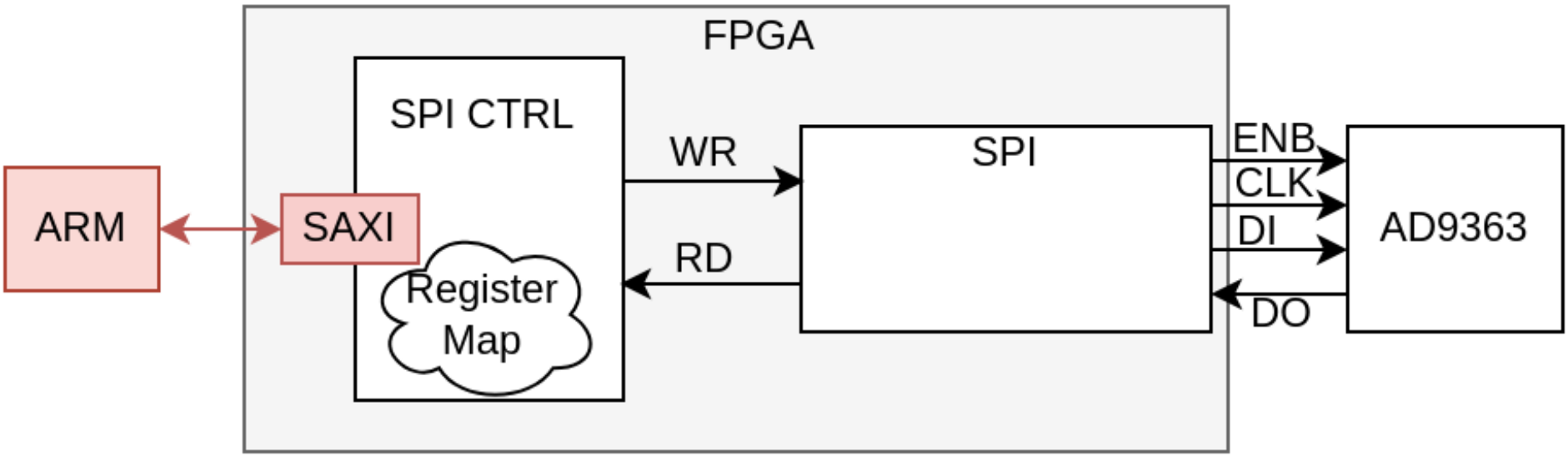
Архитектура системы

- Для своей работы обнаружителю требуются отсчеты АЦП приемника. Для их получения необходимо:
  - Используя протокол SPI, настроить микросхему приемника
  - Принять данные АЦП, доставить их в ПЛИС и отформатировать



