Desenvolvimento de algoritmos para a nova versão do Microprocessador acadêmico “SDIII-16B”

Lucas Schiavon[1], Orlando Del Bianco Filho
Departamento de Engenharia Elétrica, Centro Universitário da FEI
Lschiavon97@gmail.com; orlandof@fei.edu.br

Resumo: Este projeto de iniciação didática tem como objetivo o desenvolvimento de algoritmos para uma nova versão do microprocessador acadêmico “SDIII-16B”. Apresentaremos dois algoritmos desenvolvidos e testados até o momento desse artigo. Uma sequência de Fibonacci e um Multiplicador de dois números inteiros através de somas sucessivas. Ambos com limite de dados de 255, que corresponde a 8 bits, o tamanho dos registradores internos do microprocessador.

1. Introdução

Com o desenvolvimento da nova versão do microprocessador acadêmico utilizado em algumas aulas dos cursos de Engenharia Elétrica e de Engenharia de Controle e Automação do Centro Universitário da FEI, este projeto de iniciação didática busca desenvolver uma série de algoritmos capazes de aumentar o número de exemplos a serem discutidos nas salas de aula e nos laboratórios, mostrando as novas instruções executáveis pela nova versão do microprocessador “SDIII-16B”.

Inicialmente desenvolvido pelo Prof. Dr. Pedro Luis Benko, o processador passou por alterações que visaram o aumento do número de instruções executáveis. Essas alterações foram desenvolvidas pelo aluno Paulo Henrique Guidolin, que, resumidamente, reduziu o numero de registradores internos a UCP para 8, liberando um dos bits de endereçamento. Este bit foi agregado ao campo de instrução, o que permitiu ampliar o limite de instruções executáveis para 32[1]. Dessas 32 instruções executáveis, as 16 que eram utilizadas na primeira versão do microprocessador permaneceram, além delas, foram acrescentadas mais 6[2], que são:

1)Pulo Longo;
2)Swap do Conteúdo de Registrador;
3)Incremento do Conteúdo de Registrador;
4)Decremento do Conteúdo de Registrador;
5) Salto para Sub-rotina;
6)Retorno para Sub-rotina;

A partir dessas novas instruções, foram propostos 5 projetos de desenvolvimentos de algoritmos para a nova versão do microprocessador:

1) Geração da sequência dos números de Fibonacci, menores que 255;

2) Verificação das medidas de um triângulo retângulo (a2 = b2 + c2);

3) Multiplicação de 2 números inteiros, através de somas sucessivas, com sinalização em caso de estouro;

4) Divisão inteira entre 2 números, ou seja, com geração de quociente e resto;

5) Geração do “byte” de paridade, para um conjunto entre 4 e 8 bytes de dados.;

Até o momento foram desenvolvidos dois algoritmos, a Geração da sequência de Fibonacci e para facilitar a criação dos demais, desenvolvi primeiro a Multiplicação de dois números inteiros através das somas sucessivas.

2. Metodologia

O desenvolvimento dos algoritmos é algo, em teoria, simples. O primeiro passo é a definição do diagrama de blocos com a lógica a ser seguida para cada um deles. A seguir, deve-se detalhar instrução a instrução os comandos necessários para a execução do programa. E em seguida, introduzir a instrução no módulo correto do microprocessador, sintetizar, e por fim, testar na placa.

3. Resultados

Como este projeto de iniciação didática é dividido em projetos de desenvolvimento de algoritmos, cada um deles apresenta um resultado diferente. Portanto, apresentaremos para os projetos desenvolvidos até o momento, como cada um foi desenvolvido e o seu resultado teórico, visto que podemos apresentar os algoritmos funcionando na data da apresentação do VIII Simpósio de Iniciação Científica, Didática e de Ações Sociais da FEI.

Sequência de Fibonacci

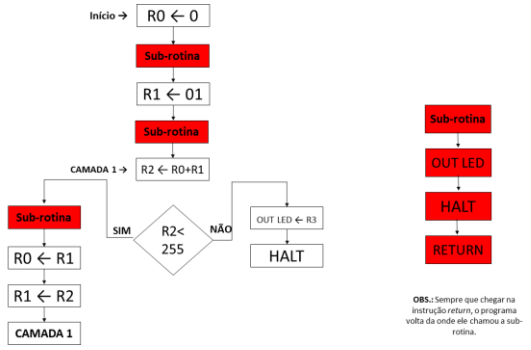

Figura 1 – Diagrama de blocos com a lógica para sequência de Fibonacci.

```
tmp_rom <= ( -- exemplo ilustrativo de operacao
 -- (programa a ser carregado (cada aluno fara o seu nessa area)

 0 => "0010000000000000", -- MVI R0,0;
 1 => "0011101100000000", -- AND R3,R0,R0;
 2 => "1010000000010010", -- CALL (18); Chamada da sub-rotina;
 3 => "0010000100000001", -- MVI R1,1;
 4 => "0011101100100100", -- AND R3,R1,R1;
 5 => "1010000000010010", -- CALL (18); Chamada da sub-rotina;
 6 => "0010101000100000", -- ADD R2,R1,R0;
 7 => "0110000000000000", -- JMPC;
 8 => "0101100000001010", -- GOTO #10;
 9 => "0101100000001111", -- GOTO #15;
 10 => "0011101101001000", -- AND R3,R2,R2;
 11 => "1010000000010010", -- CALL(18); Chamada da sub-rotina;
 12 => "0011100000100100", -- AND R0,R1,R1;
 13 => "0011100101001000", -- AND R1,R2,R2;
 14 => "0101100000000110", -- GOTO #6;
 15 => "0010001111111111", -- MVI R3,FF;
 16 => "0001101100000000", -- OUT LED, R3;
 17 => "0111100000000000", -- HALT;
 18 => "0001101100000000", -- OUT LED, R3; Sub-rotina;
 19 => "1010100000000000", --  RETURN; Fim da sub-rotina e retorno;
 others => "0000000000000000");
```

Figura 2 – Instruções para sequência de Fibonacci.

