Rose-Hulman Institute of Technology
Computer Science and Software Engineering
Prof. Laurence D. Merkle

Term Project Milestone 1 Evaluation

Evaluation Criteria Categories	Specific Criteria	Comments	Score
Consistency with higher level specifications	❑ Given the semantics of the Assembly Language (AL) specification, the sample program can be implemented ❑ Every instruction allowed by the assembly language (AL) specification has a unique machine language (ML) representation o Each instruction type includes enough fields to represent the information specified in the corresponding AL statements o Each field is allocated enough bits to represent all values allowed by the AL specification o For each instruction type, the total number of bits allocated to fields is not greater than the number of bits available ❑ Sample program is translated into binary as described in ML specification	The AL specification provides a useful instruction set. AL instructions have a unique ML representation. Bits are assigned appropriately to each field. Sample program is translated into binary.	3/3
Self-consistency	❑ Sample program uses the syntax described in AL specification ❑ Sample program uses the registers described in AL specification (number and type) ❑ Sample program uses the representation given in the ML specification, including correct values for fields specifying branch and jump targets	Sample program is consistent with AL specification. Psuedo-instruction la is not defined in AL spec. Sample program uses appropriate registers. Representation in machine code appears correct, including branch and jump target calculations.	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	❑ AL instructions are easy to understand and are not overly specialized ❑ Number of instructions is minimized ❑ Number of registers is minimized ❑ Where the above criteria conflict, good compromises are made (to make the common case fast) ❑ Number of instruction types is small ❑ Instruction types have regularity	Instructions are clearly defined. The number of instructions is reasonable. The number of registers is reasonable. Three instruction types clearly defined, but J-type could probably be combined with A-type. Good use of parallelism in jump / branch instructions.	3/3

Evaluation Criteria Categories	Specific Criteria	Comments	Score

Documentation
❑ Organization
❑ Completeness
❑ Conciseness
❑ Grammar and style
• Memo
  • Objective assessment of design and status
• Design Documentation
  • Demonstration of conceptual understanding
  • Highlights interesting features
• Design Process Journal
  • Alternatives considered
  • Tradeoffs
  • Decisions
• Website
	❑ Clear English specifications
	  o Instruction set (incl. prototypical AL statements)
	  o Registers
	  ▪ Number of general purpose registers
	  ▪ Specification of special purpose registers (if applicable)
	  ▪ Naming conventions
	  ▪ Usage conventions
	  o Instruction types
	  o Representation of each instruction
		Website shows some effort.
		
		Memo would benefit from memo formatting, should be linked from website, but is overall a good summary of progress towards project objectives.
		
		Design Process Documentation is good, reveals some alternatives considered and the reasoning behind them.
		
		Design Documentation clearly defines instructions, registers, and usage conventions.
		
		Using registers to store addresses allows 16-bit addressing, but at what cost to performance? Is there some way you could mitigate some of the problems inherent in this type of addressing (which is really quite reasonable to use for this project)?
		
		Set Less Than doesn't set, it branches. (Corrected in other documentation; keep documentation current)
			16/16