

Term Project – Milestone One

- Only requires one meeting log for each milestone, so keep growing the same meeting log each time that we meet throughout the milestone
 - o Just needs the topic discussed, extra detail about it is optional
 - o Best way is to use backlog names when describing

Introduction

- Trying to do a validation, we need to know which, what, where

Utah Valley University (UVU) hired your company IED (Innovative EDucation) to develop a simulator called UVSim for computer science students to learn machine language and computer architecture. Through the technique of software-based simulation the user can execute their machine-language programs on the simulator.

The UVSim is a simple machine, but powerful. The UVSim can only understand a machine language called Basic Machine Language, or BasicML for short.

The UVSim contains an accumulator – a register into which information is put before the UVSim uses it in calculations or examines it in various ways. All the information in the UVSim is handled in terms of words. A word is a signed four-digit decimal number, such as +1234, -5678. The UVSim is equipped with a 100-word memory, and these words are referenced by their location numbers 00, 01, ..., 99. The BasicML program must be loaded into the main memory starting at location 00 before executing. UVSim is also going to simulate Arithmetic Logic Unit (ALU) to let the students learn ALU architecture. The only operation the ALU can have is XOR and AND operation. The simulation need to simulate an n bit full adder. Missing ...

Each instruction written in BasicML occupies one word of the UVSim memory (instruction are signed four-digit decimal number). We shall assume that the sign of a BasicML instruction is always plus, but the sign of a data word may be either plus or minus. Each location in the UVSim memory may contain an instruction, a data value used by a program or an unused area of memory. The first two digits of each BasicML instruction are the operation code specifying the operation to be performed.

BasicML vocabulary defined as follow.

I/O operation:

READ = 10	Read a word from the keyboard into a specific location in memory.
WRITE = 11	Missing

Load/store operations:

LOAD = 20	Load a word from a specific location in memory into the accumulator.
-----------	--

STORE = 21 Missing

Arithmetic operation:

Add = 30 Add a word from a specific location in memory to the word in the accumulator (leave the result in the accumulator)

SUBTRACT = 31 Missing

DIVIDE = 32 Missing

MULTIPLY = 33 Missing

Control operation:

BRANCH = 40 Branch to a specific location in memory

BRANCHNEG = 41 Missing

BRANCHZERO = 42 Missing

HALT = 43 Missing

Debugging operation:

MEMDUMP Print the memory image on the screen.

BREAK Missing

CONTINUE Missing

The last two digits of a BasicML instruction are the operand – the address of the memory location containing the word to which the operation applies.

Deliverable

- (1) (5 points) (Hard Copy) Meeting log. Every team member need to sign the log with contribution. Submission without signature will be count as late and late submission rules will be applied.
- (2) (10 points) (Hard Copy) Decompose the milestone to backlogs and sprints.
- (3) (15 points) (Hard Copy) Write one page to compare SCRUM and one another process model.
 - (a) Give 3 reasons to support, and 3 reasons to against SCRUM process model.
 - (b) Give 3 reasons to support, and 3 reasons to against the process model your team chose.
- (4) (20 points) (Hard Copy) Describe the system to all stakeholders by using Use Case Diagram. The number of use cases is no more than 15 and no less than 10. The range of number is used

to control the granularity of the use cases. Try your best to make sure the use cases are in the same detailed level.

(5) (40 Points) (Hard Copy of the GUI and running results) Develop a prototype of UVSim. You are free to use any object oriented language. Every team member need to make comment on your part of the implementation. Note: the prototype can skip the some of the implementation detail, such as the implementation of ALU, the debugging operation.

(6) (10 points) Your need to test your prototype to show reasonable quality to release. I will provide two test cases and you will develop one. When testing, you can introduce stub in the system if necessary.

Test Case 1:

Test Case 2:

Test Case 3: