Digital Design of a HVCMOS pixel sensor prototype in 55nm for CEPC

Ruoshi Dong[1,2], Ivan Peric[2], Hui Zhang[2], Yang Zhou[1], Hongbo Zhu[3], Jianchun Wang[1], Yiming Li[1]

Keywords: HVCMOS, Pixel sensor, Readout, UVM

[1] The Institute of High Energy Physics of the Chinese Academy of Sciences (IHEP-CAS)
[2] Karlsruhe Institute of Technology (KIT)
[3] Zhejiang University (ZJU)

1. Introduction

Monolithic Active Pixel Sensor (MAPS) in high-voltage CMOS (HVCMOS) technology is one of the most advanced detectors for high-energy particle detection. It is a high performance and cost effective choice for particle tracking in CEPC. In order to make research on new circuit structures and test their performance on a more advanced process node, an HVCMOS pixel sensor prototype is designed by a collaborative team from KIT and IHEP for Multi Project Wafer (MPW) running using the 55 nm process.

In this analog-digital mixed prototype, a 26×26 pixel matrix and related digital readout circuits are integrated. The digital module receives hit pulses from the pixel matrix and it is responsible for the timestamp measurements and data uploading. A readout structure based on quad pixel units together with joint transmission of address and hit pulse is developed. Additionally, a more standardized design process using Universal Verification Methodology (UVM) is utilized within this design. This poster provides an introduction to the digital design of this pixel sensor prototype.

2. Design of digital module

The digital module of the CEPC HVCMOS pixel is responsible for receiving hit pulses from the front-end pixel array via the address bus and performing time-to-digital conversion. The left-most two columns of pixels are dedicated to digital readout, while other pixels produce analog hit outputs. As a result, multi-readout strategies are aimed to be verified in this prototype. In this 26×2 pixels on digital readout, a 2×2 pixel array is divided as a quad pixel unit (QPU), resulting in a total of 13 QPUs. Each QPU is assigned a unique address, numbered from 1 to 13, with address 0 reserved to indicate no hit on the pixels. Pixels within the same corner direction across all QPUs share a common address bus, resulting in a total of four address buses as inputs to the digital module. When a pixel is hit, it drives out its 4-bit address on its address bus and this achieves a mixed transfer of hit pulse and address.

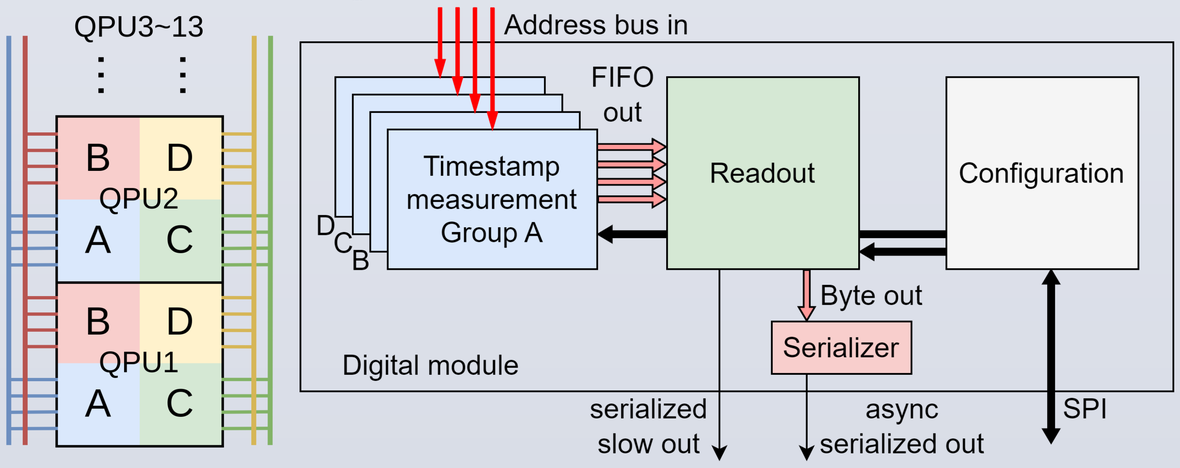

Fig. 1. QPU and address bus (left), and diagram of digital module (right)

The digital logic comprises four timestamp measurement modules, a data readout state machine, an asynchronous serializer, and a configuration block. An internal counter operates under the timestamp clock and produces timestamp values in gray code. The hit pulse from the address bus is operated with an OR logic in the timestamp measurement state machine to capture the leading and trailing edges. As the address bus is wired-OR driven, it could happen that multiple pixels generate overlapping hit pulses simultaneously. To address this, two check position parameters are set for a double check on the hit address, issuing an error bit in case of a mismatch. Additionally, to control the data rate and shield the noise spurs, a maximum detect length parameter is set. When a hit pulse is detected, the state machine locks for a specified clock period by this length, even if a trailing edge has already been detected. In the event of an exceptionally long hit pulse, potentially due to overlapping on the address bus, the state machine will terminate the current detection after reaching this maximum length. The measurement results are pushed into four parallel FIFOs and sent to the readout module. All of these parameters can be configured via the SPI configuration module.

The readout module performs polling check among the four FIFOs and generates a 48-bit data package for each hit. When a FIFO is enabled for reading, the readout state machine locks its number, captures the data package and waits for the byte shifter from the output interface to load the data package. Once this is complete, the readout state machine returns to the polling check state for the next data package.