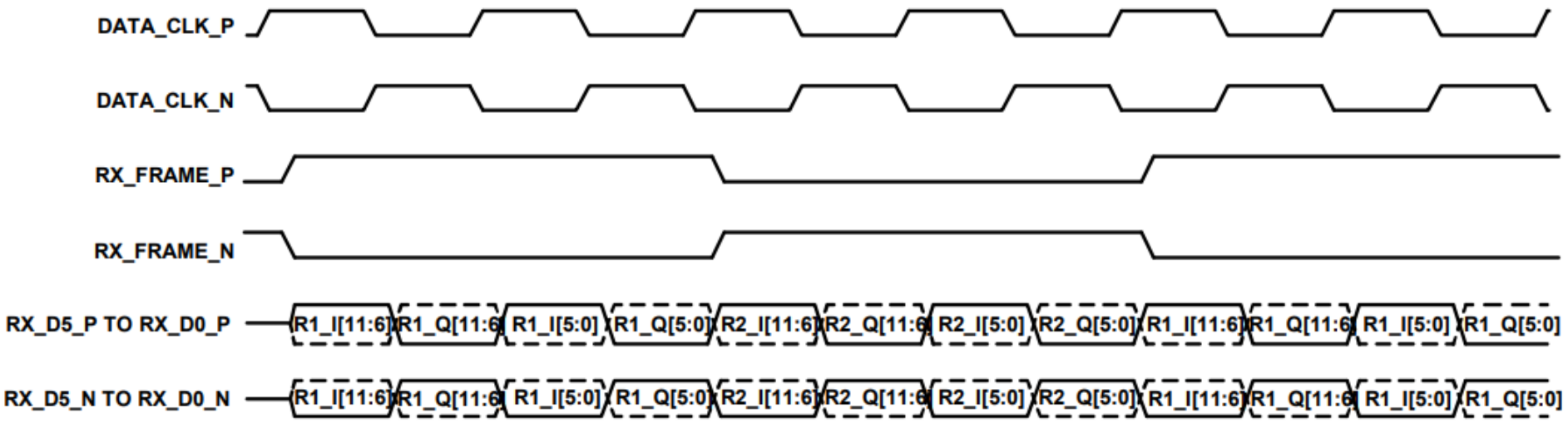
Реализация SPI

- Сущность SPI формирует набор сигналов, в соответствии с SPI-протоколом AD9363
- Процессор, выполняя чтение и запись регистров сущности SPI CTRL через AXI memory mapped порт, управляет сущностью SPI



Реализация ВВР

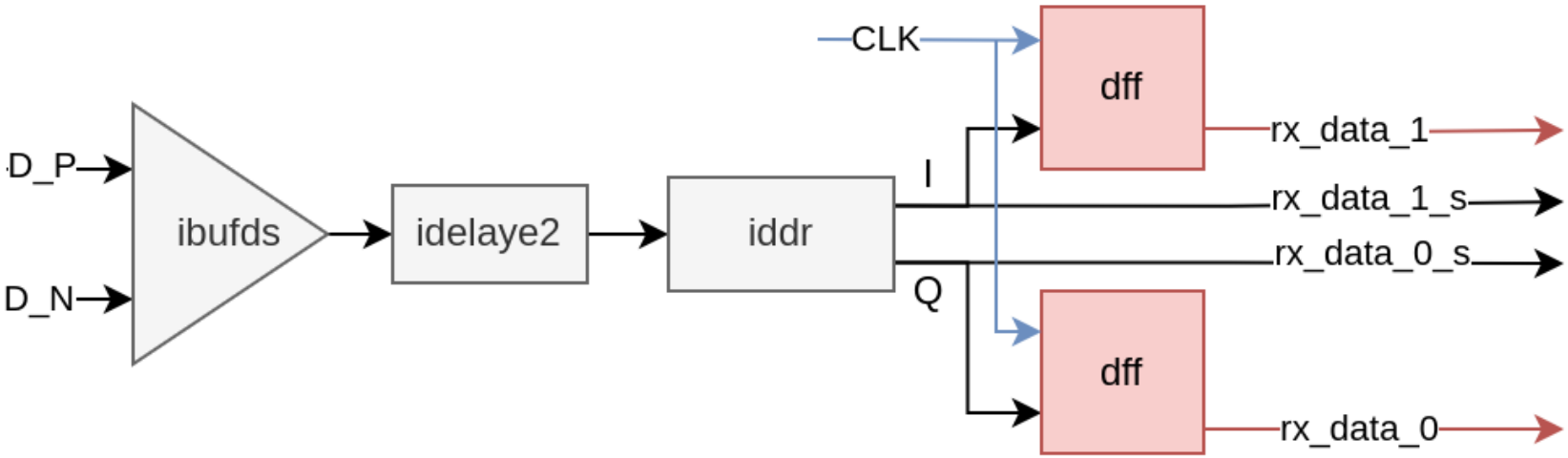
- Приемник осуществляет передачу данных в формате edge-aligned source synchronous ddr
- Отсчет сигнала это комплексное 24-битное число
- Необходимо за 2 такта обработать 4 порции данных
- Предельная частота тактового сигнала составляет 245.76 Mhz
- К реализации предъявляются строгие временные ограничения!



