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	Instructions are, for the most part, clearly defined. What addressing mode do instructions use? Sample program is translated into machine code. Bits are properly allocated to instructions.	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 uses conventions of AL. Branch / Jump targets do not appear correct. Sample program doesn't preserve saved register. It is not necessary to goto the next line of a program. AL code should be included in comments to ML code.	2/3
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	Number of instructions is reasonable. Instruction types are clearly identified. Registers are well specified. Shift does division? (only by powers of 2)	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
		Supplied Specifications are clear.
		
		No progress memo supplied in turnin folder.
		
		Documentation is lacking in several key areas:
		compensation for 12-bit addressing
		no (explicit) way to load a 16-bit constant
		no logical operators (are they necessary?)
		is CONSTANT signed or unsigned?
		what addressing mode or modes do instructions use?
		no register usage conventions
		how are procedure calls handled? (no jal)
		will a stack be provided?
		
		Provide separate Design Documentation and Design Process Journal—do not combine into one file.
		
		set is a pseudo-instruction (and should say so) that sets /t0 to the **value** in register /t1
			9/16