Documentation for digital pattern generator-based imaging controller

Ryan Thomas

March 16, 2020

1 Introduction

This document describes the design and operation of the digital pattern generator (DPG)-based imaging controller used in the Kjærgaard lab. While both this controller and its predecessor are implemented using a field-programmable gate array (FPGA), namely the Xlinx Spartan 3AN development board, the DPG imaging controller has a much simpler FPGA architecture and is more easily customizable than the previous, bespoke version.

Whereas the previous imaging controller had many different modules, each of which controlled a different aspect of the timing sequence, the DPG solution has only two modules: the DPG itself and a module to generate triggers for the FlexDDS. The reason that these are separate is that we need millions of triggers for the FlexDDS at a fixed rate and duty cycle, and using a DPG to generate these would require an enormous amount of memory. It is easier to use a dedicated module for this purpose.

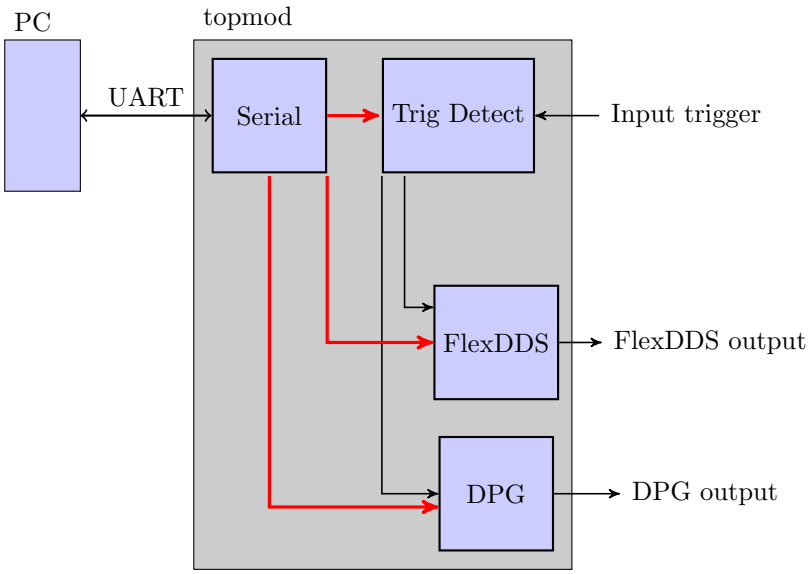

Figure 1: Diagram of the DPG-based imaging controller. A PC transmits parameters to the FlexDDS trigger generator and a digital pattern to the DPG. The sequence starts either when an input trigger is received or when a software trigger is issued by the host PC over serial. Thick red lines indicate the transmission of information to and from the serial controller.

2 Digital Pattern Generator (DPG)

A DPG is simple to understand – it is a device that stores a list of digital output patterns and the times at which they should be enacted. In the case of the current DPG, these are stored as a series of 40-bit (5 byte) instructions, where the most-significant byte (MSB) is the type of instruction and bytes 0 to 3 (for a total of 32 bits) are the *data*. There are currently three types of instructions:

1. MSB = 0x00 This instruction tells the DPG to wait for a time equal to *data* sample clock cycles.

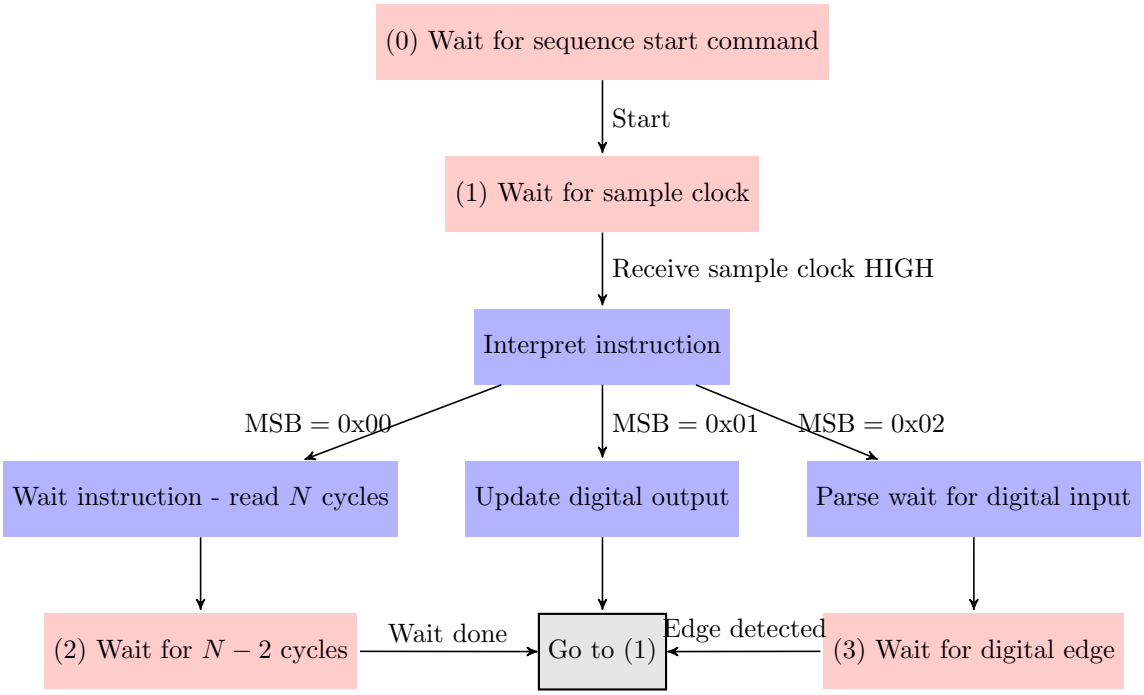

Figure 2: Diagram of the finite-state machine (FSM) at the heart of the DPG. Red boxes indicate states of the state machine; blue boxes indicate conditional statements. New instructions are requested when a "start" command is received and also whenever a sample clock HIGH is detected. When all instructions are read, the FSM returns to state 0, as well as when a sequence "stop" command is received.

2. MSB = 0x01 This instruction tells the DPG to output the pattern equal to *data*.

3. MSB = 0x02 This instruction tells the DPG to wait for a digital input event before continuing to the next instruction. Bits 0 to 3 of *data*, interpreted as an unsigned integer, indicate the digital input channel to look at, and bits 8 to 9 indicate the type of edge to look for.