Реализация ВВР

- Сигналы data и frame проходят ibufds, idelaye2 и iddr
- Выходы iddr задерживаются на такт
- Данные сохраняются в 48-битный регистр

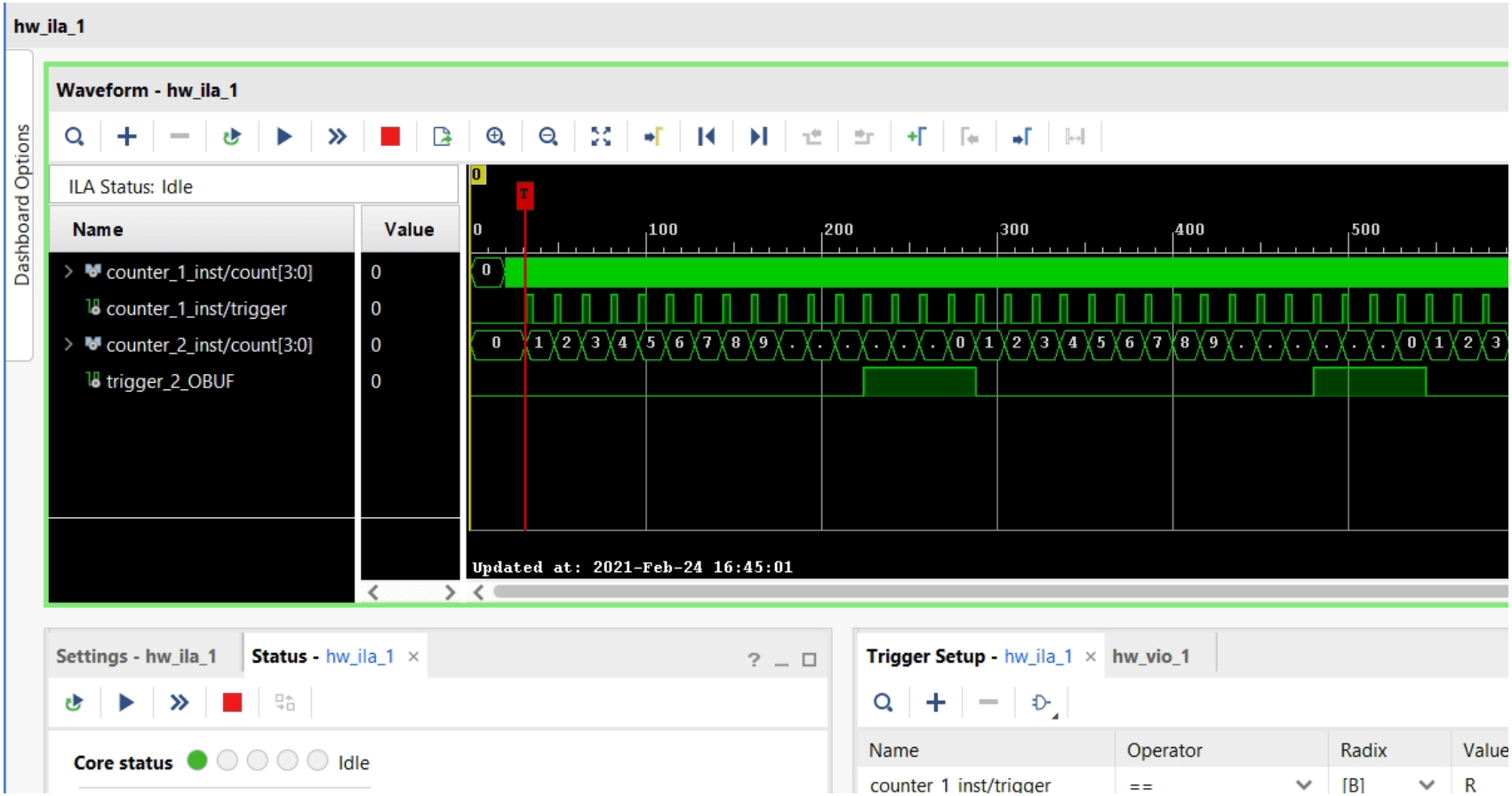
```
case rx_frame_s & rx_frame is
  when "1111" =>
    adc_valid <= 1'b0;
    adc_data[23:12] <= rx_data_1 & rx_data_1_s;
    adc_data[11: 0] <= rx_data_0 & rx_data_0_s;
  when "0000" =>
    adc_valid <= 1'b1;
    adc_data[47:36] <= rx_data_1 & rx_data_1_s;
    adc_data[35:24] <= rx_data_0 & rx_data_0_s;
```



