

CprE 488 – Embedded Systems Design

HW-1: Pulse Position Modulation

Assigned: Friday of Week 2

Due: Friday of Week 3

Points: 10

1) Relearning VHDL. Carefully read through Sections 7-7.1 of the *Free Range VHDL* book (available here: http://class.ece.iastate.edu/cpre488/resources/free_range_vhdl.pdf).

- i. Describe some aspect you thought was unclear, and post a question to the “In-Class Discussion” topic on Blackboard. If everything was perfectly clear (great!), instead describe what is happening line-by-line in Listing 7.1. Be specific.
- ii. In Section 7.4, complete exercises 1 and 2.

2) Pulse Position Modulation. In CprE 288 you were introduced to the concept of Pulse Width Modulation (PWM). In this lab you will be working with Pulse Position Modulation (PPM). Visit the following link for a brief introduction to PPM: <http://www.endurance-rc.com/ppmtut.php>.

- i. Describe the similarities and differences between PWM and PPM.
- ii. Draw a 6 channel PPM frame having a 20ms period, with Channel 1 = 1.5ms, Channel 2 = 800us, Channel 3 = 1ms, Channel 4 = 1.25ms, Channel 5 = 2ms, Channel 6 = 1.75ms, and an inter-channel gap 400us (Note: most RC controllers additionally have an “idle” channel, which sets the PPM frame to ‘1’ after the last channel). Assuming a 100 MHz clock is used to measure time, label each aspect of the PPM frame in terms of milliseconds and in clock cycles.

In CprE 281 you learned about a model of computation known as the Finite State Machine (FSM). FSMs are often used for generating, capturing, and analyzing communication protocols. For MP-1, you will need to design an FSM-based hardware module to capture PPM frame input from an RC transmitter, as well as generate PPM frames to output to a second RC transmitter.

- iii. Draw the bubble diagram design of a Mealy-type state machine for capturing PPM frames that are comparable to the 6 channel PPM frame in part 2ii).
- iv. Draw the bubble diagram design of a Mealy-type state machine for generating PPM frames that are comparable to the 6 channel PPM frame in part 2ii).

Data sheets are the means by which embedded software developers find information needed to interface to hardware. Throughout CprE 288 (as well as in MP-0) you used data sheets in this manner. Data sheets can also be viewed as a contract between software and hardware developers. More specifically, a hardware peripheral's register file definition provides an abstract view of the component for software developers. Once this has been defined, in theory, software development and hardware development can proceed independently of each other.

For MP-1, as you will be designing both the software side and hardware side of the system, you will need to have a well-defined HW/SW interface. Read through the entire MP-1 document and provide a table, where each row of the table has a column for: a) register name, b) relative register offset, and c) a short register description. See the VDMA datasheet from MP-0 for an example.

Finally, before one begins to implement a system, it is important to have a clear vision of the "big picture" layout in terms of how the relevant components relate to one another. Draw the high-level architecture of the MP-1 PPM interfacing system that illustrates the location and interconnection of the primary components: a) external inputs and outputs, b) PPM frame capture, c) PPM frame generation, d) axi_ppm register file, e) software instruction and data memory storage, f) the CPU, and g) system bus.