Implementação de nova instrução Marie

Andrey Mota, Bruno Caike e Gabriel Cardoso

FACOM – Faculdade de Computação

UFMS -Universidade Federal de Mato Grosso do Sul

RESUMO

No relatório a seguir é descrito um sistema com base na arquitetura Marie onde é implementado para reconhecer uma nova instrução que pode realizar tanto multiplicação quanto divisão de números inteiros. A instrução adicionada utiliza-se de um dos registros propostos pela arquitetura Marie, denominado "AC" *(Accumulator)*. O AC possui um endereçamento de 16 bits (4 Bytes) que serve para instruções condicionais. A instrução implementada utiliza o registrador AC como parâmetro de funcionamento, de forma que, se o mesmo for igual a 0 executará uma multiplicação entre o seu valor e o valor armazenado no endereço informado como operando e se o AC for 1 acontece uma divisão entre o AC e o valor informado como operando.

Palavras-chave: Sistema. Arquitetura. Marie. Instrução. Endereço

ABSTRACT

The following report describes a system based on the Marie architecture where it is implemented to recognize a new instruction that can perform both multiplication and division of integers. The added instruction uses one of the records proposed by the Marie architecture, called "AC" (Accumulator). The AC has a 16-bit address (4 Bytes) that is used for conditional instructions. The implemented instruction uses the AC register as a functioning parameter, so that, if it is equal to 0, it will perform a multiplication between its value and the value stored in the address informed as operand and if the AC is 1, a division occurs between the AC and the value reported as operating.

Keywords: System. Architecture. Marie. Instruction. Address

INTRODUÇÃO

A arquitetura Marie (Machine Architecture that is Really Intuitive and Easy) é uma linguagem de arquitetura de máquinas proposta no livro escrito por Linda Null e Julia Lobur (The Essentials of Computer Organization and Architecture), bastate intuitiva e fácil de ser utilizada.

Na ISA (instruction set architecture – arquitetura de conjunto de instrução) da arquitetura Marie cada instrução possui endereçamento de 16 bits, que podem ser divididos em duas partes, o código de instrução (opcode) com 4 bits que indicam o número em hexadecimal da instrução e o endereço da instrução com 12 bits que são utilizados como operando.

As instruções básicas da arquitetura Marie são 10, sendo elas: Load (carrega valor), Store (armazena valor), Add (adiciona valor), Subt (subtrai valor), Input (recebe valor de entrada), Output (imprime um valor como saída), Halt (indica a finalização do programa), Skipcond (pula uma linha conforme condições definidas e Jump (pula para determinado trecho do programa.

A figura 2, apresentada pelo professor Ricardo em uma de suas aulas apresenta as instruções básicas da arquitetura Marie e algumas informações importantes que foram necessárias durante o desenvolvimento do programa, como o código em hexadecimal e a função de cada uma.

Figura 1 – Instruções básicas da arquitetura Marie.

Instruction Number			
Binary	Hex	Instruction	Meaning
0001	1	Load X	Load contents of address X into AC.
0010	2	Store X	Store the contents of AC at address X.
0011	3	Add X	Add the contents of address X to AC.
0100	4	Subt X	Subtract the contents of address X from AC.
0101	5	Input	Input a value from the keyboard into AC.
0110	6	Output	Output the value in AC to the display.
0111	7	Halt	Terminate program.
1000	8	Skipcond	Skip next instruction on condition.
1001	9	Jump X	Load the value of X into PC.

Apesar do simulador online utilizado ser programado para reconhecer instruções diferentes das citadas (como é o caso das instruções JnS, LoadI e StoreI que não serão abordadas), que foram também utilizadas no desenvolvimento do sistema, todas elas são construídas utilizando as instruções básicas, sendo assim, apenas uma facilitação ao utilizar algo que poderia ser desenvolvido com elas.

