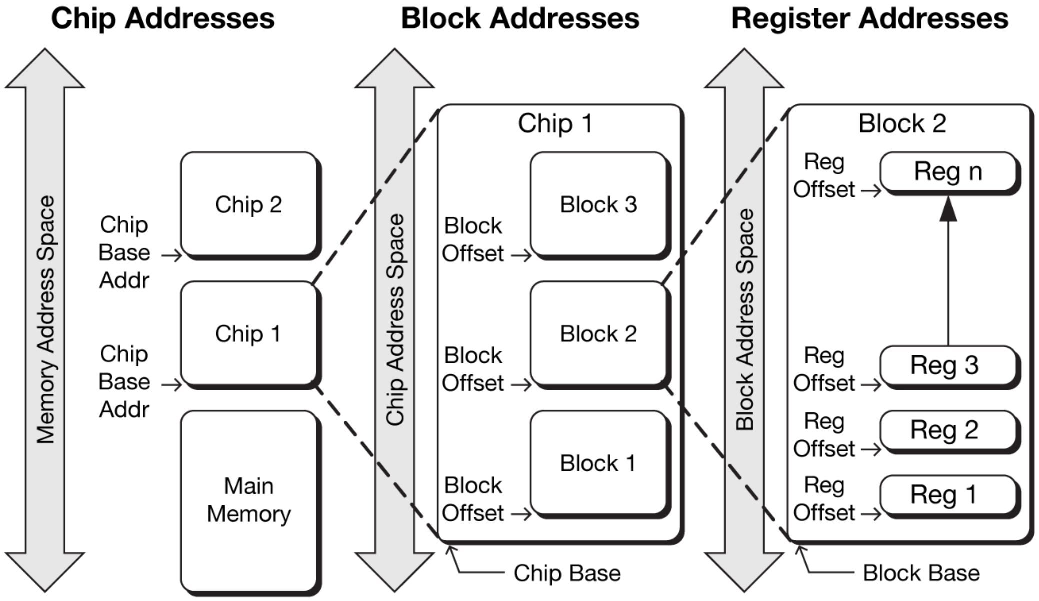
# EELE 475
# Project Proposal
# Due Tuesday November 10, 2015

Write up a proposal of what your class project will be. Describe the hardware/software interface. Outline several tasks that the hardware will accomplish as well as several tasks that the software will accomplish. You will need to **explicitly define the hardware/software interface** given the instructions below. The model for organizing registers is shown in the following figure. Use this figure in your description and give the appropriate address offsets to the blocks and registers.



Group registers into blocks that are related functionally.  Blocks can be initialization registers, data registers, control registers, debug registers and/or status registers, etc.   Follow the best practices as outlined in the lecture notes on registers.

An example is given below for the PWM registers that have the functional blocks *Control* and *Initialization*. Remember to align the starting address of each block at 256-byte boundaries (i.e. groups of 64 32-bit registers).

## *Block* Description Table

| **Register Name** | **Block_Offset** (in bytes) | **Register_Offset** (in bytes) | **What happens on a** | |
|---|---|---|---|---|
| | | | **Write?** | **Read?** |
| ***Control Registers*** | | | | |
| **PWM_CTRL** | 0x0 | 0x0 | Pulse width is set | The current control word is returned, signed extended (the way it should work!) |
| **PWM_ENABLE** | 0x0 | 0x4 | Turns on & off the PWM signal | Current value is returned |
| ***Initialization Registers*** | | | | |
| **PWM_PERIOD** | 0x100 | 0x0 | Pulse period is set | Current value is returned |
| **PWM_NEUTRAL** | 0x100 | 0x4 | Pulse width when PWM_CTRL = 0 | Current value is returned |
| **PWM_LARGEST** | 0x100 | 0x8 | Pulse width when PWM_CTRL = 127 | Current value is returned |
| **PWM_SMALLEST** | 0x100 | 0xC | Pulse width when PWM_CTRL = -128 | Current value is returned |

For each register, you will need to give a detailed explanation of the register. Below is an example description of the PWM_CTRL register. A word template of the register table is on the D2L site.

## Detailed Register Descriptions

### PWM_CTRL

The PWM_CTRL register controls the PWM pulse width. It is an **8-bit *signed* integer** where a control word of zero sets the pulse width to the defined PWM_NEUTRAL pulse width setting (in clock cycles). A PWM_CTRL value of +127 sets the pulse width to the defined PWM_LARGEST pulse width setting (in clock cycles). A PWM_CTRL value of -128 sets the pulse width to the defined PWM_SMALLEST pulse width setting (in clock cycles).

Reading PWM_CTRL returns the current value as a signed 32-bit integer where the 8-bit signed integer has been signed extended to fill the 32-bit register.

### PWM_CTRL Register Description

| | **MSB** | **PWM_CTRL** | **(Block_Offset = 0x0, Register_Offset=0x0)** | | | | | **LSB** |
|---|---|---|---|---|---|---|---|---|
| Bits | 31 30 29 28 | 27 26 25 24 | 23 22 21 20 | 19 18 17 16 | 15 14 13 12 | 11 10 9 8 | 7 6 5 4 | 3 2 1 0 |
| R/W | - - - - | - - - - | - - - - | - - - -<br>Sign Extend integer to full 32 bits | - - - - | - - - - | **A A A A** | **A A A A** |
| Reset | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 |

**Your proposal should include the following:**

1. A description of what your class project will be.
2. A block description table that shows the register groupings and related information (as in the above example).
3. A description for ***each*** register that shows the bit mappings as well as a description of the registers, what happens on reads and writes, and a description of the data type(s).
4. A description of what your software will do with these registers.
5. Identify the tasks that will go into hardware and the tasks the will be implemented via software.
6. This information should be organized into a typical proposal format (see below).

Note: A register table & figure of the register blocks can be found on the D2L site and in the document *EE475_proposal_templates.doc*

Your proposal should be organized at follows:

## Proposal Organization

Your proposal should have the following sections and headings:

### Title Page

- Title of project
- Names of team members
- Date

### Executive Summary (one page)

- *Content:* A brief summary of the proposal
- *Length:* A paragraph or two
- *Emphasis:* Your mom or dad (who aren't engineers) should be able to read the executive summary, understand it, and say "Cool".
  (Note: Assume the executive does not have a technical background...).

### Body of proposal

**Statement of Problem**: the "Why?"

- Relevance or importance of problem.
- Background information to educate the reader.
- Detailed problem description, as you understand it.

**Objectives**: the “What?”

- Translate quantitative and qualitative needs into clear, objective design specifications. Define the scope of work and clearly state the project.
- Design specifications should be in specific, quantitative terms.
- Emphasize critical design issues, constraints, limitations.
- **Give an overview of the registers you are using, how they are grouped, and what they will be used for.**

**Technical Approach**: the “How?”

- Explain the functionality of your VHDL component.
- **The specific details of each register** should be explained.
- Explain how the **hardware** will use the registers.
- Explain how the **software** will use the registers.