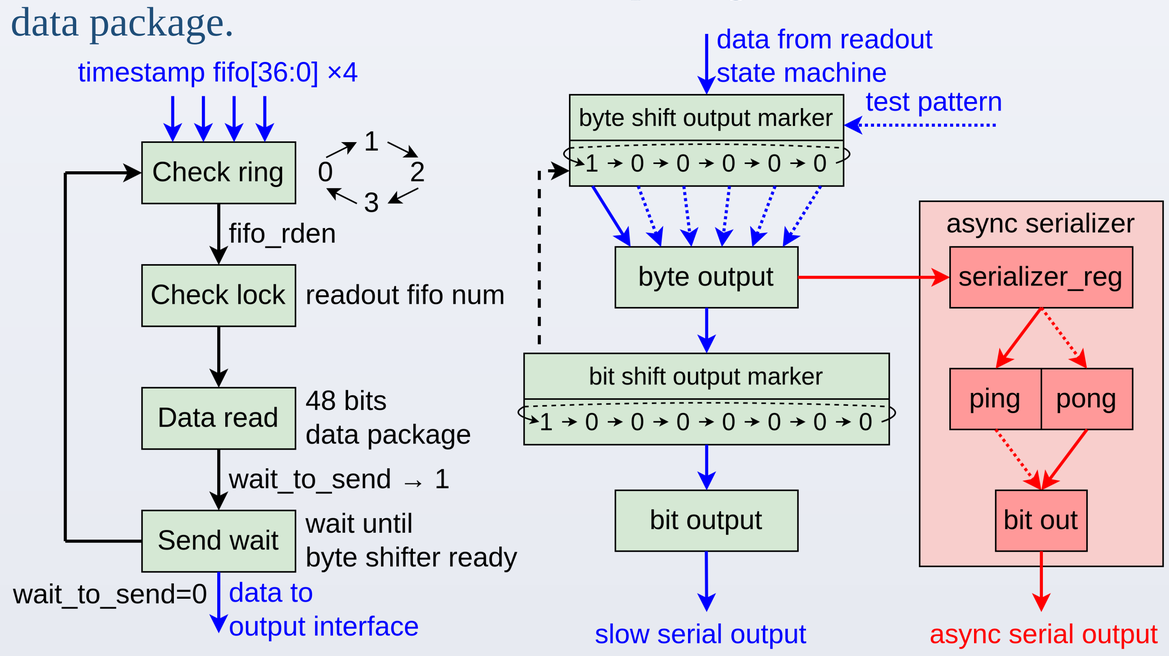

Fig. 2. Diagram of readout module and data interface

Two mutually exclusive readout interfaces are implemented for readout logic test, which can be selected active by configuration. In the slow serial output interface, the data package is transmitted on a single bit under the readout state machine's clock domain. In the asynchronous readout interface, the data package is sent on byte set to the serializer. The serializer locks the byte data with a ping-pong register, ensuring a sufficient setup time when crossing the clock domain. With this structure, the data can be readout on bit under an asynchronous clock whose frequency is eight times faster than that of the state machine.

3. Functional Verification by UVM

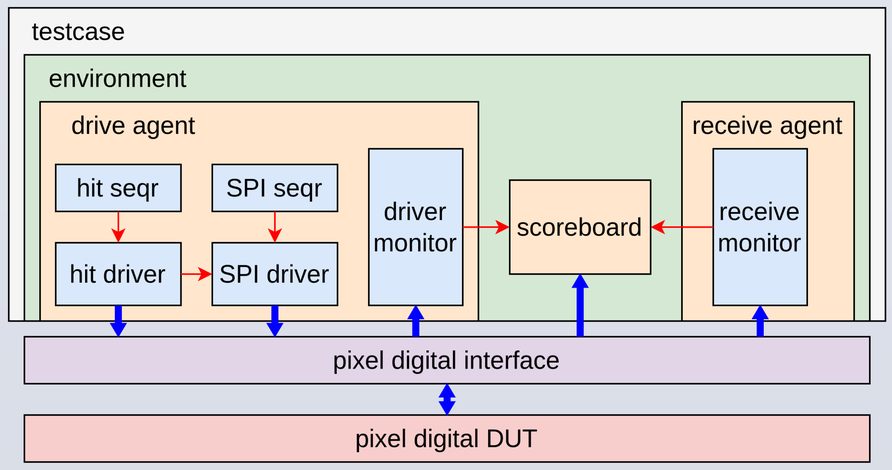

Fig. 3. UVM verification platform

For functional verification of the digital module, a UVM platform is established, as shown in Fig. 3. This testbench mainly focuses on two functions: configuration and readout. Two sequence classes are defined for generating SPI transactions and hit transactions separately. Correspondingly, two drivers control the start of these sequencers. The SPI driver starts the SPI sequence, randomizing a certain number of transactions and drives them to the DUT interface as random data/address configuration operations. It can also control the generation of abnormal short/long configuration operations to verify that the configuration module doesn't crash during incorrect operations.

The hit driver starts the hit sequence to generate a series of hit pulses on the address bus. This includes random, super long, spurs and overlapped pulses under different test cases when simulating various hit conditions. It also declares parameter settings to the SPI driver via a put-port. The driver monitor collects the hit pulse information and calculates expected output data packages, and in parallel, the receive monitor deserializes the output of the logic module into received data packages. Transactions from both monitors are sent to the scoreboard via tlm-analysis-fifo for matching and the scoreboard gives out the final verification result.

Author e-mail: dongrs@ihep.ac.cn