

Universidade Federal da Fronteira Sul
Campus Chapecó
Curso de Ciência de Computação
Disciplina Sistemas Digitais
Prof. Adriano Sanick Padilha
Acadêmicos: Luan Bortoli e Yuri Luis Malinski Lanzini

Relatório do Projeto

Este relatório tem como objetivo explicar a implementação na DE1 do projeto completo baseado em transferência de registradores, bloco operativo (BO) e bloco de controle (BC) para resolução da função de segundo grau ax^2+bx+c .

Detalhamento do projeto do bloco operativo

O projeto do bloco operativo é composto pelos seguintes componentes:

- 3 multiplexadores 4x1 de 16 Bits cada, denominados mux0, mux1 e mux2;
- 3 registradores de 16 bits, chamado de Saida_LS, Saida_LH e Saida_LX;
- 1 unidade aritmética chamada de operador.

As entradas do BO são alavancas SW de 10 bits, Push Bottons (KEY) de 3 bits, e as saídas são HEX0, HEX1, HEX2 e HEX3 de 7 bits, pois são 7 segmentos no display.

Os multiplexadores são compostos por 4 entradas de 16 bits, 1 seletor de 2 bits e 1 saída de 10 bits, as entradas são definidas de acordo com o seletor, sendo 00 para a entrada de Y, 01 para a entrada do valor de A, 10 para a entrada do valor de B e 11 para a entrada de valor de C.

Os valores de A, B e C são valores fixos definidos por parâmetros, tendo valor de 5, 2 e 9, respectivamente.

Para o mux0, as entradas de A, B e C são os valores definidos por parâmetros, o valor de Y é definido por 0, e a saída do mux0 chamamos de saida_M0. Para o mux1, as entradas são a Saida_LX, Saida_LS, Saida_LH e saida_M0, e a saída chamamos de saida_M1. Para o mux2, as entradas são saida_M0, Saida_LS, Saida_LX e a saída chamados de saida_M2. Todos os mux, eles possuíam uma alavanca para definir quando o valor de A, B, C e Y deveriam ser carregados.

O operador tem 2 entradas de 16 bits cada, 1 entrada de 2 bits para h e uma saída 16 bits. As entradas são a saida_M2, saida_M1 e H, e a saída é chamada de saida_Somador. H é representado por 0 para ser soma e 1 para multiplicação, onde a cada estado, ele assume uma operação diferente.

Cada registrador possui uma funcionalidade, onde a Saida_LX recebe o X que possui entrada de 8 bits e saída de 16 bits, a Saida_LS e Saida_LH recebem o valor da saida_Somador, cada registrador ele possui um valor de 0 e 1, e assume a

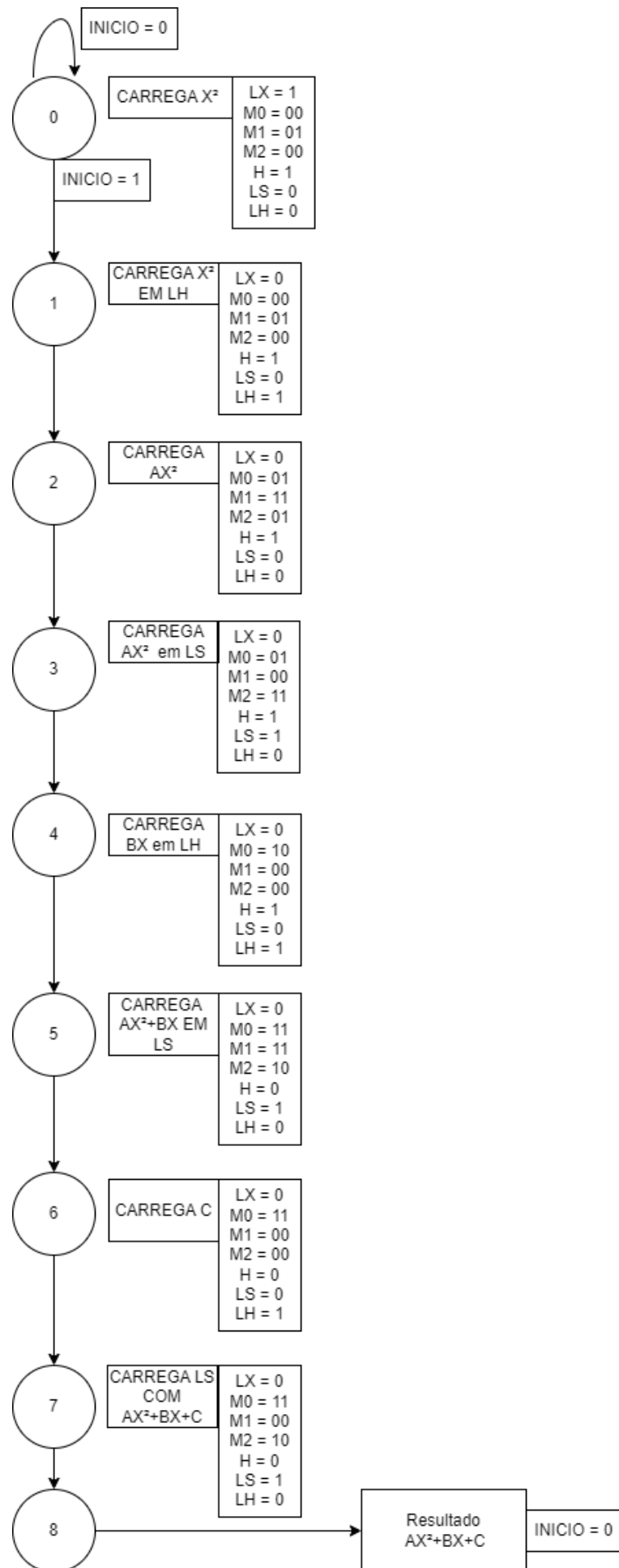
sua função a cada pulso de clock, após a cada operação o valor de `saida_LS` é armazenado em uma saída chamada `result` e é apresentado este valor no display.