METODOLOGIA DE DESENVOLVIMENTO

Para implementar um sistema que reconhecesse uma nova instrução Marie, foram necessários conhecimentos previamente adquiridos, em sua grande parte, no conteúdo da disciplina “Organização de Computadores, ministrada pelo docente: Doutor Ricardo Ribeiro dos Santos. Antes de tudo foi necessário revisar o material disponibilizado pelo mesmo, que dentre outros conteúdos apresenta também os fundamentos essenciais da Arquitetura de Processador Marie, fundamentando no livro escrito por Linda Null e Julia Lobur (The Essentials of Computer Organization and Architecture).

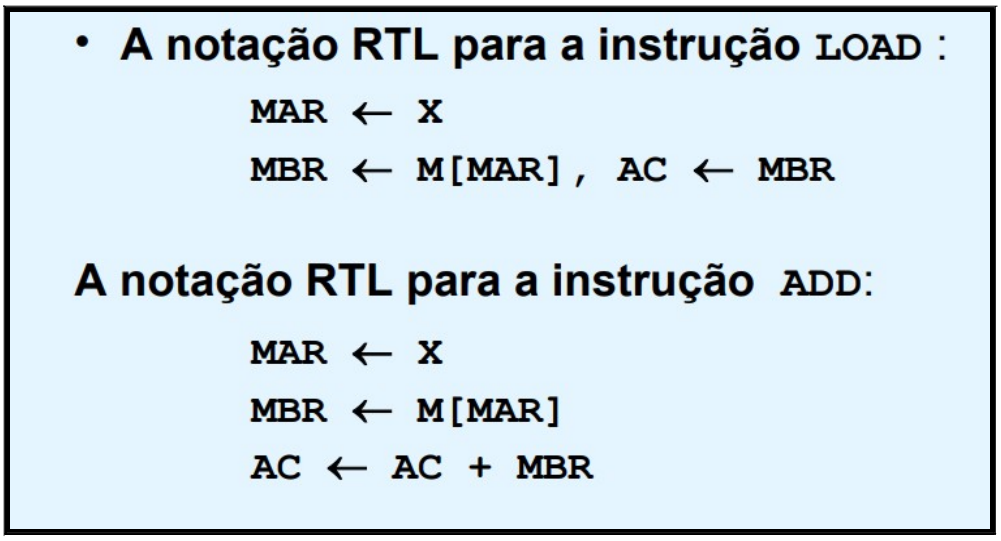
Ao iniciar o desenvolvimento do sistema em linguagem de montagem, ou linguagem Assembly, realizado em um simulador Marie online escrito em JavaScript (Marie.js), foi percebida a necessidade de revisar as instruções manipuladas. Os principais pontos de revisão constante, foram a análise dos números de cada instrução em hexadecimal (Opcodes) e as microinstruções contidas em cada uma delas.

A análise do Opcode de cada instrução possibilitou diferenciá-las através de seus números de instrução. Inicialmente a proposta era fazer com que o opcode fosse lido junto com o endereço da instrução em um único endereço de memória, no entando, foram encontradas dificuldades nas diversas tentativas de implementação. A principal dificuldade encontrada foi a limitação de representação númericada do simulador, que não permitiu manipulações de operações matemáticas simples (soma e subtração), pois resultados eram completamente alterados ao atingir tanto o limite superior, quanto o limite inferior das representações possíveis.

Com isso, a única alternativa encontrada foi fazer com que cada instrução ocupasse 2 endereços de memória (um para o opcode e o outro para o endereço). Apesar da noção do desperdício de memória, foi a única solução que o grupo encontrou para conseguir realizar a implementação de forma precisa, o que também facilitou a manipulação das instruções no decorrer do desenvolvimento.