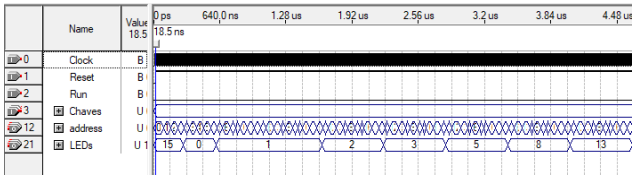

Figura 3 – Primeira parte da simulação da sequência de Fibonacci.

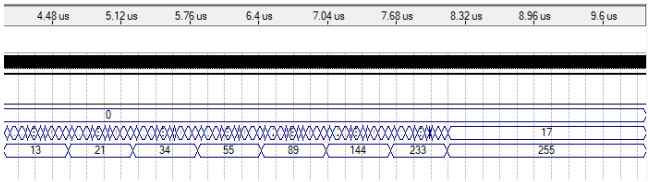

Figura 4 – Segunda parte da simulação da sequência de Fibonacci.

Multiplicador por somas sucessivas

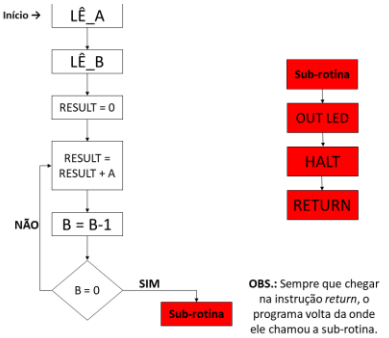

Figura 5 – Diagrama de blocos com a lógica do multiplicador por somas sucessivas.

```
31  tmp_rom <= ( -- exemplo ilustrativo de operacao
32   -- (programa a ser carregado (cada aluno fara o seu nessa area)
33
34   0 => "0001000000000000", -- IN R0,CH1;
35   1 => "0011101100000000", -- AND R3,R0,R0;
36   2 => "1010000000010001", -- CALL (17); Chama sub-rotina;
37   3 => "0001000100000000", -- IN R1,CH2;
38   4 => "0011101100100100", -- AND R3,R1,R1;
39   5 => "1010000000010001", -- CALL (17);Chama sub-rotina;
40   6 => "0010001000000000", -- MVI R2,#0;
41   7 => "0010101001000000", -- ADD R2,R2,R0;
42   8 => "0110000000000000", -- JMPC;
43   9 => "0101100000001011", -- GOTO #11;
44   10 => "0101100000001111", -- GOTO #15;
45   11 => "1001000100000000", -- DECREMENTA UM R1,R1-1;
46   12 => "0111000000000000", -- JMPZ;
47   13 => "0101100000000111", -- GOTO #7;
48   14 => "0011101101001000", -- AND R3,R2,R2;
49   15 => "0001101100000000", -- OUT LED R3;
50   16 => "0111100000000000", -- HALT;
51   17 => "0001101100000000", -- OUT LED R3;
52   18 => "1010100000000000", -- RET;
53   others => "0000000000000000");
54
55
```

Figura 5 – Instruções para o multiplicador por somas sucessivas.

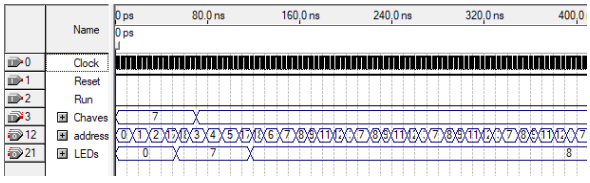

Figura 6 – Primeira parte da simulação do multiplicador por somas sucessivas.

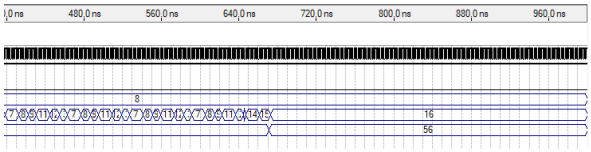

Figura 7 – Segunda parte da simulação do multiplicador por somas sucessivas.

4. Conclusões

O projeto visa a utilização das novas instruções do microprocessador "SDIII-16B" e até o presente momento cumpre com o seu objetivo principal, porque nos dois projetos aqui apresentados, foram utilizadas algumas das novas instruções

5. Referências

[1] Vahid, F. Digital Desing: with RTL Desing, VHDL and Verilog, 2ª ed. John Wiley and Sons, 2011.

[2] Guidolin, P.H. Relatório final do projeto de iniciação científica "Novas Instruções para o Microprocessador acadêmico "SDIII-16B", 2017.

Agradecimentos

À instituição Centro Universitário da FEI pela realização das medidas ou empréstimo de equipamentos.

[1] Aluno de ID do Centro Universitário FEI (ou FAPESP, CNPq ou outra). Projeto com vigência de 04/18 a 03/19.