Vale salientar que a `saida_M0`, `saida_M1`, `saida_M2` e `saida_Somador` possuem 16 bits e são utilizados para conectar dois pontos, em código é denominado de “Wire”.

O display possui a entrada do resultado da `saida_LS`, e as saídas chamadas de `HEX0`, `HEX1`, `HEX2`, `HEX3`, onde para realizar a impressão na forma correta em cada HEX do valor correspondente, foi realizada operações matemáticas.

Detalhamento do bloco de controle baseada em máquina de estados finitos (FSM)

O bloco de controle possui as entradas de 1 bits para o Clock e Início, as saídas de 1 bits para `LX`, `H`, `LS` e `LH`, e de 2 Bits para `M0`, `M1` e `M2`. Possui uma variável chamada de estado para controlar em qual estado está ocorrendo tal operação. Pode ser observado na imagem abaixo, o comportamento da função em 2º grau em cada estado que a mesma passa:



Explicação da conexão entre BC:BO

Para a conexão do BC com o BO, foi necessário realizar alguns ajustes no BO para que ele funcionasse corretamente, as entradas anteriores foram substituídas por LS, LX, LH, H, CLK de 1 bits, M0, M1, M2 de 2 bits, X de 8 bits. e a saída seria o result de 16 bits, este correspondente ao resultado final que será apresentado no display. O display foi implementado no top level, onde ocorreram as conexões de BC com BO.

O módulo do top level foi chamado de final, onde possui como entrada SW de 10 bits, e KEY de 4 bits, e como saída o HEX0, HEX1, HEX2 e HEX3 de 7 bits onde representam o display.

O Clock foi definido para ocorrer sempre que clicado no KEY0, o início da operação foi definido na alavanca SW[9], e o valor do X seria definido nas alavancas SW[7] até SW[0].

As saídas do Bloco de Controle lx, h, ls, m0, m1 e m2 se tornam as entradas do Bloco Operativo, os mesmos são conectados pelo fio de conexão "wire". O Bloco Operativo tem como saída do resultado DO que é conectado com o módulo do display para que seja impresso o valor da operação no display.

As conexões lx, ls, lh e h possuem 1 bit, m0, m1 e m2 possuem 2 bits, e o resultado_BO possui 16 bits.

O top level pode ser observado na imagem abaixo:

```
module final(
    input [9:0] SW,
    input [3:0] KEY,
    output[6:0] HEX0, HEX1, HEX2, HEX3
);

wire[15:0] resultado_BO;
wire[1:0] m0, m1, m2;
wire lx, ls, lh, h;

BC BlocoControle(.clk(KEY[0]), .inicio(SW[9]), .LX(lx), .H(h), .LS(ls), .LH(lh), .M0(m0), .M1(m1), .M2(m2)); //Saidas lx,h,ls,m0,m1,m2
BO BlocoOperativo(.LS(ls), .LX(lx), .LH(lh), .H(h), .clk(KEY[0]), .M0(m0), .M1(m1), .M2(m2), .X(SW[7:0]), .result(resultado_BO)); //Saida Result
display dis(.R(resultado_BO), .display0(HEX0), .display1(HEX1), .display2(HEX2), .display3(HEX3)); //Mostra o AX, Mostra AX^2 + B, Mostra AX^2+BX+C
endmodule
```

Discussão dos resultados encontrados

O desenvolvimento do projeto foi de extrema importância para que aplicássemos todos os conhecimentos adquiridos em sala de aula e nas atividades práticas, demonstrando que saímos da abstração para algo real e funcional, o que incentiva o estudante a continuar aprofundando o conhecimento e construindo novos projetos nesta área.

O uso do Quartus II foi de extrema importância, pois foi através dele que desenvolvemos os algoritmos e conseguimos fazer a conexão entre o bloco operativo e o bloco de controle, assim tendo sucesso no resultado final.

Para apresentação do resultado no display, foi colocado o X valendo 7.