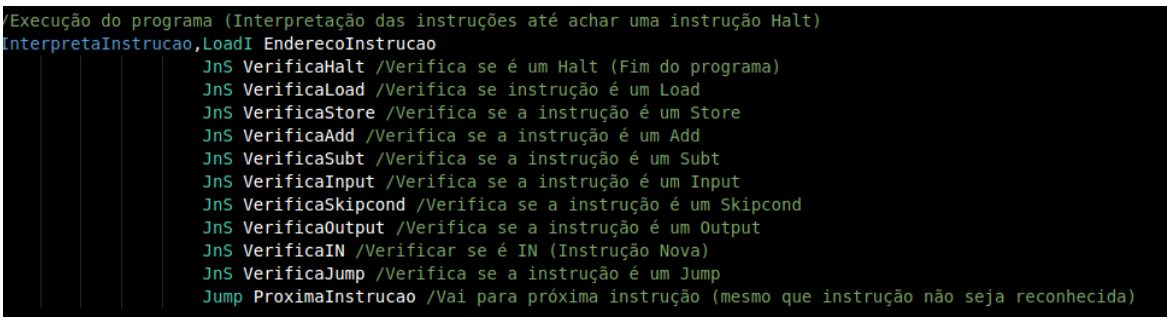
A análise das microinstruções foram necessárias para replicar o funcionamento de cada uma delas no sistema. Além de ser necessário entender todas as etapas que cada instrução passa para ser executada, as microinstruções análisadas facilitaram a abstração das possíveis microinstruções da nova instrução implementada.

Figura 1: Microinstruções das instruções Load e Add.



A execução do programa da implementação tem uma etapa que pode ser definida como principal, que é a interpretação de instrução. Com base no código de instrução (opcode) de cada uma das instruções da arquitetura Marie é possível reconhecer as instruções e simular a execução de suas microinstruções. O processo de interpretação utiliza ainda o endereço da instrução para simular suas microinstruções.

Figura 3 – Trecho do código que realiza a interpretação das instruções



```
/Execução do programa (Interpretação das instruções até achar uma instrução Halt)
InterpretaInstrucao,LoadI EnderecoInstrucao
                    JnS VerificaHalt /Verifica se é um Halt (Fim do programa)
                    JnS VerificaLoad /Verifica se instrução é um Load
                    JnS VerificaStore /Verifica se a instrução é um Store
                    JnS VerificaAdd /Verifica se a instrução é um Add
                    JnS VerificaSubt /Verifica se a instrução é um Subt
                    JnS VerificaInput /Verifica se a instrução é um Input
                    JnS VerificaSkipcond /Verifica se a instrução é um Skipcond
                    JnS VerificaOutput /Verifica se a instrução é um Output
                    JnS VerificaIN /Verificar se é IN (Instrução Nova)
                    JnS VerificaJump /Verifica se a instrução é um Jump
                    Jump ProximaInstrucao /Vai para próxima instrução (mesmo que instrução não seja reconhecida)
```

Após a interpretação e simulação de uma instrução, o programa segue para o próximo endereço de memória que contém uma nova instrução (no caso uma instrução está dividida em dois endereços logo os endereços serão passados de dois em dois).

O programa continuará sua execução e interpretando instruções até que seja reconhecida uma instrução Halt (que tem o opcode em hexadecimal: 7), pois ela indicará que o programa chegou até o fim.

CONCLUSÃO

O trabalho realizado contribuiu grandemente com os conhecimentos técnicos e teóricos dos integrantes do grupo com relação a arquitetura de computadores, principalmente a arquitetura Marie.

A implementação de uma nova instrução Marie proposta foi realizada com sucesso ainda que com algumas limitações impostas no decorrer do desenvolvimento que de certa forma torna o sistema uma implementação teórica que não pode ser implementada na prática em uma organização de computador real.

Para que fosse possível implementar a nova instrução de uma forma mais fundamentada seria necessário um longo período de testes e estudos de como lidar com as limitações encontradas na arquitetura Marie que dificultaram o desenvolvimento.

REFERÊNCIA BIBLIOGRÁFICA:

Essentials of Computer Organization and Architecture. Linda Null e Julia Lobur.