**RISC-V Class Project Phase 8 – MEMCTL Ideas**

This document provides some ideas for the Codasip implementation of the MEMCTL delay block.

1. **Overall Concept**

MEMCTL implements a delay in the start of a memory operation (load or store). When a memory operation is detected, a counter memcnt (which should normally be 0) is incremented on each clock, and when it reaches MAXCNT (a constant initially defined as 4) the memory operation is executed, meaning it is passed to the OP input of the Data Memory address part. When the operation is executed, the correct memory address must also be passed to the Data Memory. For a store operation, the correct memory write data must be sent to the ME pipeline for use in the Data Memory data part in the ME stage. Until the memory operation is executed, the signal s\_ex\_resp must be set to the enum RESP\_WAIT. s\_ex\_resp must be ORed into s\_ex\_stall so that the pipeline is held when it is RESP\_WAIT. When the pipeline can be released, the value of s\_ex\_resp is set to RESP\_ACK.

1. **Saving a Value**

In several cases MEMCTL must save the input value of a signal so that it can be used later even if the input value changes. For example, assume there is an input signal A which needs to be saved when save\_condition is true. There must be a corresponding MEMCTL output signal s\_ex\_B, which goes to a register in the ME pipeline and returns as the MEMCTL input r\_me\_B. The code for saving the value would be:

if (save\_condition) s\_ex\_B = A;

else s\_ex\_B = r\_me\_B;

This will hold the value of A in r\_me\_B until save\_condition is asserted again.

1. **Managing memcnt**

The value of the counter memcnt is held in the ME pipeline register r\_me\_memcnt, and its next value is created by MEMCTL as s\_ex\_memcnt. Just as in the saving case, there is a condition inc\_condition which determines when the counter should be incremented. The code for a pure increment would be:

if (inc\_condition) s\_ex\_memcnt = r\_me\_memcnt + 1;

else s\_ex\_memcnt = r\_me\_memcnt;

This is not the complete logic for memcnt, but provides an example.

1. **Memory Control**

The memory operation is specified by the control signal r\_ex\_memop. MEMCTL uses this input and other information to create signal s\_ex\_memop which presents the memory operation to the Data Memory. Use the memop enums for this process.