Term Project Milestone 2 Evaluation

Evaluation Criteria Categories	Specific Criteria	Comments	Score
Consistency with higher level specifications	Every instruction allowed by the machine language (ML) specification has a unique register transfer language (RTL) representation The sequences of register transfers specified by each RTL description correctly implement the functions described in the assembly language (AL) specification Every component referenced in the RTL descriptions is determined For each component, input, output, and control signals that are sufficient to implement the RTL descriptions are identified, including the size of each signal	RTL is specified for instruction set. RTL follows instruction specifications reasonably closely Several of the goto instructions differ from their expressed function golt and goeq appear to skip the next instruction—a valid design, but they would be better name "skip" instructions, and your assembly/machine language needs to reflect this change. gotob differs from description in that it automatically returns to the address stored in Jump. This means that you still have no way to address the entire memory-space (i.e. jumping across pages) goto and gotof try to cram 17-bits into a 16 bit PC Components are listed and appear reasonable. Reasonable control signals are listed. You cannot dynamically address memory—this is a problem.	2/3
Self-consistency	The effect of each individual RTL statement is unambiguous No state element is assigned more than one value in any given clock cycle	RTL instruction descriptions are clear. No element is assigned multiple values in one cycle. You still don't appear to be able to load a 16-bit constant.	3/3
Demonstration of design principles 1. Simplicity favors regularity 2. Smaller is faster 3. Good design demands good compromises 4. Make the common case fast	Significant delays are balanced between cycles, so that the clock cycle can be as short as reasonably possible Each instruction uses as few clock cycles as possible without extending the clock cycle Each component is used efficiently at each clock cycle, and components are not duplicated unnecessarily	Delay seems, for the most part, to be reasonably distributed. In specific, load and store each have a long final cycle. You should consider the ramifications of completing a memory access in the same cycle as a register access. Since you are accessing one specific register, it may be possible to "bypass" the register file delays. Is it necessary to devote the entire first cycle to checking interrupts? Consider that the third cycle is currently only doing register fetch and instruction decode—would it be reasonable to do interrupt processing there instead?	3/3

		You have to read three values from the register file in one cycle. This will add significant complexity to your design; is there any way to change your design so that you only need two outputs?	
Documentation (see below) Organization Completeness Conciseness Grammar and style	Clear English specifications o The behavior of each component is described unambiguously o Documentation, as listed in the following page, demonstrates all the design issues discussed above	Memo would benefit from memo formatting—you wouldn't turn in something that basic to a boss who asked you for a memo. Word even supplies templates, so the amount of work involved is trivial. Memo is somewhat short, should provide an objective assessment of progress. Design Documentation and Design Process Journal are two different things and should be treated as such. Your Design Journal is a mix of the two, and should be separated into its component parts. Your Design Documentation should not include the discussions at meetings and should be kept up-to-date with any design changes. Your Design Process Journal, on the other hand, should reflect all informal considerations / alternatives discussed, and should not be modified, only appended to each week. This is the "history" of your design, and should explain the motivation behind design decisions. You haven't provided any means to access your special registers. Also, you include A, B, C, and Sum in the special registers, but it is unlikely that you will ever have any need to access them directly. All Design Documentation (AL, ML, etc.) should be kept current. Website looks good, provides access to relevant files / information.	12/16

<u>Required Documents</u>

- Memo
  - Objective assessment of design and status
- Design Documentation
  - Demonstration of conceptual understanding
  - Highlights interesting features
- Design Process Journal
  - Alternatives considered
  - Tradeoffs
  - Decisions
- Webpage