Отладка и тестирование

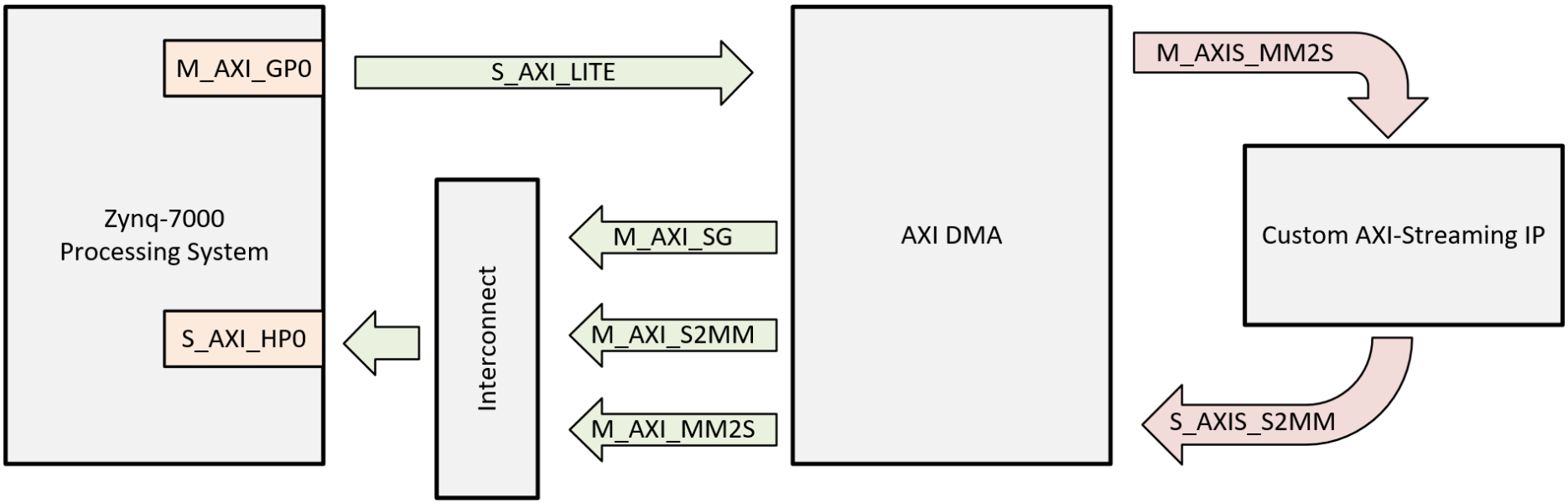
Штатные средства:

- Симуляция
- ILA



Отладка и тестирование

- Для более продвинутой отладки можно задействовать процессор
- В таком случае необходимы средства обмена данными между ПЛИС и ARM
- Для этой цели можно использовать IP-ядро AXI DMA и AXI memory mapped порты



Отладка и тестирование

- При отладке обнаружителя понадобится построение графиков корреляции и спектра сигналов
- Сделать такое на SDR трудно т.к. xilinx использует собственную урезанную библиотеку C/C++
- И очень легко на ПК с использованием Python, numpy и matplotlib
- Для этого нужно передать данные с PlutoSDR+ на HOST-ПК
- Можно использовать UART для отправки данных с ARM на HOST-ПК,
- Host-ПК выполняет чтение данных из COM-порта
- Недостатком является низкая скорость UART
- В дальнейшем интересен переход на Ethernet

Результаты

В ходе работы за осенний семестр были достигнуты следующие результаты.

1. Создана инфраструктура для разработки и отладки IP-ядра обнаружителя
   - Разработаны VHDL-драйвера для приемопередатчика AD9363, позволяющие подать на вход обнаружителя целевой сигнал в оцифрованном виде
   - Реализованы средства обмена данными между ПЛИС, процессорной системой и HOST-ПК, позволяющие осуществлять промежуточную отладку разрабатываемого IP-ядра